Final Report of Convex Optimization
An Introduction to Robust Optimization

Kairui Jin

December 2022

1 Motivation

Consider a factory that produces $Drug1$ and $Drug2$ with $Raw1$ and $Raw2$. The parameters for this problem are shown in the following figure.

Parameter	DrugI	DrugII
Selling price, $ per 1000 packs	6,200	6,900
Content of agent A, g per 1000 packs	0.500	0.600
Manpower required, hours per 1000 packs	90.0	100.0
Equipment required, hours per 1000 packs	40.0	50.0
Operational costs, $ per 1000 packs	700	800

(a) Drug production data

Raw material	Purchasing price, $ per kg	Content of agent A, g per kg
RawI	100.00	0.01
RawII	199.90	0.02

(b) Contents of raw materials

Budget, $	Manpower, hours	Equipment, hours	Capacity of raw materials storage, kg
100,000	2,000	800	1,000

(c) Resources

The problem can be immediately modeled as the following linear programming

$$\begin{array}{rcl}\text{Opt} & = & \min\Big\{ \overbrace{[100\cdot RawI + 199.90\cdot RawII + 700\cdot DrugI + 800\cdot DrugII]}^{\text{purchasing and operational costs}} \\ & & \quad - \underbrace{[6200\cdot DrugI + 6900\cdot DrugII]}_{\text{income from selling the drugs}} \Big\} \qquad \text{[minus total profit]}\end{array}$$

$$\begin{array}{lr}\text{subject to} & \\ 0.01\cdot RawI + 0.02\cdot RawII - 0.500\cdot DrugI - 0.600\cdot DrugII \geq 0 & \text{[balance of active agent]} \\ RawI + RawII \leq 1000 & \text{[storage constraint]} \\ 90.0\cdot DrugI + 100.0\cdot DrugII \leq 2000 & \text{[manpower constraint]} \\ 40.0\cdot DrugI + 50.0\cdot DrugII \leq 800 & \text{[equipment constraint]} \\ 100.0\cdot RawI + 199.90\cdot RawII + 700\cdot DrugI + 800\cdot DrugII \leq 100000 & \text{[budget constraint]} \\ RawI, RawII, DrugI, DrugII \geq 0 & \end{array}$$

The optimal solution of the LO problem is $Opt = -8819.658$, $RawI = 0$, $RawII = 438.789$ and $DrugI = 17.552$, $DrugII = 0$.

There are, however, some problems. First, in real production, real data fluctuates around the mean, so it is not possible to capture this uncertainty in a deterministic-data-based model. Second, when the data fluctuates, the above solution will lose its optimality, even feasibility. Therefore, we need use robust optimization, stochastic optimization, sensitivity analysis and other methods to overcome the impact of data fluctuates. The goal of robust optimization is to find an optimal solution in the robust feasible solution. In this article, we introduce robust optimization which find an optimal solution among robust feasible solution.

2 Notation and Definition

An robust Optimization problem can be written as follow

$$\min_x \left\{c^T x : Ax \leq b \quad \forall (A, b) \in U\right\}$$

where U is a uncertainty set as follow

$$U = \left\{ \begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} A_0 & b_0 \end{bmatrix} + \textstyle\sum_{\ell=1}^{L} \zeta_\ell \begin{bmatrix} A_\ell & b_\ell \end{bmatrix} : \zeta \in Z \subset \mathbf{R}^L \right\}$$

ζ is a perturbation vector and Z is a perturbation set. The perturbation set Z can take a variety of forms. A vector $x \in \mathbf{R}^n$ is a robust feasible solution, if it satisfies all realizations of the constraints from the uncertainty set.

In this article, we always consider a affine uncertainty set as above. In addition, we only consider the problem with certain objective and uncertain constraints since if the objective is uncertain, then we can written the problem as a epigraph form to get a problem with certain objective and uncertain constraints.

3 Simple Cases

3.1 Box uncertainty

The outlined strategy allow us to focus on the single constraint. So we will analyze the single constraint problems, and the conclusion can be applied to the multiple constraints problems.

$$\min_x \{c^T x : a^T x \leq b, [a, b] \in U\}$$

Review the definition of uncertainty set as follow where ζ is perturbation vector and Z is perturbation set.

$$U = \left\{ \begin{bmatrix} a & b \end{bmatrix} = \begin{bmatrix} a_0 & b_0 \end{bmatrix} + \textstyle\sum_{\ell=1}^{L} \zeta_\ell \begin{bmatrix} a_\ell & a_\ell \end{bmatrix} : \zeta \in Z \subset \mathbf{R}^L \right\}$$

Consider the case of interval uncertainty, where Z in a box. W.l.o.g. we can normalize the situation by follow

$$Z = \{\zeta \in \mathbf{R}^L : \|\zeta\|_\infty \leq 1\}$$

Then the robust model can be written as follow

$$\min_x \{c^T x : a^T x \leq b \ \ \forall (a, b) \in U\}$$

Further, the uncertainty-affected constraint can be written as

$$
\begin{aligned}
& [a_0 + \sum_{\ell=1}^{L} \zeta_\ell a_\ell]^T x \le b_0 + \sum_{\ell=1}^{l} \zeta_\ell b_\ell \\
\Leftrightarrow & a_0^T x + \sum_{\ell=1}^{L} \zeta_\ell a_\ell^T x \le b_0 + \sum_{\ell=1}^{l} \zeta_\ell b_\ell \\
\Leftrightarrow & \sum_{\ell=1}^{L} \zeta_\ell [a_\ell^T x - b_\ell] \le b_0 - a_0 x \\
\Leftrightarrow & \max_{\|\zeta\|_\infty \le 1} \left\{ \sum_{\ell=1}^{L} \zeta_\ell [a_\ell^T x - b_\ell] \right\} \le b_0 - a_0^T x
\end{aligned}
$$

The left side is a optimization problem with optimal value $\sum_{\ell=1}^{L} |a_\ell^T x - b_\ell|$. Therefore, the above can be written as

$$
\begin{aligned}
\Leftrightarrow & \sum_{\ell=1}^{L} |a_\ell^T x - b_\ell| \le b_0 - a_0^T x \\
\Leftrightarrow & \begin{cases} -u_\ell \le a^T x - b_\ell \le u_\ell, \quad \forall \ell = 1, \cdots, L \\ \qquad a_0^T x + \sum_{\ell=1}^{L} u_\ell \le b_0 \end{cases}
\end{aligned}
$$

In this way, the robust model with box uncertainty can be rewritten as a tractable representation.

$$
\begin{aligned}
\min \quad & c^T x \\
s.t. \quad & -u_\ell \le a^T x - b_\ell \quad \forall \ell = 1, \cdots, L \\
& a^T x - b_\ell \le u_\ell \quad \forall \ell = 1, \cdots, L \\
& a_0^T x + \sum_{\ell=1}^{L} u_\ell \le b_0
\end{aligned}
$$

3.2 Ball uncertainty

Similar to the box uncertainty, but we consider a different perturbation set Z as follow where Ω is the radius.

$$
Z = \{\zeta \in \mathbf{R}^L : \|\zeta\|_2 \le \Omega\}
$$

Further, the uncertainty-affected constraint can be written as

$$
\begin{aligned}
&[a_0 + \sum_{\ell=1}^{L} \zeta_\ell a_\ell]^T x \le b_0 + \sum_{\ell=1}^{l} \zeta_\ell b_\ell \\
\Leftrightarrow &\max_{\|\zeta\|_\infty \le 1} \left\{ \sum_{\ell=1}^{L} \zeta_\ell [a_\ell^T x - b_\ell] \right\} \le b_0 - a_0^T x \\
\Leftrightarrow &\Omega \sqrt{\sum_{\ell=1}^{L} (a_\ell^T x - b_\ell)^2} \le b_0 - a_0^T x
\end{aligned}
$$

In this way, the robust model with ball uncertainty can be rewritten as a tractable representation.

$$
\begin{aligned}
\min \quad & c^T x \\
s.t. \quad & a_0^T x + \Omega \sqrt{\sum_{\ell=1}^{L} (a_\ell^T x - b_\ell)^2} \le b_0
\end{aligned}
$$

4 General Cases

4.1 General Cases

Now consider a rather general case when the perturbation set Z is given by a conic representation as follow where $\mathbf{K}$ is proper cone.

$$Z = \{\zeta \in \mathbf{R}^L : P\zeta + p \in \mathbf{K}\}$$

the uncertainty-affected constraint can be written as

$$
\begin{aligned}
&[a_0 + \sum_{\ell=1}^{L} \zeta_\ell a_\ell]^T x \le b_0 + \sum_{\ell=1}^{l} \zeta_\ell b_\ell \\
\Leftrightarrow &\max_{\zeta \in Z} \left\{ \sum_{\ell=1}^{L} \zeta_\ell [a_\ell^T x - b_\ell] \right\} \le b_0 - a_0^T x
\end{aligned}
$$

x is feasible if and only if the optimal value of the conic program is less than $b_0 - a_0^T x$.

$$\max_{\zeta} \left\{ \sum_{\ell=1}^{L} \zeta_\ell [a_\ell^T x - b_\ell] : P\zeta + p \in \mathbf{K} \right\}$$

The optimal value of the conic program is less than $b_0 - a_0^T x$ if and only if the conic dual problem as follow is attained and is less than $b_0 - a_0^T x$

$$
\begin{aligned}
\min \quad & p^T y \\
s.t. \quad & (P^T y)_\ell = -[a_\ell^T x - b_\ell] \quad \forall \ell = 1, \cdots, L \\
& y \in K_*
\end{aligned}
$$

Then the tractable representation of the robust counterpart model is as follow

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & p^T y + a_0^T x \le b_0 \\ & (P^T y)_\ell + a_\ell^T x = b_\ell \;\; \forall \ell = 1, \cdots, L \\ & y \in \mathbf{K}_* \end{aligned}$$

Further, consider the perturbation set Z is a intersection of multiple cone as follow.

$$Z = \{\zeta : P_s\zeta + p_s \in K^s, s = 1, \cdots, S\}$$

The tractable representation of the robust model is as follow

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & \sum_{s=1}^{S} p_s^T y^s + a_0^T x \le b_0 \\ & \sum_{s=1}^{S} (P_s^T y^s)_\ell + a_\ell^T x = b_\ell \;\; \forall \ell = 1, \cdots, L \\ & y^s \in \mathbf{K}_*^s, \forall s = 1, \cdots, S \end{aligned}$$

4.2 Example - Budgeted Uncertainty

Consider the case where perturbation set Z is the intersection of $\|\cdot\|_\infty$ and $\|\cdot\|_1$, specifically,

$$Z = \{\zeta \in \mathbf{R}^L : \|\zeta\|_\infty \le 1, \|\zeta\|_1 \le \gamma\}$$

where γ is a given "uncertainty budget". The perturbation set Z can be rewirtten as

$$Z = \{\zeta \in \mathbf{R}^L : P_1\zeta + p_1 \in K^1, P_2\zeta + p_2 \in K^2\}$$

where

$$\begin{aligned} &(1) P_1\zeta = [\zeta; 0],\ p_1 = [0_{L\times 1}; 1],\ K^1 = \{[u; t] \in \mathbf{R}^L \times \mathbf{R} : \|u\|_\infty \le t\} \\ &(2) P_2\zeta = [\zeta; 0],\ p_1 = [0_{L\times 1}; \gamma],\ K^2 = \{[u; t] \in \mathbf{R}^L \times \mathbf{R} : \|u\|_1 \le t\} \end{aligned}$$

Further, we have $K_*^1 = \{[u; t] \in \mathbf{R}^L \times \mathbf{R} : \|u\|_1 \le t\} = K^2$ and $K_*^2 = \{[u; t] \in \mathbf{R}^L \times \mathbf{R} : \|u\|_\infty \le t\} = K^1$.

Define the dual variable of sub dual problem as $y^1 = [u; \tau_1]$ and $y^2 = [v; \tau_2]$ where $u, v \in \mathbf{R}^L$ and $\tau_1, \tau_2 \in \mathbf{R}$. From the conclusion of general case, we can write the tractable representation of the robust model with budgeted uncertainty as follow

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & \tau_1 + \gamma\tau_2 + a_0^T x \le b_0 \\ & (u + v)_\ell = b_\ell - a_\ell^T x, \quad \forall \ell = 1, \cdots, L \\ & \|u\|_1 \le \tau_1 \\ & \|v\|_\infty \le \tau_2 \end{aligned}$$

Further, the above model can be simplified by eliminating τ_1 and τ_2.

$$\begin{aligned}
\min \quad & c^T x \\
s.t. \quad & \|u\|_1 + \gamma \|v\|_\infty + a_0^T x \le b_0 \\
& (u+v)_\ell = b_\ell - a_\ell^T x, \quad \forall \ell = 1, \cdots, L
\end{aligned}$$

5 Robust Approximations of Scalar Chance Constraints

5.1 Scalar Chance Constraints

We consider a scalar chance constraints as follow

$$Prob_{\zeta\sim P}\left\{\zeta : a_0^T x + \sum_{\ell=1}^{L} \zeta_\ell a_\ell^T x > b_0 + \sum_{\ell=1}^{L} \zeta_\ell b_\ell \right\} \le \epsilon$$

There is two assumption of the distribution of ζ as follow

- 1. ζ_ℓ,$\ell = 1, \cdots, L$ are independent random variables;
- 2. The distribution P_ℓ of the components ζ_ℓ are such that

  $$\int \exp\{t\zeta_\ell\} dP_\ell(\zeta_\ell) \le exp\{|\mu_\ell t| + \tfrac{1}{2}\sigma_\ell^2 t^2\} \quad \forall t \in R$$

  with known constant μ_ℓ and $\sigma_\ell \ge 0$

For convenience, let z_0 denote $z_0 = a_0^T x - b_0$ and z_ℓ denote $a_\ell^T x - b_\ell$ for $\ell = 1, \cdots, L$. Then, the chance constraints can be written as

$$p(z) = Prob_{\zeta\sim P}\left\{\zeta : z_0 + \textstyle\sum_{\ell=1} z_\ell \zeta_\ell > 0\right\} \le \epsilon$$

5.2 Robust Approximation

We will show how to get an approximation of scalar chance constraints. First, we can deal with a constraint violation as follows.

$$\begin{aligned}
& z_0 + \sum_{\ell=1}^{L} z_\ell \zeta_\ell > 0 \\
\Leftrightarrow & \exp\{\alpha[z_0 + \sum_{\ell=1}^{L} z_\ell \zeta_\ell]\} > 1 \quad \forall \alpha > 0 \\
\Rightarrow & \mathbf{E}\left\{\exp\{\alpha[z_0 + \sum_{\ell=1}^{L} z_\ell \zeta_\ell]\}\right\} \ge p(z) \quad \forall \alpha > 0 \\
\Rightarrow & \exp\{\alpha z_0 + \sum_{\ell=1}^{L}[\alpha|\mu_\ell z_\ell| + \frac{\alpha^2}{2}\sigma_\ell^2 z_\ell^2]\} \ge p(z) \quad \forall \alpha > 0 \\
\Leftrightarrow & \alpha z_0 + \sum_{\ell=1}^{L}[\alpha|\mu_\ell z_\ell| + \frac{\alpha^2}{2}\sigma_\ell^2 z_\ell^2] \ge \ln(p(z)) \quad \forall \alpha > 0
\end{aligned}$$

Therefore, we can say $p(z) \le \epsilon$ if the following inequality holds

$$\exists \alpha : \quad \alpha z_0 + \sum_{\ell=1}^{L} [\alpha|\mu_\ell z_\ell| + \tfrac{\alpha^2}{2}\sigma_\ell^2 z_\ell^2] \le \ln(\epsilon)$$

The left side of above is a quadratic function of α. Then the above inequality holds if and only if

$$z_0 + \sum_{\ell=1}^{L} |\mu_\ell z_\ell| + \sqrt{2\ln(1/\epsilon)}\sqrt{\sum_{\ell=1}^{L} \sigma_\ell^2 z_\ell^2} \le 0$$

In this way, we get a robust counterpart approximations of the chance constraints as above.

5.3 The interpretation

Now, we consider a simple distribution as follow

$$\begin{gathered} \mathbf{E}\{\zeta_\ell\} = 0 \\ |\zeta_\ell| \le 1 \quad \forall \ell = 1, \cdots, L \\ \{\zeta_\ell\} \textit{ are independent.} \end{gathered}$$

Then, the robust approximations can be written as follow which is equivalent to the ball uncertainty with $\Omega = \sqrt{2\ln(1/\epsilon)}$.

$$z_0 + \sqrt{2\ln(1/\epsilon)}\sqrt{\sum_{\ell=1}^{L} \sigma_\ell^2 z_\ell^2} \le 0$$

Further, we can have the follow inequality

$$StD(\sum_{\ell=1}^{L} z_\ell \zeta_\ell) = \sqrt{\sum_{\ell=1}^{L} \sigma_\ell^2 z_\ell^2}$$

The above approximation holds since

$$Prob\left\{\zeta : \sum_{\ell=1}^{L} z_\ell \zeta_\ell > \sqrt{2\ln(1/\epsilon)}\sqrt{\sum_{\ell=1}^{L} \sigma_\ell^2 z_\ell^2}\right\} \le \epsilon$$

That is, for a random variable satisfies the two assumptions, the probability that the value is greater than $\sqrt{2\ln(/\epsilon)}$ times variance is ϵ.

This form is equivalent to approximate the chance constraint with ball uncertainty. Next, we will show how to approximate the chance constraint with budgeted uncertainty.

5.4 Another Approximation - Budgeted Uncertainty

In the "General case", we get a robust corresponding constraints for budgetary uncertainty as follow.

$$\|u\|_1 + \gamma\|v\|_\infty + a_0^T x \le b_0 \tag{1}$$

$$(u+v)_\ell = b_\ell - a_\ell^T x, \quad \forall \ell = 1, \cdots, L \tag{2}$$

Then, we can prove the above satisfy $p(z) \le \exp(\frac{-\gamma^2}{2L})$.

Before that, we should have $\|v\|_2 \le \sqrt{L}\|v\|_\infty$ from the follow inequality

$$\|v\|_2^2 = \sum_{\ell=1}^{L} v_\ell^2 \leq \sum_{\ell=1}^{L} |v_\ell| \|v\|_\infty \leq \sqrt{L}\|v\|_2\|v\|_\infty$$

Proof:

$$\begin{aligned}
&\sum_{\ell=1}^{L} \zeta_\ell [a_\ell^T x - b_\ell] > b_0 - a_0^T x \\
\Rightarrow & -\sum_{\ell=1}^{L} u_\ell \zeta_\ell - \sum_{\ell=1}^{L} v_\ell \zeta_\ell > b_0 - a_0^T x \quad (from\ (2)) \\
\Rightarrow & \|u\|_1 - \sum_{\ell=1}^{L} v_\ell \zeta_\ell > b_0 - a_0^T x \\
\Rightarrow & -\sum_{\ell=1}^{L} v_\ell \zeta_\ell > b_0 - a_0^T x - \|u\|_1 \geq \gamma \|v\|_\infty \quad (from(1)) \\
\Rightarrow & -\sum_{\ell=1}^{L} v_\ell \zeta_\ell > \frac{\gamma}{\sqrt{L}} \|v\|_2 \Rightarrow -\sum_{\ell=1}^{L} v_\ell \zeta_\ell > StD(-\sum_{\ell=1}^{L} v_\ell \zeta_\ell) \\
\Rightarrow & Prob\{z_0 + \sum_{\ell=1}^{L} z_\ell \zeta_\ell > 0\} \leq \exp(\frac{-\gamma^2}{2L})
\end{aligned}$$

Therefore, the budgeted uncertainty set can also be used to approximate the chance constraint, and the probability of violation is less than $\exp(\frac{-\gamma^2}{2L})$.

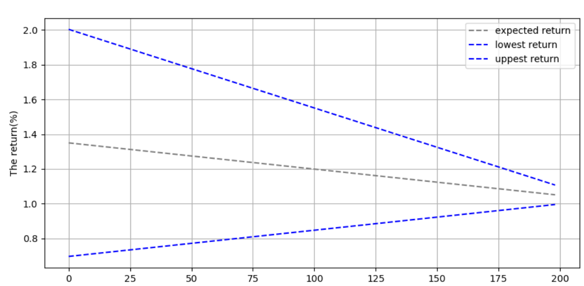
6 Numerical Experiments

6.1 A Single-Period Portfolio Selection

There are 200 assets. The 200th Asset has yearly return $r_{200} = 1.05$ and zero variability. The yearly returns r_ℓ, $i = 1, \cdots, 199$ of the remaining assets are independent random variables taking values in $[\mu_\ell - \sigma_\ell, \mu_\ell + \sigma_\ell]$ where

$$\mu_\ell = 1.05 + 0.3\frac{(200-i)}{199} \qquad and \qquad \sigma_\ell = 0.05 + 0.06\frac{(200-i)}{199}$$

The following figure shows the expected, uppest and lowest return of all assets, decreasing with the index. We want to solve the robust optimization problem as follow.



$$
\begin{aligned}
\max_{x,R} \quad & R \\
s.t. \quad & Prob_{r\in U}\{\sum_{\ell=1}^{199} r_\ell x_\ell + r_{200}x_{200} \le R\} \le \epsilon \\
& \sum_{\ell=1}^{200} x_\ell = 1 \\
& \mathbf{x} \succeq 0
\end{aligned}
$$

The perturbation set is $Z = \{\zeta :\in \mathbf{R}^1 99 : \|\zeta\|_\infty \le 1\}$. And The uncertain set U is as follow.

$$U = \{r : r_\ell = \mu_\ell + \sigma_\ell\zeta_\ell, \zeta_\ell \in [-1, 1]\}$$

6.2 Robust Counterpart Model

The box RC is as follow:

$$\max_{x,R} \left\{ R : \begin{array}{c} \sum_{\ell=1}^{199}(\mu_\ell - \sigma_\ell)x_\ell + r_{200}x_{200} \ge R \\ \sum_{\ell=1}^{200} x_\ell = 1 \\ \sum_{\ell=1}^{200} x_\ell = 1\mathbf{x} \succeq 0 \end{array} \right\}$$

The ball RC is as follow ($\Omega = \sqrt{2\ln(1/\epsilon)}$):

$$\max_{x,R,z,w} \left\{ R : \begin{array}{c} \sum_{\ell=1}^{199} \mu_\ell x_\ell + r_{200}x_{200} - \|z\|_1 - \sqrt{2\ln(1/\epsilon)}\sqrt{\sum_{\ell=1}^{L} w_\ell^2} \ge R \\ z_\ell + w_\ell = \sigma_\ell x_\ell \quad \forall \ell = 1, \cdots, 199 \\ \sum_{\ell=1}^{200} x_\ell = 1 \qquad \mathbf{x} \succeq 0 \end{array} \right\}$$

The budgeted RC is as follow ($\gamma = \sqrt{2L\ln(1/\epsilon)}$):

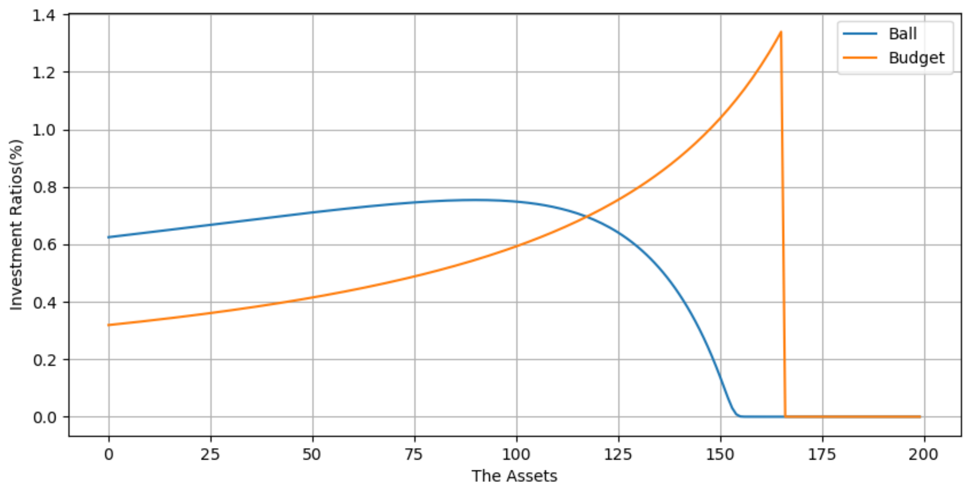
$$\max_{x,R,z,w} \left\{ R : \begin{array}{c} \sum_{\ell=1}^{199} \mu_\ell x_\ell + r_{200}x_{200} - \|z\|_1 - \sqrt{2L\ln(1/\epsilon)}\|w\|_\infty \ge R \\ z_\ell + w_\ell = \sigma_\ell x_\ell \quad \forall \ell = 1, \cdots, 199 \\ \sum_{\ell=1}^{200} x_\ell = 1 \qquad \mathbf{x} \succeq 0 \end{array} \right\}$$

6.3 Performance

Take $\epsilon = 0.005$, then the corresponding $\Omega = 3.255$ and $\gamma = 45.921$. The return of three models are $R_{box} = 1.05$, $R_{ball} = 1.12$ and $R_{budget} = 1.10$, respectively. The ball uncertainty have the highest return, the budgeted uncertainty is the second, and the box have the lowest return.

The solution of box is $x_{200} = 1$ and $x_\ell = 0$ for all $\ell = 1, \cdots, 199$. The solution of "ball RC" and "budget RC" is shown as follow. In the following figure, along x-axis: indices $1, 2, \cdots, 200$ of the assets; along y-axis: investment ratios. From the figure, we can see that ball uncertainty is more radical than budget uncertainty because more risky assets are invested; budget uncertainty, by contrast, is more conservative.

Finally, we simulate the solutions of the three models 10^6 times, and the number of constraint violations is 0. So all three solutions are robust



6.4 Comparison

To summarize the above analysis, we can compare the performance of the three models, as shown in the following table.

Conservatism	*Box* > *Budget* > *Ball*
Risk	*Ball* = *Budget* > *Box*
Return	*Ball* > *Budget* > *Box*

Although the box uncertainty can guarantee 100% robustness, but at the cost of low returns. Ball uncertainty is the best option, but its disadvantage is the nonlinear model. Budgeted uncertainty is a trade-off, preserving linear model with linear model, low conservatism, and high returns.

7 Appidex

A The code of box uncertainty

```
import gurobipy as gp
from gurobipy import GRB
import numpy as np


def box():
    ### Compute the parameters
    mu = np.array([1.05 + 0.3 * ((200 - i - 1) / 199) for i in range(199)])
    sigma = np.array([0.05 + 0.6 * ((200 - i) / 199) for i in range(199)])

    ### Generate a model
    m = gp.Model('Box')
    m.setAttr('ModelSense', GRB.MAXIMIZE)
    m.Params.OutputFlag = 0

    ### Add variables
    x = []
    for i in range(200):
        x.append(m.addVar(lb=0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="x
            {}".format(i)))
    R = m.addVar(lb=0, ub=GRB.INFINITY, obj=1, vtype=GRB.CONTINUOUS, name='R')

    ### Add Constraints
    cons = []
    cons.append(m.addConstr(gp.quicksum([(mu[i] - sigma[i]) * x[i] for i in
        range(199)]) + 1.05 * x[199] >= R,
                                name='RC1_Constraint'))
    cons.append(m.addConstr(gp.quicksum(x[:200]) == 1, name='RC2_Constraint'))

    ### Solve the model
    m.update()
    m.optimize()
    R = R.X
    x = [i.X for i in x]

    return [R, x]


R, x = box()
```

B The code of ball uncertainty

```
import gurobipy as gp
from gurobipy import GRB
import numpy as np
import math


def ball(e):
    ### Compute the parameters
    Omega = math.sqrt(2 * math.log(1 / e, math.e))
    mu = np.array([1.05 + 0.3 * ((200 - i - 1) / 199) for i in range(199)])
    sigma = np.array([0.05 + 0.6 * ((200 - i) / 199) for i in range(199)])

    ### Generate a model
    m = gp.Model('Box')
    m.setAttr('ModelSense', GRB.MAXIMIZE)
    m.Params.OutputFlag = 0

```

```
    ### Add variables
    x = []
    z = []
    absz = []
    w = []
    R = m.addVar(0, GRB.INFINITY, 1, vtype=GRB.CONTINUOUS, name='R')
    sqrtw = m.addVar(0, GRB.INFINITY, vtype=GRB.CONTINUOUS, name='Sqrtw')
    for i in range(199):
        x.append(m.addVar(lb=0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name='x
            {}'.format(i)))
        z.append(m.addVar(vtype=GRB.CONTINUOUS, name='z{}'.format(i)))
        absz.append(m.addVar(vtype=GRB.CONTINUOUS, name='absz{}'.format(i)))
        w.append(m.addVar(vtype=GRB.CONTINUOUS, name='w{}'.format(i)))
    x.append(m.addVar(lb=0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name='x199'
        ))

    ### Add Constraints
    cons = []
    cons.append(m.addConstr(gp.quicksum([i * i for i in w]) <= sqrtw * sqrtw,
        name='Sqrt_W'))
    for i in range(199):
        cons.append(m.addConstr(z[i] <= absz[i], name='Abs1_z{}'.format(i)))
        cons.append(m.addConstr(z[i] >= -absz[i], name='Abs2_z{}'.format(i)))
        cons.append(m.addConstr(z[i] + w[i] == sigma[i] * x[i], name='
            RC_Constraints{}'.format(i)))
    cons.append(m.addConstr(gp.quicksum(x) == 1, name='Money_Constraint'))
    cons.append(m.addConstr(
        gp.quicksum([mu[i] * x[i] for i in range(199)]) + 1.05 * x[199] - gp.
            quicksum(absz) - Omega * sqrtw >= R,
        name='RC2_Constraints'))

    ### Solve the model
    m.update()
    m.optimize()
    x = [i.x for i in x]
    R = R.x

    return [R, x]


R, x = ball(0.005)
```

C The code of budgeted uncertainty

```
import gurobipy as gp
import math
from gurobipy import GRB
import numpy as np


def budget(e):
    ### Compute the parameters
    gamma = math.sqrt(-2 * math.log(e, math.e) * 199)
    mu = np.array([1.05 + 0.3 * ((200 - i - 1) / 199) for i in range(199)])
    sigma = np.array([0.05 + 0.6 * ((200 - i) / 199) for i in range(199)])

    ### Generate a model
    m = gp.Model('Box')
    m.setAttr('ModelSense', GRB.MAXIMIZE)
    m.Params.OutputFlag = 0

    ### Add variables
    x = []
    z = []
    absz = []
```

```
    w = []
    R = m.addVar(0, GRB.INFINITY, 1, vtype=GRB.CONTINUOUS, name='R')
    maxw = m.addVar(0, GRB.INFINITY, vtype=GRB.CONTINUOUS, name='Maxw')
    for i in range(199):
        x.append(m.addVar(lb=0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name='x{}'.format(i)))
        z.append(m.addVar(vtype=GRB.CONTINUOUS, name='z{}'.format(i)))
        absz.append(m.addVar(vtype=GRB.CONTINUOUS, name='absz{}'.format(i)))
        w.append(m.addVar(vtype=GRB.CONTINUOUS, name='w{}'.format(i)))
    x.append(m.addVar(lb=0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name='x199'))

    ### Add Constraints
    cons = []
    for i in range(199):
        cons.append(m.addConstr(z[i] + w[i] == sigma[i] * x[i], name='RC_Constraints{}'.format(i)))
        cons.append(m.addConstr(z[i] <= absz[i], name='Abs1_z{}'.format(i)))
        cons.append(m.addConstr(z[i] >= -absz[i], name='Abs2_z{}'.format(i)))
        cons.append(m.addConstr(w[i] <= maxw, name='Max_w{}'.format(i)))
    cons.append(m.addConstr(gp.quicksum(x) == 1, name='Money_Constraint'))
    cons.append(m.addConstr(
        gp.quicksum([mu[i] * x[i] for i in range(199)]) + 1.05 * x[199] - gp.quicksum(absz) - gamma * maxw >= R,
        name='RC2_Constraints'))

    ### Solve the model
    m.update()
    m.optimize()
    x = [i.x for i in x]
    R = R.x

    return [R, x]


R, x = budget(0.005)
```

D The code of simulation

```
from box import box
from ball import ball
from budget import budget
import matplotlib.pyplot as plt
import numpy as np

### Simulation Parameters
epsilon = 0.005
itera = 1000000

### Compute the expected,lowest and uppest returns of all assets; Plot the figure
mu = np.array([1.05 + 0.3 * ((200 - i - 1) / 199) for i in range(199)])
sigma = np.array([0.05 + 0.6 * ((200 - i) / 199) for i in range(199)])
plt.subplot(2, 1, 1)
plt.plot(mu, color='grey', linestyle='--', label='expected return')
plt.plot(mu - sigma, color='b', linestyle='--', label='lowest return')
plt.plot(mu + sigma, color='b', linestyle='--', label='uppest return')
plt.grid()
plt.xlabel('The Assets')
plt.ylabel('The return(%)')
plt.legend()

#### Solve the three models
R1, x1 = box()
R2, x2 = ball(epsilon)
```

```python
R3, x3 = budget(epsilon)
# x1=[i*100 for i in x1]
x2 = [i * 100 for i in x2]
x3 = [i * 100 for i in x3]
# plt.plot(x1,label='Box')
plt.subplot(2, 1, 2)
plt.plot(x2, label='Ball')
plt.plot(x3, label='Budget')
plt.grid()
plt.xlabel('The Assets')
plt.ylabel('Investment Ratios(%)')
plt.legend()

### Simulation
P1, P2, P3 = [], [], []
for i in range(itera):
    zeta = np.random.random(199) * 2 - 1
    P1.append(1 if (np.dot(mu + zeta * sigma, x1[:199]) + 1.05 * x1[-1] >= R1) else 0)
    P2.append(1 if (np.dot(mu + zeta * sigma, x2[:199]) + 1.05 * x2[-1] >= R2) else 0)
    P3.append(1 if (np.dot(mu + zeta * sigma, x3[:199]) + 1.05 * x3[-1] >= R3) else 0)
P1 = np.array(P1)
P2 = np.array(P2)
P3 = np.array(P3)
print("\n Box\t: {}\n Ball\t: {}\n Budget\t: {}".format(P1.mean(), P2.mean(), P3.mean()))
```