A short introduction to robust optimization

Erick Delage
Dept. of decision sciences, HEC Montréal

(these slides)

HEC MONTRÉAL

A production problem

- A company produces two kinds of drugs, DrugI and DrugII, containing a specific agent A, which is extracted from raw materials purchased on the market. Its goal is to find the production plan that maximizes the profit of the company.

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

Linear programming formulation (see Excel file)

$$
\begin{aligned}
\underset{\text{RI,RII,DI,DII}}{\text{maximize}} \quad & 6200\text{DI} + 6900\text{DII} - (100\text{RI} + 199.90\text{RII} + 700\text{DI} + 800\text{DII}) \\
\text{subject to} \quad & \text{RI} + \text{RII} \leq 1000 && \text{(Storage)} \\
& 90\text{DI} + 100\text{DII} \leq 2000 && \text{(Manpower)} \\
& 40\text{DI} + 50\text{DII} \leq 800 && \text{(Equipment)} \\
& 100\text{RI} + 199.9\text{RII} + 700\text{DI} + 800\text{DII} \leq 100000 && \text{(Budget)} \\
& 0.01\text{RI} + 0.02\text{RII} - 0.5\text{DI} - 0.6\text{DII} \geq 0 && \text{(Agent A)} \\
& \text{RI} \geq 0, \text{RII} \geq 0, \text{DI} \geq 0, \text{DII} \geq 0,
\end{aligned}
$$

- Optimal solution :
  RI* = 0, RII* = 438 kg, DI* = 17 552 packs, DII* = 0 pack, optimal profit = 8820$
- Optimal solution relies heavily on extracting 8,78 g of agent A and using all of it in the production of Drug I (no left-over).
- If there is a 2% estimation error in conversion rate for this raw material, then we are missing 0,18g of agent A and can only produce 17201 packs of Drug I. Profit drops to 6889$ (i.e. -22%)

Robust formulation

- If there is a 2% estimation error in conversion rate for Raw II and 0.5% for Raw I, then it would be more robust to replace the (Agent A) constraint with:

$$(0.995 \cdot 0.01)\mathrm{RI} + (0.98 \cdot 0.02)\mathrm{RII} - 0.5\mathrm{DI} - 0.6\mathrm{DII} \geq 0$$

- The optimal solution would then be :
  RI* = 878 kg, RII* = 0 kg, DI* = 17 467 packs, DII* = 0 pack, guaranteed profit = 8295$ (-6%)
- This solution can be considered immuned to estimation error of the conversion rates of Raw I and Raw II

Generalized need for robust optimization

Ben-Tal & Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming, Series A, 88(3):411–424, 2000.

« small perturbations of obviously uncertain data coefficients can make the nominal optimal solution heavily infeasible and thus practically meaningless »

- Nbad : number of constraints where there is more than 2% chance that violation is > 5%
- Index: maximum violation in % among all constraints

Problem	Size[a]	$\epsilon = 0.01\%$		$\epsilon = 1\%$	
		Nbad[b]	Index[c]	Nbad	Index
80BAU3B	2263 × 9799	37	84	364	8,420
25FV47	822 × 1571	14	16	35	1,620
ADLITTLE	57 × 97			7	58
AFIRO	28 × 32			2	50
BNL2	2325 × 3489			24	34
BRANDY	221 × 249			1	5
CAPRI	272 × 353			14	390
CYCLE	1904 × 2857	2	110	6	11,000
D2Q06C	2172 × 5167	107	1,150	168	115,000
E226	224 × 282			2	15
FFFFF800	525 × 854			6	8
FINNIS	498 × 614	12	10	97	1,040
GREENBEA	2393 × 5405	13	116	37	11,600
KB2	44 × 41	5	27	10	2,680
MAROS	847 × 1443	3	6	73	566
NESM	751 × 2923			37	20
PEROLD	626 × 1376	6	34	58	3,390
PILOT	1442 × 3652	16	50	379	4,980
PILOT4	411 × 1000	42	210,000	75	21,000,000
PILOT87	2031 × 4883	86	130	990	13,000
PILOTJA	941 × 1988	4	46	59	4,630
PILOTNOV	976 × 2172	4	69	47	6,940
PILOTWE	723 × 2789	61	12,200	69	1,220,000
SCFXM1	331 × 457	1	95	11	9,460
SCFXM2	661 × 914	2	95	21	9,460
SCFXM3	991 × 1371	3	95	32	9,460
SHARE1B	118 × 225	1	257	1	25,700

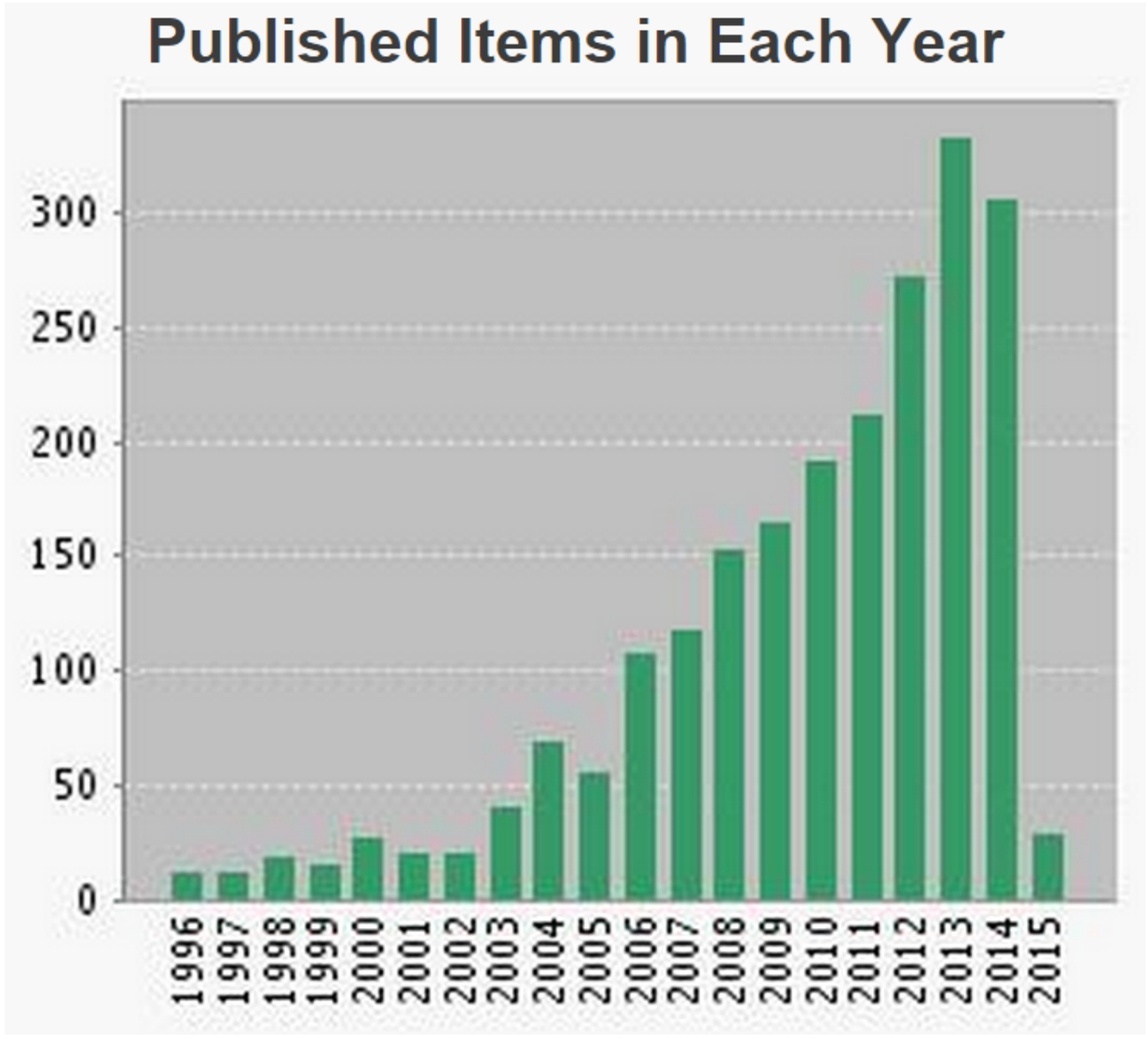
« In applications of LP, there exists a real need of a technique capable of detecting cases when data uncertainty can heavily affect the quality of the nominal solution, and in these cases to generate a reliable solution, i.e. one which is immuned against uncertainty. »

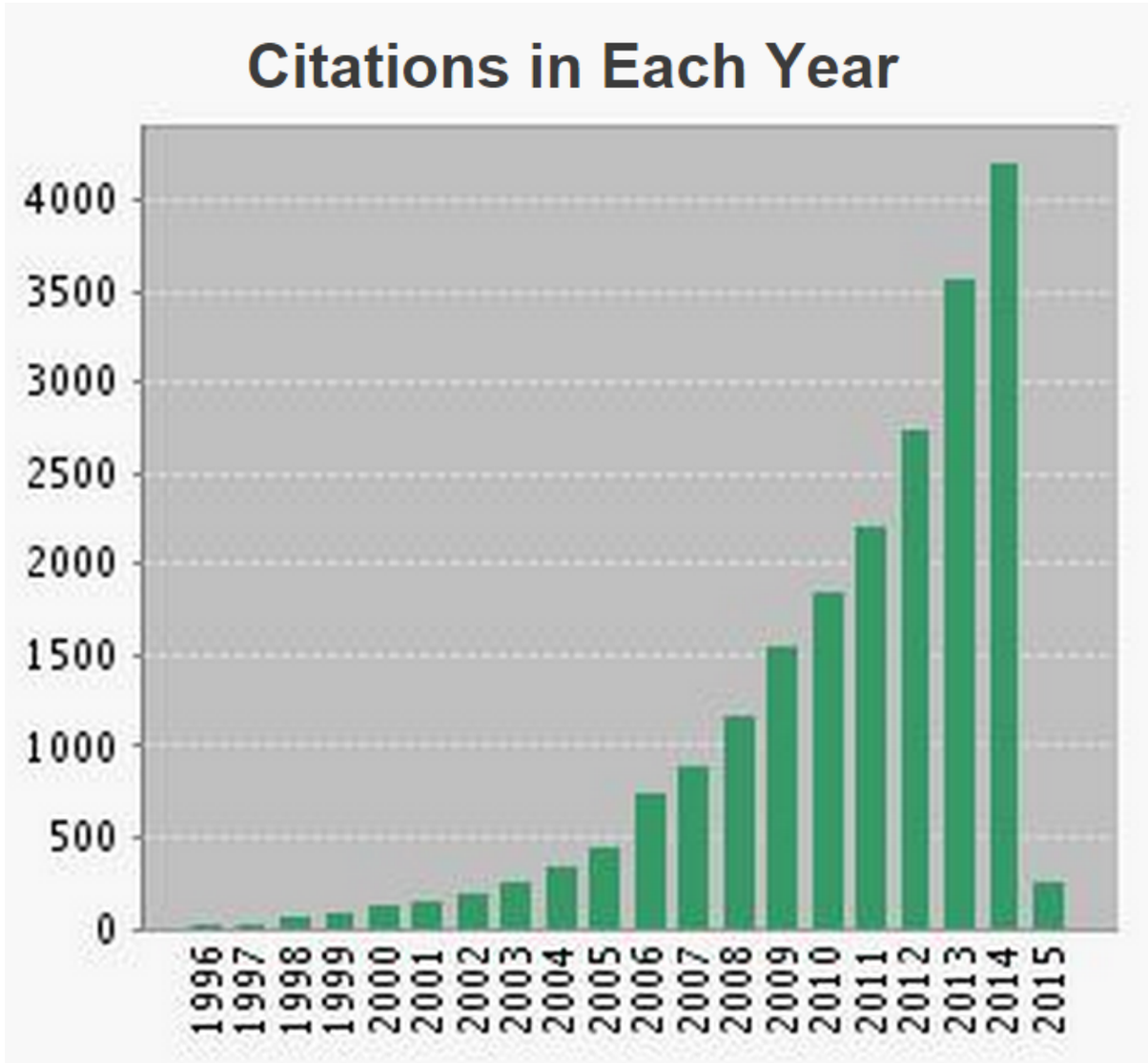
-Ben-Tal & Nemirovski, 2000

History of robust optimization

- « maximin » introduced by Abraham Wald in 1945 to manage risk and is influenced by zero-sum games (von Neumann & Morgenstern, 1944)
- Allen Soyster (1973) proposed robustifying constraints of a linear program by considering interval uncertainty.
- Idea was considered too conservative for a long time
- In 1998, Ben-Tal & Nemirovski gave a second life to the framework in context of convex optimization by exploiting ellipsoidal uncertainty

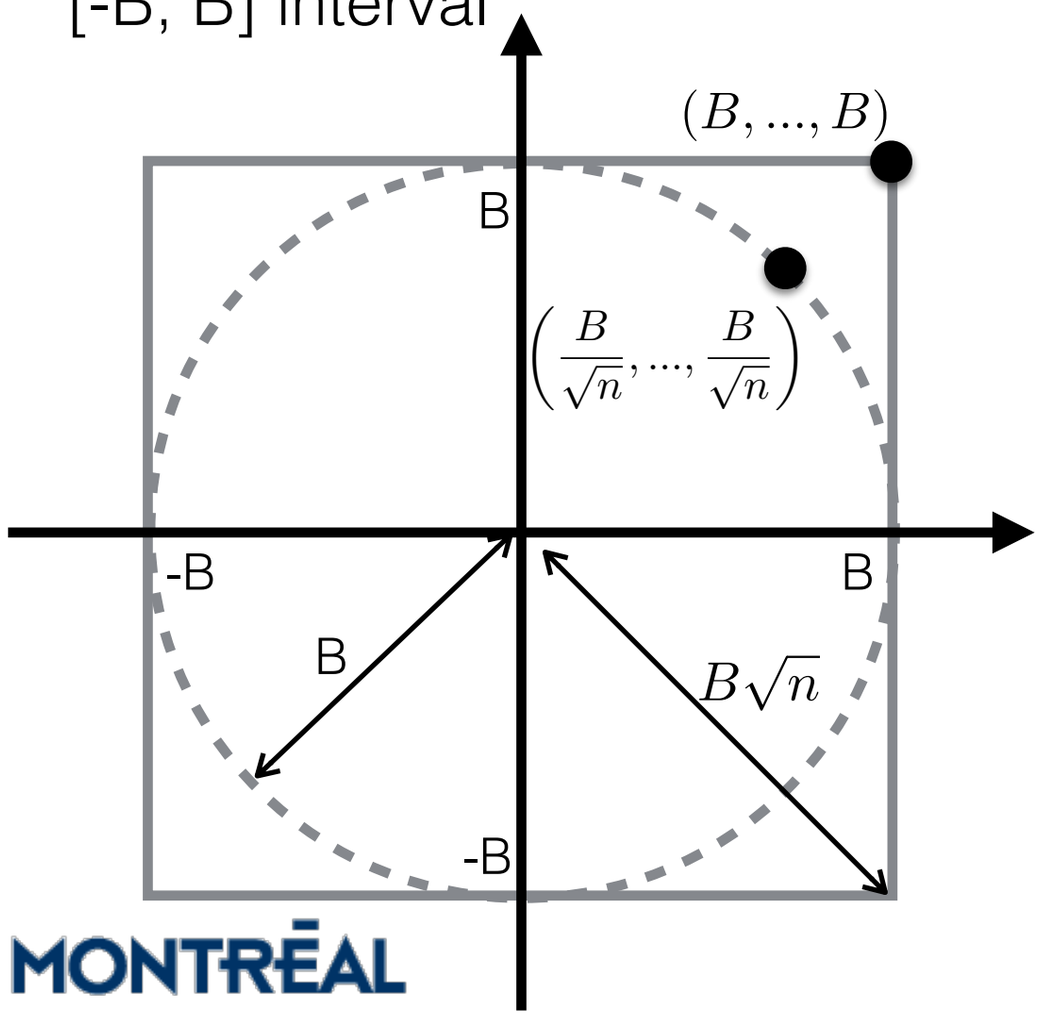
Rebirth of robust optimization



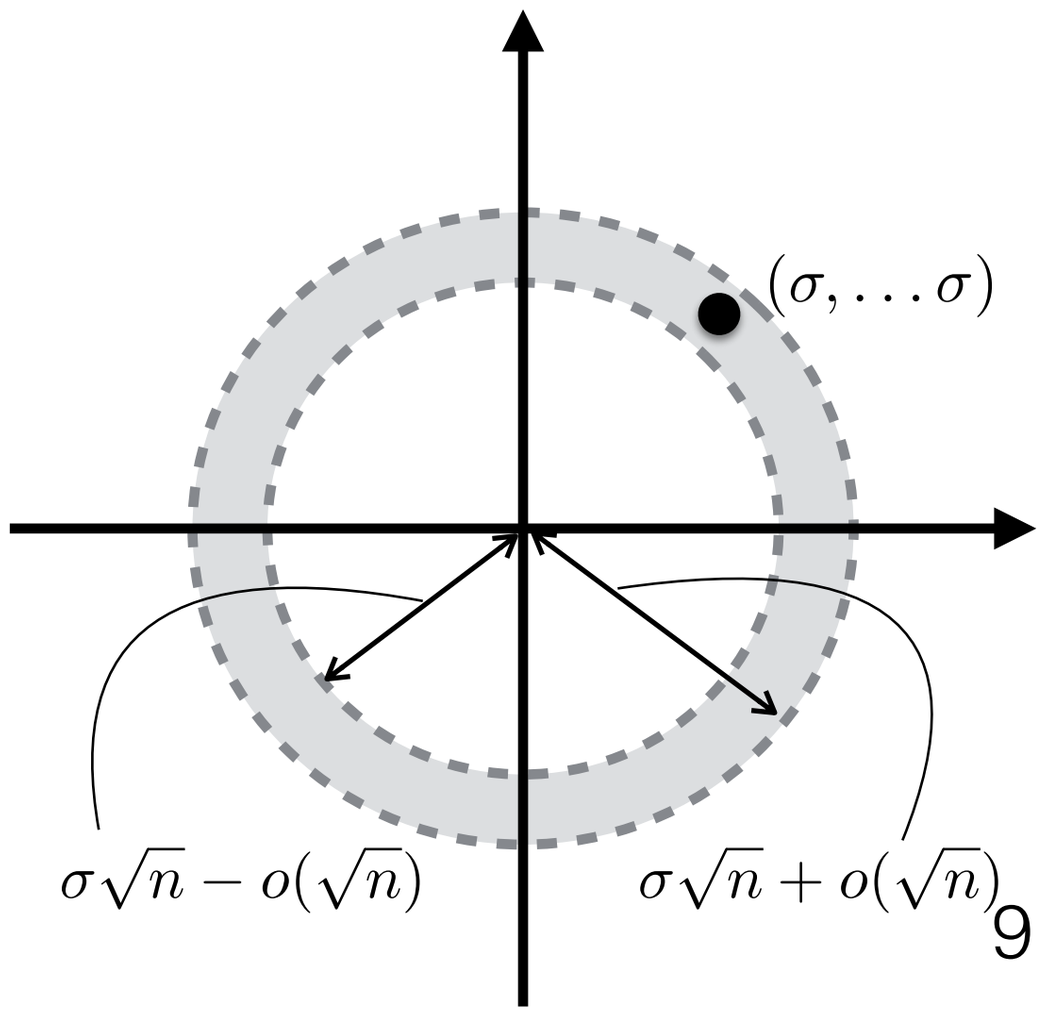


Why ellipsoidal sets are less conservative than boxes?

- In high dimension, a point in the box can be up to $\sqrt{n}$ times further from zero than any point in the ellipsoid that covers each [-B, B] interval



- In high dimension, vectors composed of i.i.d. components lie with high probability in the following hyper-orbit



Possible reasons for rebirth of robust optimization

- Characterizing uncertainty with a set of possible realization can be easier than with a multivariate distribution
- Some uncertainty sets have the following properties:
  - they reduce conservativeness
  - they lead to reformulations that are easier to solve than SP
  - they provide statistical guarantees.
- The development of more powerful computers and in particular of fast interior point methods.

Defining robust optimization

$$\text{(Nominal problem)} \quad \begin{array}{ll} \underset{x}{\text{maximize}} & h(x,z) \\ \text{subject to} & g_j(x,z) \leq 0\,,\, \forall\, j = 1,...,J\ , \end{array}$$

$$\text{(Robust counterpart)} \quad \begin{array}{ll} \underset{x}{\text{maximize}} & \min\limits_{z \in \mathcal{Z}} \ h(x,z) \\ \text{subject to} & g_j(x,z) \leq 0\,,\, \forall\, z \in \mathcal{Z},\ ,\, \forall\, j = 1,...,J\ . \end{array}$$

Reformulating a robust optimization problem using duality theory

- Consider the robust LP

$$\mathcal{RLP} := \underset{x\in\mathcal{X}}{\text{maximize}} \quad \left\{ \min_{z:Pz\leq q} \; c(x)^\top z + d(x) \right\} ,$$

- Reformulating inner problem using Lagrangian

$$\mathcal{RLP} \equiv \quad \underset{x\in\mathcal{X}}{\text{maximize}} \quad \left\{ \min_{z} \; \max_{\lambda\geq 0} \; c(x)^\top z + d(x) - \lambda^\top (q - Pz) \right\}$$

- LP strong duality

$$\mathcal{RLP} \equiv \quad \underset{x\in\mathcal{X}}{\text{maximize}} \quad \left\{ \max_{\lambda\geq 0} \; \min_{z} \; c(x)^\top z + d(x) - \lambda^\top (q - Pz) \right\}$$

- Proper book-keeping leads to compact reformulation

$$\begin{aligned} \mathcal{RLP} \equiv \quad \underset{x\in\mathcal{X},\lambda\geq 0}{\text{maximize}} \quad & d(x) - q^\top \lambda \,, \\ \text{subject to} \quad & P^\top \lambda + c(x) = 0 \end{aligned}$$

Robust production problem

$$
\begin{aligned}
\underset{\mathrm{RI,RII,DI,DII}}{\text{maximize}} \quad & 6200\mathrm{DI} + 6900\mathrm{DII} - (100\mathrm{RI} + 199.90\mathrm{RII} + 700\mathrm{DI} + 800\mathrm{DII}) \\
\text{subject to} \quad & \mathrm{RI} + \mathrm{RII} \leq 1000 \\
& 90\mathrm{DI} + 100\mathrm{DII} \leq 2000 \\
& 40\mathrm{DI} + 50\mathrm{DII} \leq 800 \\
& 100\mathrm{RI} + 199.9\mathrm{RII} + 700\mathrm{DI} + 800\mathrm{DII} \leq 100000 \\
& 0.01(1 + 0.005z_1)\mathrm{RI} + 0.02(1 + 0.02z_2)\mathrm{RII} - 0.5\mathrm{DI} - 0.6\mathrm{DII} \geq 0 \ , \forall\, z \in \mathcal{Z} \\
& \mathrm{RI} \geq 0, \mathrm{RII} \geq 0, \mathrm{DI} \geq 0, \mathrm{DII} \geq 0,
\end{aligned}
$$

- Box uncertainty would assume:

$$\mathcal{Z} := \{z \in \mathbb{R}^2 \mid \max_i |z_i| \leq 1\}$$

- Ellipsoidal uncertainty would assume:

$$\mathcal{Z} := \{z \in \mathbb{R}^2 \mid \sum_i z_i^2 \leq 1\}$$

- Budgeted uncertainty would assume:

$$\mathcal{Z}(\Gamma) := \{z \in \mathbb{R}^2 \mid \max_i |z_i| \leq 1\,,\ \sum_i |z_i| \leq \Gamma\}$$

Robust production problem

$$
\begin{aligned}
\underset{\mathrm{RI,RII,DI,DII}}{\text{maximize}} \quad & 6200\mathrm{DI} + 6900\mathrm{DII} - (100\mathrm{RI} + 199.90\mathrm{RII} + 700\mathrm{DI} + 800\mathrm{DII}) \\
\text{subject to} \quad & \mathrm{RI} + \mathrm{RII} \leq 1000 \\
& 90\mathrm{DI} + 100\mathrm{DII} \leq 2000 \\
& 40\mathrm{DI} + 50\mathrm{DII} \leq 800 \\
& 100\mathrm{RI} + 199.9\mathrm{RII} + 700\mathrm{DI} + 800\mathrm{DII} \leq 100000 \\
& 0.01(1 + 0.005z_1)\mathrm{RI} + 0.02(1 + 0.02z_2)\mathrm{RII} - 0.5\mathrm{DI} - 0.6\mathrm{DII} \geq 0 \ , \forall\, z \in \mathcal{Z} \\
& \mathrm{RI} \geq 0, \mathrm{RII} \geq 0, \mathrm{DI} \geq 0, \mathrm{DII} \geq 0,
\end{aligned}
$$

- Box uncertainty leads to :

$$0.01\mathrm{RI} + 0.02\mathrm{RII} - 0.5\mathrm{DI} - 0.6\mathrm{DII} \geq 0.00005|\mathrm{RI}| + 0.0004|\mathrm{RII}|$$

- Ellipsoidal uncertainty leads to :

$$0.01\mathrm{RI} + 0.02\mathrm{RII} - 0.5\mathrm{DI} - 0.6\mathrm{DII} \geq \sqrt{0.00005^2\mathrm{RI}^2 + 0.0004^2\mathrm{RII}^2}$$

- Budgeted uncertainty leads to (with new « t » variable):

$$0.01\mathrm{RI} + \ldots - 0.6\mathrm{DII} \geq \Gamma t + \max(0; |0.00005\mathrm{RI}| - t) + \max(0; |0.0004\mathrm{RII}| - t)$$

RSOME implementation (see Google Colab)

Chen et al. Robust Stochastic Optimization Made Easy with RSOME. MS, 2020.

```
#Create model
model = ro.Model('production')

# Define decision variables
DrugI =model.dvar()              # Define a decision variable DrugI
DrugII =model.dvar()             # Define a decision variable DrugII
RawI =model.dvar()               # Define a decision variable RawI
RawII =model.dvar()              # Define a decision variable RawII

#Objective to maximize the profit
model.max(6200*DrugI+6900*DrugII - (100*RawI+199.90*RawII+700*DrugI+800*DrugII))

# Storage constraint
model.st(RawI + RawII <= 1000)

# Manpower constraint
model.st(90*DrugI + 100*DrugII <= 2000)

# Equipment constraint
model.st(40*DrugI+50*DrugII <= 800)

# Budget constraint
model.st(100*RawI+199.9*RawII+700*DrugI+800*DrugII <= 100000)

# Constraint to have enough active agent A
model.st(0.01*RawI+0.02*RawII-0.5*DrugI-0.6*DrugII >= 0)

# Constraints that decision variables are non-negative
model.st(DrugI >= 0)                                  # DrugI is non-negative
model.st(DrugII >= 0)                                 # DrugII is non-negative
model.st(RawI >= 0)                                   # RawI is non-negative
model.st(RawII >= 0)                                  # RawII is non-negative

# solve the model
model.solve(my_solver)
```

RSOME implementation (see Google Colab)

Chen et al. Robust Stochastic Optimization Made Easy with RSOME. MS, 2020.

```
#Create model
model = ro.Model('robustproduction_box')

# Define decision variables
DrugI =model.dvar()              # Define a decision variable DrugI
DrugII =model.dvar()             # Define a decision variable DrugII
RawI =model.dvar()               # Define a decision variable RawI
RawII =model.dvar()              # Define a decision variable RawII

# Declare uncertain parameters and the box uncertainty set
z = model.rvar(2)  # Random variables
boxSet = (abs(z) <= 1);  #each parameter is between [-1, 1]

# Objective to maximize the profit
model.max(6200*DrugI+6900*DrugII - (100*RawI+199.90*RawII+700*DrugI+800*DrugII))

# Constraints
model.st(RawI + RawII <= 1000) # Storage constraint
model.st(90*DrugI + 100*DrugII <= 2000) # Manpower constraint
model.st(40*DrugI+50*DrugII <= 800) # Equipment constraint
model.st(100*RawI+199.9*RawII+700*DrugI+800*DrugII <= 100000) # Budget constraint

# Constraints that decision variables are non-negative
model.st(DrugI >= 0)                             # DrugI is non-negative
model.st(DrugII >= 0)                            # DrugII is non-negative
model.st(RawI >= 0)                              # RawI is non-negative
model.st(RawII >= 0)                             # RawII is non-negative

# Constraint to have enough active agent A
model.st((0.01*(1+0.005*z[0])*RawI+0.02*(1+0.02*z[1])*RawII-0.5*DrugI-0.6*DrugII >= 0).forall(boxSet))

# solve the model
model.solve(my_solver)
```

Box uncertainty set

RSOME implementation (see Google Colab)

Chen et al. Robust Stochastic Optimization Made Easy with RSOME. MS, 2020.

```
#Create model
model = ro.Model('robustproduction_ellipsoidal')

# Define decision variables
DrugI =model.dvar()              # Define a decision variable DrugI
DrugII =model.dvar()             # Define a decision variable DrugII
RawI =model.dvar()               # Define a decision variable RawI
RawII =model.dvar()              # Define a decision variable RawII

# Declare uncertain parameters
z = model.rvar(2)
ellipsoidalSet= (rso.norm(z,2)<=1) #the perturbations are within a ball of radius 1

# Objective to maximize the profit
model.max(6200*DrugI+6900*DrugII -(100*RawI+199.90*RawII+700*DrugI+800*DrugII))

# Constraints
model.st(RawI + RawII <= 1000) # Storage constraint
model.st(90*DrugI + 100*DrugII <= 2000) # Manpower constraint
model.st(40*DrugI+50*DrugII <= 800) # Equipment constraint
model.st(100*RawI+199.9*RawII+700*DrugI+800*DrugII <= 100000) # Budget constraint

# Constraints that decision variables are non-negative
model.st(DrugI >= 0)                          # DrugI is non-negative
model.st(DrugII >= 0)                         # DrugII is non-negative
model.st(RawI >= 0)                           # RawI is non-negative
model.st(RawII >= 0)                          # RawII is non-negative

# Constraint to have enough active agent A
model.st((0.01*(1+0.005*z[0])*RawI+0.02*(1+0.02*z[1])*RawII-0.5*DrugI-0.6*DrugII >= 0).forall(ellipsoidalSet))

# solve the model
model.solve(my_solver)
```

Ellipsoidal uncertainty set

RSOME implementation (see Google Colab)

Chen et al. Robust Stochastic Optimization Made Easy with RSOME. MS, 2020.

```
#Create model
model = ro.Model('robustproduction_budgeted')

# Define decision variables
DrugI =model.dvar()              # Define a decision variable DrugI
DrugII =model.dvar()             # Define a decision variable DrugII
RawI =model.dvar()               # Define a decision variable RawI
RawII =model.dvar()              # Define a decision variable RawII

# Declare uncertain parameters
z = model.rvar(2)
budgetedSet=(abs(z) <= 1, rso.norm(z, 1) <=1)  # define the budgeted uncertainty set

# Objective to maximize the profit
model.max(6200*DrugI+6900*DrugII - (100*RawI+199.90*RawII+700*DrugI+800*DrugII))

# Constraints
model.st(RawI + RawII <= 1000) # Storage constraint
model.st(90*DrugI + 100*DrugII <= 2000) # Manpower constraint
model.st(40*DrugI+50*DrugII <= 800) # Equipment constraint
model.st(100*RawI+199.9*RawII+700*DrugI+800*DrugII <= 100000) # Budget constraint

# Constraints that decision variables are non-negative
model.st(DrugI >= 0)                          # DrugI is non-negative
model.st(DrugII >= 0)                         # DrugII is non-negative
model.st(RawI >= 0)                           # RawI is non-negative
model.st(RawII >= 0)                          # RawII is non-negative

# Constraint to have enough active agent A
model.st((0.01*(1+0.005*z[0])*RawI+0.02*(1+0.02*z[1])*RawII-0.5*DrugI-0.6*DrugII >= 0).forall(budgetedSet))

# solve the model
model.solve(my_solver)
```

Budgeted uncertainty set

Robust portfolio selection

- Consider the following portfolio selection problem:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{maximize}} \quad & \sum_{i=1}^{n} r_i x_i \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = 100\% \\ & x_i \geq 0\,, \forall\, i = 1, ..., n\,, \end{aligned}$$

- If returns follow a distribution then obj. becomes $\sum_{i=1}^{n} \mathbb{E}[r_i] x_i$
- If distribution of returns is unknown then one might use

$$\min_{r\in\mathcal{U}} \sum_{i=1}^{n} r_i x_i$$

Robust portfolio selection

- Assume that

$$r_i = \mu_i + \sigma_i z_i \ , \ z_i \in [-1, 1]$$

with

$$\mu_i := 0.15 + i\frac{0.05}{150}$$

$$\sigma_i := \frac{0.05}{450}\sqrt{2in(n+1)}$$

- What is robust solution?
- What is robust solution if only four (4) assets can deviate from expected value?

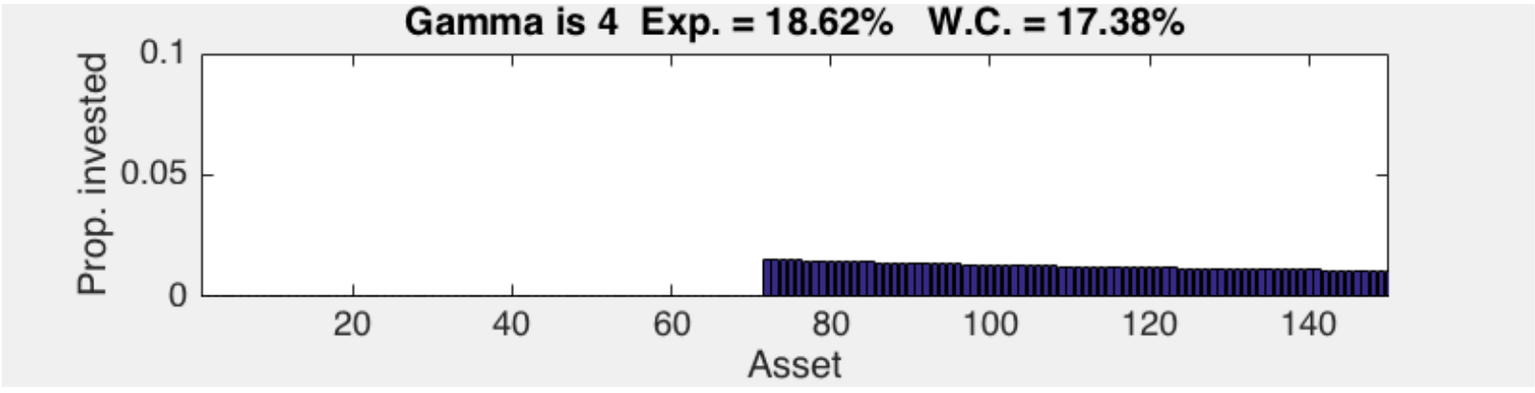
Asset	Interval	Expected
#1	[12.67%, 17.40%]	15.03 %
#2	[11.72%, 18.41%]	15.06 %
#3	[11.00%, 19.20%]	15.10 %
#4	[10.40%, 19.86%]	15.13 %
...	...	...
#147	[-8.77%, 48.57%]	19.90 %
#148	[-8.84%, 48.70%]	19.93 %
#149	[-8.90%, 48.83%]	19.97 %
#150	[-8.96%, 48.96%]	20.00 %

Robust portfolio selection

(see Google Colab)

- If only four assets can deviate from expected value, then robust portfolio has expected return of 18.62% while worst-case is 17.38%.

- The robust invested proportions are: (see animation url)



Asset	Interval	Expected
#1	[12.67%, 17.40%]	15.03 %
#2	[11.72%, 18.41%]	15.06 %
#3	[11.00%, 19.20%]	15.10 %
#4	[10.40%, 19.86%]	15.13 %
...	...	...
#147	[-8.77%, 48.57%]	19.90 %
#148	[-8.84%, 48.70%]	19.93 %
#149	[-8.90%, 48.83%]	19.97 %
#150	[-8.96%, 48.96%]	20.00 %

Distributionally robust optimization

- Robust optimization can also be applied on top of a stochastic program when one is not comfortable about making a distribution assumption:

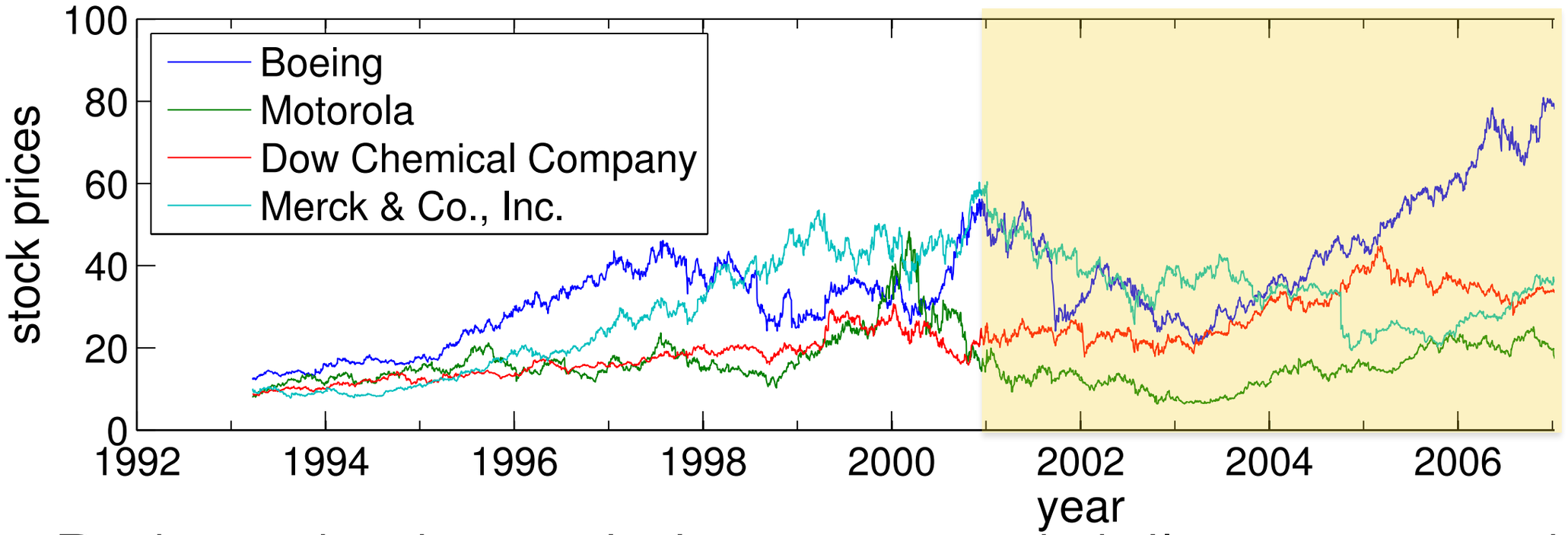
$$\underset{x \in \mathcal{X}}{\text{maximize}} \quad \underset{F \in \mathcal{D}}{\min} \; \mathbb{E}_F[h(x, z)]$$

  where « D » is the set of all plausible distributions for « z ».

- For example, one might consider the following distribution set:

$$\mathcal{D} := \left\{ F \;\middle|\; \begin{array}{l} \mathbb{P}_F(z \in \mathcal{Z}) = 1 \\ (\mathbb{E}_F[z] - \hat{\mu})^\top \hat{\Sigma}^{-1} (\mathbb{E}_F[z] - \hat{\mu}) \leq \gamma_1 \\ \mathbb{E}_F[(z - \hat{\mu})(z - \hat{\mu})^\top] \preceq \gamma_2 \hat{\Sigma} \end{array} \right\}$$

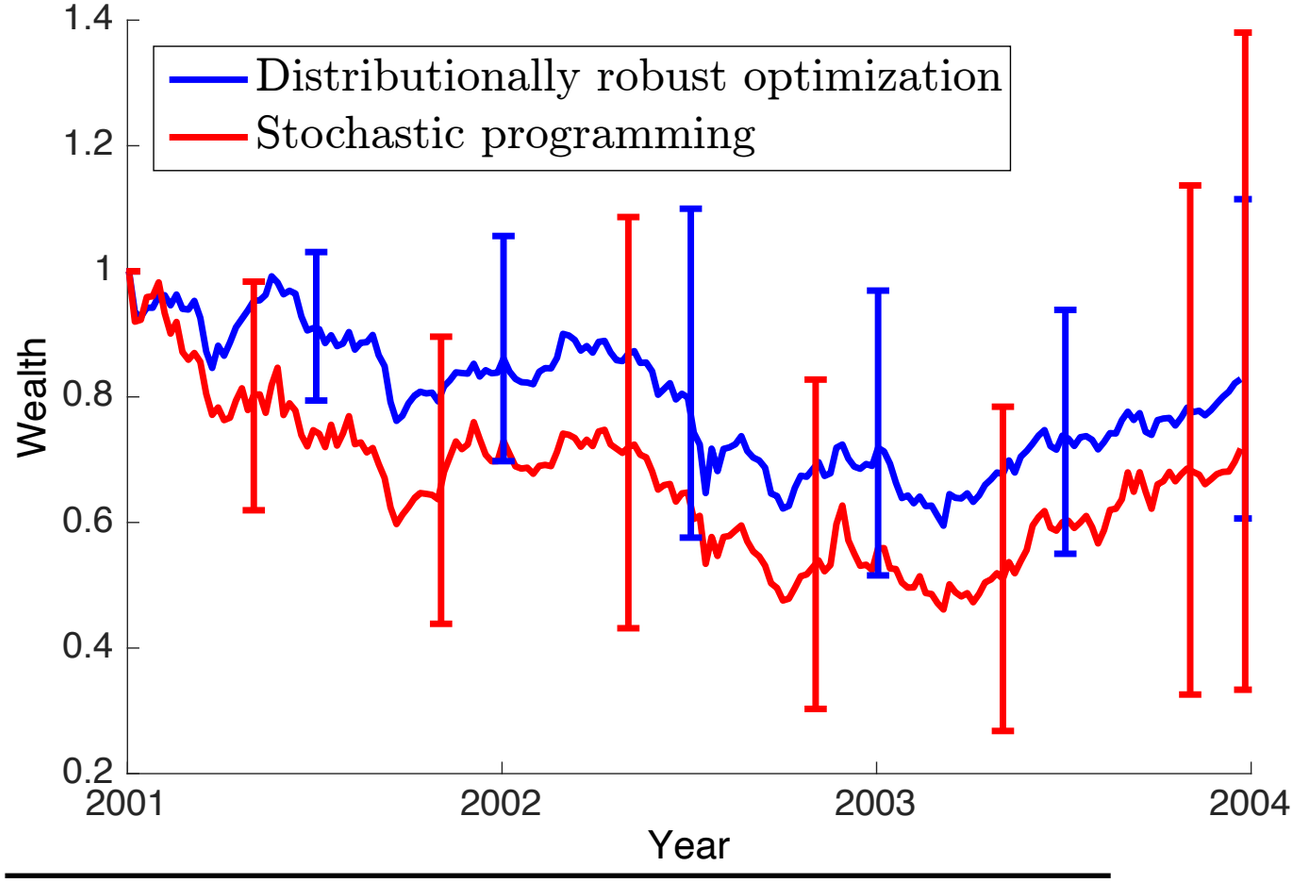
SP vs. DRO : a portfolio selection case study

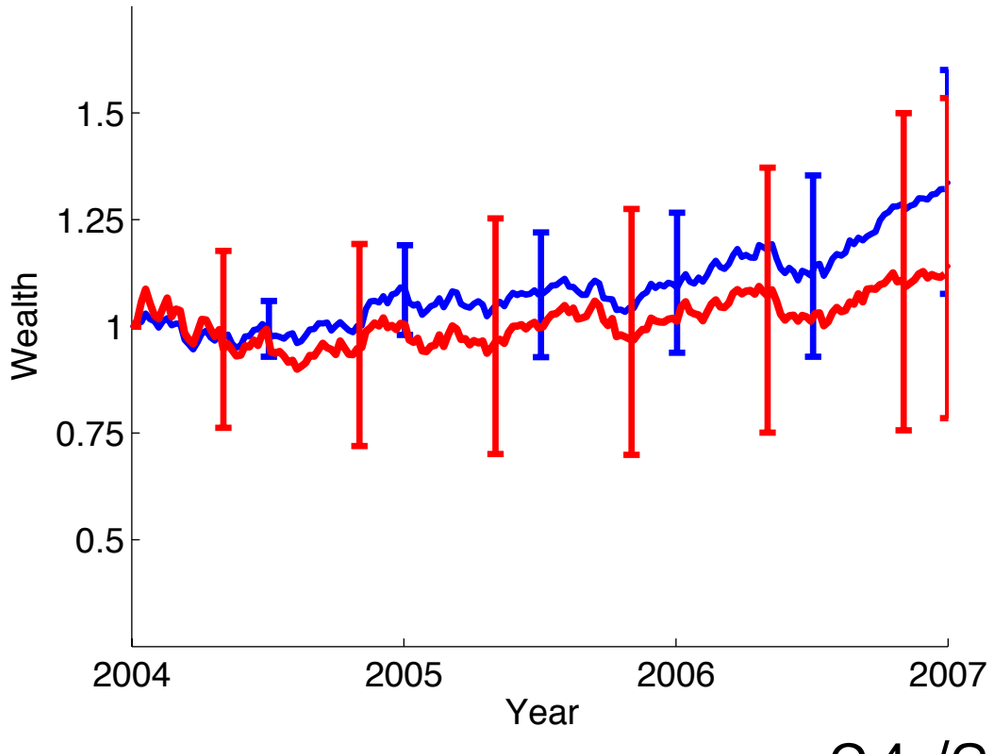


- Both methods maximize expected daily returns and use the last 30 days returns to build an uncertainty model for next daily return (distribution vs. distribution set based on moments)
- 300 experiments are performed

SP vs. DRO : a portfolio selection case study

- Below we present statistics of cumulated returns obtained using both methods
- Curves are the average cumulated wealth while bars indicate 5th and 95th percentiles





*) Delage & Ye, *Operations Research*, 2010.

Preference robust optimization

- Expected utility theory states that rational risk averse decision makers should optimize:

$$\underset{x \in \mathcal{X}}{\text{maximize}} \qquad \mathbb{E}[u(h(x, z))]\,,$$

where u( ) is a non-decreasing concave function

- In practice, how do we know what utility function to use?

Can we use risk assessment surveys?

Grable & Lytton, Financial Services Review (1999)

- You have just finished saving for a « once-in-a-lifetime » vacation. Three weeks before you plan to leave, you lose your job. You would:
  - Cancel the vacation
  - Take a much more modest vacation
  - Go as scheduled, reasoning that you need the time to prepare for a job search
  - Extend your vacation, because this might be your last chance to go first-class

- You are on a TV game show and can choose one of the following. Which would you take?
  - $1,000 in cash
  - A 50% chance at winning $ 5000
  - A 25% chance at winning $ 10,000
  - A 5% chance at winning $100,000

Preference robust optimization

- Pairwise comparisons of gambles can be used to construct an uncertainty set of plausible utility functions:

$$\mathcal{U} := \left\{ u : \mathbb{R} \to \mathbb{R} \;\middle|\; \begin{array}{l} u(\cdot) \text{ is non-decreasing} \\ u(\cdot) \text{ is concave (i.e. risk aversion)} \\ u'(\cdot) \text{ is convex (i.e. prudence)} \\ \mathbb{E}[u(W_k)] \geq \mathbb{E}[u(Y_k)] \, \forall \, k = 1, \ldots, K \end{array} \right\}$$

- Based on this set, different u( ) leads to different expected utility measurement, and different certainty equivalent amounts:

$$u(\cdot) \in \mathcal{U} \;\rightarrow\; \mathbb{E}[u(h(x,z))] \;\rightarrow\; u^{-1}(\mathbb{E}[u(h(x,z))])$$

Preference robust optimization

- One can try to maximize the worst-case certainty equivalent:

$$\underset{x\in\mathcal{X}}{\text{maximize}} \qquad \min_{u\in\mathcal{U}} u^{-1}(\mathbb{E}[u(h(x,z))])\,,$$

- When h(x,z) is concave in x, objective function is quasiconcave and reduces to:

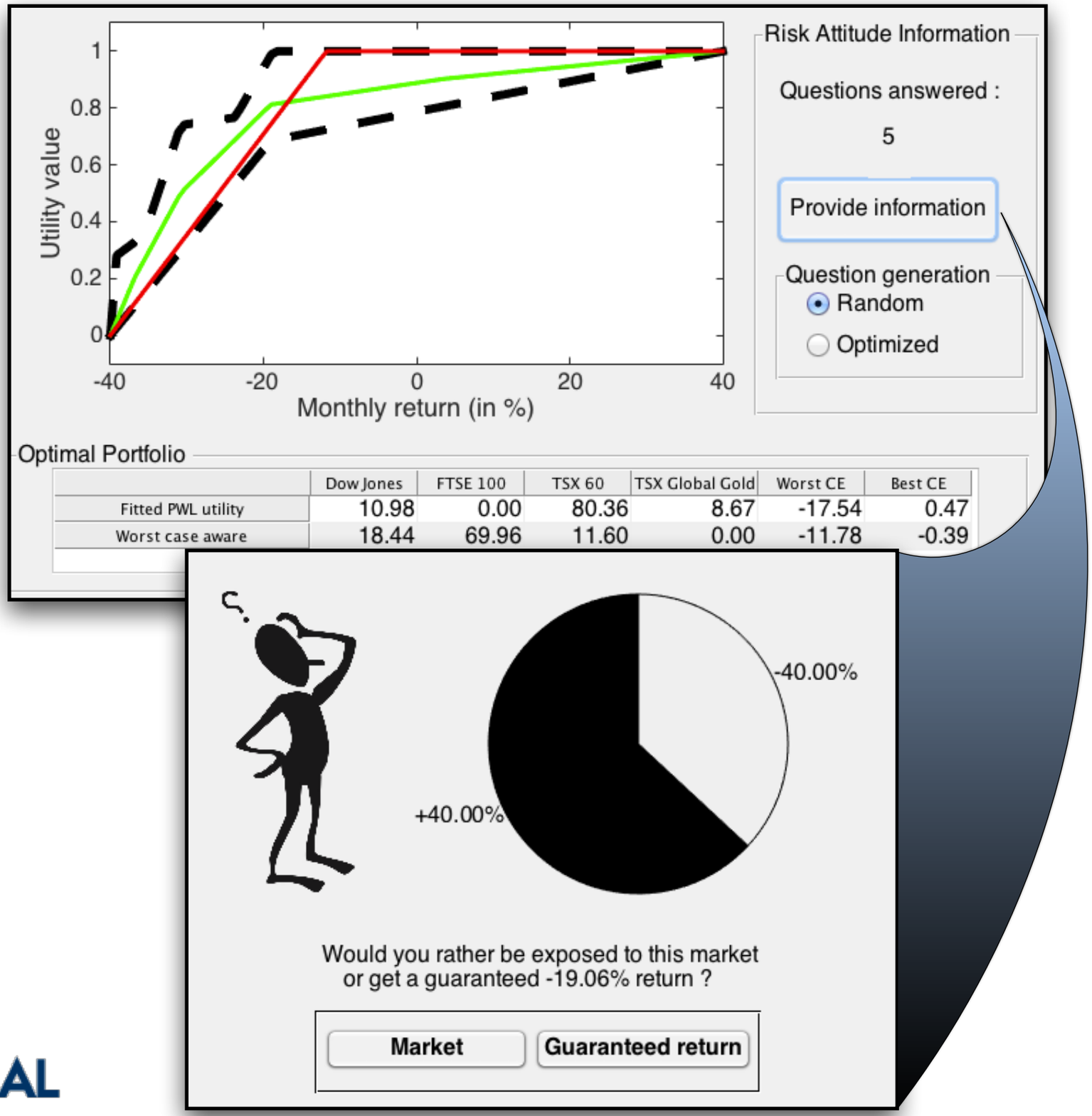
$$\underset{x\in\mathcal{X},t}{\text{maximize}} \;\; t \;\; \text{subject to} \;\; u^{-1}(\mathbb{E}[u(h(x,z))]) \geq t \;\; \forall\, u \in \mathcal{U}$$

- Equivalently one can solve:

$$\max_{t} \;\; t \;\; \text{subject to} \;\; \max_{x\in\mathcal{X}} \min_{u\in\mathcal{U}} \mathbb{E}[u(h(x,z))] - u(t) \geq 0$$

*) Armbruster & Delage, *Management Science*, 2015.
Delage & Li, *Management Science*, 2017.

A new tool for interacting with investors



MATH80624A : Quantitative risk management using robust optimization

- Offered in **Fall 2026**, held on Fridays 12:00 - 15:00
- Applications:
  - Logistics
  - Finance
  - Marketing
  - etc.
  - Electrical engineering
  - Aerospace
  - Data mining
- Topics:
  - Robust counterpart of Linear Programs
  - Data-driven uncertainty set design
  - Robust nonlinear programming
  - Adjustable robust linear programming
  - Distributionally robust optimization
  - Globalized robust counterparts
  - Pareto efficiency in robust optimization
- More details available in Lecture Notes (url)