Robust Optimization in Finance

Reha H. Tütüncü

Carnegie Mellon University
Department of Mathematical Sciences

Coimbra, July 5, 2005

Outline

1. Uncertainty in Optimization
2. Objective vs. Constraint Robustness
3. Absolute vs. Relative Robustness
4. Adjustable Robustness
5. Tools, Strategies, Methods

Optimization Problems and Uncertainty

Consider the following "generic" optimization problem:

$$(\mathcal{OP}) \qquad \begin{array}{lrcl} \min_x & f(x,p) & & \\ & G(x,p) & \in & K. \end{array} \tag{1}$$

where

- x is the variable vector
- p is a parameter vector
- f is the objective function that may depend on parameters p
- G represents m constraint functions that may depend on p
- K is a fixed subset of $\Re^m$.

When p is given this is a standard optimization problem.
Robust optimization is concerned with the case when p is uncertain or unknown.

Uncertainty in Optimization

Traditional approaches include

- Sensitivity analysis: Solve the problem with a fixed choice of the parameters p and then compute the sensitivity of the solution to the variations in this parameter.
  - Advantages: Relatively easy to do. Duality an important tool.
  - Disadvantages: A reactive rather than pro-active approach to uncertainty.
- Stochastic Programming: Develop a distributional model for uncertainty, generate numerous sample realizations using these distributions, replace uncertain quantities with expected values and solve the resulting problem.
  - Advantages: Pro-active, intuitive, can benefit from probabilistic information, easy to model recourse actions
  - Disadvantages: Resulting problems can be very large and hard to solve, low-probability extreme events are mostly ignored, difficult to incorporate "hard" constraints.

A Robust Optimization Formulation

Consider the following optimization problem:

$$(\mathcal{OP}_{uc}) \qquad \begin{array}{rcl} \min_x & f(x,p) & \\ & G(x) & \in & K. \end{array}$$

where we now assume that p is uncertain/unknown.

Although we do not know p with certainty, we may know a set $\mathcal{U}$ (an uncertainty set) that p must lie in.

A typical RO question: What choice of the variables of the problem will optimize the worst case objective value?

$$(\mathcal{OROP}) \qquad \begin{array}{rcl} \min_x & \max_{p\in\mathcal{U}} f(x,p) & \\ & G(x) & \in & K. \end{array}$$

This is the robust counterpart of $(\mathcal{OP}_{uc})$.

What is Robust Optimization?

- A modeling approach for optimization problems with uncertain inputs.
- A complementary alternative to stochastic programming and sensitivity analysis.
- Seeks a solution that will have a "good" performance under many/most/all possible realizations of the uncertain input parameters.
- Unlike stochastic programming, no distribution assumptions on uncertain parameters–each possible value equally important (this can be good or bad).
- Represents a conservative viewpoint when it is worst-case oriented.

Robust Optimization is especially useful when ...

- ... some of the problem parameters are estimates and carry estimation risk.
- ... there are constraints with uncertain parameters that must be satisfied regardless of the values of these parameters.
- ... the objective functions/optimal solutions are particularly sensitive to perturbations.
- ... decision-maker can not afford low-probability high-magnitude risks.

Challenges and Opportunities

- Uncertainty sets are used describe the uncertainty on the parameters. How should we choose the shape and size of an uncertainty set?
- Some guidelines: The shape should be related to the sources of uncertainty and levels of sensitivity. The size should be determined by the desired level of robustness.
- Robust counterparts are typically more complicated than the original problem because of the two (and even three) levels of optimization.
- As a result, the robust problem may be “nasty” (e.g., nonconvex opt) even when the original problem is “nice” (convex opt). Robust problems are harder to solve.
- With the variations on both the original problems and also the models of robustness, there is no unified approach/strategy to solve/reformulate/simplify robust optimization problems. Conic optimization methods, however, are emerging as very useful tools.

Uncertainty Sets

- Uncertainty is described through sets that contain many/most/all possible values of uncertain parameters.
- Some examples:
  - $\mathcal{U} = \{p_1, p_2, \ldots, p_k\}$ (scenarios)
  - $\mathcal{U} = \mathrm{conv}(p_1, p_2, \ldots, p_k)$ (polytopic sets)
  - $\mathcal{U} = \{p : l \leq p \leq u\}$ (intervals)
  - $\mathcal{U} = \{p : p = p_0 + Mu, \|u\| \leq 1\}$ (ellipsoids)
- Uncertainty sets can represent/be formed by difference of opinions, alternative estimates, historical data, Bayesian techniques, etc.

Robust Optimization Formulations

- Consider an optimization problem with input parameters p and decision variables x represented as follows:

$$\min f(x,p) \text{ s.t. } G(x,p) \in K.$$

A standard constrained optimization problem when p is known and given.

- Consider the case when p is uncertain but is known to be an element of the uncertainty set $\mathcal{U}$. A pessimistic perspective: What is the worst-case objective value for a particular choice $\hat{x}$ of the decision variables?

$$\bar{f}(\hat{x}) := \max_{p \in \mathcal{U}} f(\hat{x}, p).$$

- Now choose among all decision vectors x that are feasible for all p (that is, $x \in \{y : G(y,p) \in K, \forall p \in \mathcal{U}\}$) the vector that minimizes $\bar{f}(x)$:

$$\min_{x \in \{y: G(y,p) \in K, \forall p \in \mathcal{U}\}} \bar{f}(x) = \min_{x \in \{y: G(y,p) \in K, \forall p \in \mathcal{U}\}} \max_{p \in \mathcal{U}} f(x,p).$$

This is a robust optimization formulation for the problem above.

Outline

Objective vs. Constraint Robustness

- Objective robustness (Solution robustness):
  - Objective function of the optimization problem depends on uncertain parameters
  - We seek solutions that will remain close to optimal for all possible realizations of these uncertain parameters
  - Or, we seek solutions that optimize the worst-case behavior of the objective function with different parameters (min-max problems)
- Constraint robustness (Model robustness):
  - Constraints in the optimization problem depend on uncertain parameters
  - We seek solutions that will satisfy these constraints for all possible values of these parameters (semi-infinite problems)

Objective Robustness

- Hard to obtain solutions that will remain "close" to optimal for all possible uncertain inputs (more later on this)
- An alternative (and conservative) approach: Find a solution that has the best worst-case behavior.
- Here, worst-case behavior corresponds to the value of the objective function under the worst-possible realization of the uncertain inputs.

Objective Robustness: Formulation

This is what we saw before: Consider the following optimization problem:

$$(\mathcal{OP}_{uc}) \qquad \begin{array}{rcl} \min_x & f(x,p) & \\ & G(x) & \in \quad K. \end{array}$$

where p is uncertain and belongs to the uncertainty set $\mathcal{U}$.

$$(\mathcal{OROP}) \qquad \begin{array}{rcl} \min_x & \max_{p \in \mathcal{U}} f(x,p) & \\ & G(x) & \in \quad K. \end{array}$$

This is the robust counterpart of $(\mathcal{OP}_{uc})$.
A min-max optimization problem that can be solved efficiently for many classes of objective functions and uncertainty sets.

Objective Robustness: Asset Allocation Example

- Usual framework: n asset classes, expected returns given by μ and covariance matrix Σ. A portfolio of the available asset classes is denoted by the vector $x = (x_1, x_2, \ldots, x_n)$.
- Using a "risk-adjusted return" objective and representing portfolio constraints in the generic form $x \in \mathcal{X}$, we get a simple quadratic optimization problem:

$$\max_{x \in \mathcal{X}} \; \mu^T x - \lambda x^T \Sigma x$$

  This is one of the three alternative formulations of Markowitz' *mean-variance optimization (MVO)* problem. It is easily solved with quadratic programming software.
- Solutions are sensitive to μ and Σ, which must be estimated. Many different approaches address this sensitivity. We explore the usefulness of robust optimization as an alternative.

The Objective Robust Formulation

- We can use "box" type uncertainty sets that may be obtained from "confidence intervals" on the elements of the vector μ and the matrix Σ:

$$\mathcal{U} = \{(\mu, \Sigma) : \mu^L \leq \mu \leq \mu^U, \Sigma^L \leq \Sigma \leq \Sigma^U, \Sigma \text{ sym, pos. semidef.}\}.$$

- Then, the objective robust formulation is obtained as follows:

$$\max_{x \in \mathcal{X}} \{ \min_{(\mu,\Sigma) \in \mathcal{U}} \mu^T x - \lambda x^T \Sigma x \}.$$

- Using a saddle-point characterization of optimal solutions to this problem, one can obtain a solution using, for example, interior-point methods.

Constraint Robustness: An example

- Consider a process optimization problem for a multi-phase engineering process (e.g., chemical distillation)
- Balance constraints: materials entering a particular phase can not exceed materials produced/left over from the previous phase
- Uncertainty: external, uncontrollable factors that affect the yield of the processes in each phase
- Balance constraints must be satisfied, regardless of the values of these external factors $\rightarrow$ constraint robustness is desired

Constraint Robustness: Formulation

Consider an optimization problem of the form:

$$
(\mathcal{OP}_{uc}) \qquad \begin{array}{rll} \min_x & f(x) & \\ & G(x,p) & \in \; K. \end{array}
$$

where p is uncertain and must belong to the uncertainty set $\mathcal{U}$.
Then, a constraint-robust optimal solution can be found by solving the following problem:

$$
(\mathcal{CROP}^{)} \qquad \begin{array}{rll} \min_x & f(x) & \\ & G(x,p) & \in \; K, \quad \forall p \in \mathcal{U}. \end{array} \tag{2}
$$

A semi-infinite optimization problem, but can be solved efficiently for some classes of constraints and uncertainty sets.

Outline

Absolute vs. Relative Robustness

- We call a solution that optimizes the worst-case behavior of the objective function under uncertainty an absolute robust solution.
- Such conservatism may not be desirable in many modeling and decision-making environments.
- An alternative is to seek robustness in a relative sense.
- People whose performance is judged relative to their peers will want to make decisions that avoid falling severely behind their competitors under all scenarios rather than protecting themselves against the worst-case scenarios.

Relative Robustness: Formulation

Consider the following optimization problem:

$$(\mathcal{OP}_{uc}) \qquad \begin{array}{lrcl} \min_x & f(x,p) & & \\ & G(x) & \in & K. \end{array}$$

where p is uncertain with uncertainty set $\mathcal{U}$.
Given $p \in \mathcal{U}$ (fixed) let $z^*(p)$ denote the optimal value function, i.e.

$$z^*(p) = \min_x f(x,p) \text{s.t.} G(x) \in K$$

and let

$$x^*(p) = \arg\min_x f(x,p) \text{s.t.} G(x) \in K.$$

If we choose x as our vector and p is the realized value, the regret associated with not choosing $x^*(p)$ is

$$r(x,p) = f(x,p) - z^*(p) = f(x,p) - f(x^*(p),p).$$

Relative Robustness: Formulation

$$r(x,p) = f(x,p) - z^*(p) = f(x,p) - f(x^*(p),p).$$

Now, for a given x consider the maximum regret function:

$$R(x) := \max_{p \in \mathcal{U}} r(x,p) = \max_{p \in \mathcal{U}} f(x,p) - f(x^*(p),p).$$

A relative robust solution is a vector x that minimizes the maximum regret:

$$(\mathcal{RR}) \quad \min_{x:G(x)\in K} \max_{p \in \mathcal{U}} f(x,p) - z^*(p).$$

- Since $z^*(p)$ is the optimal value function, this is a 3-level optimization problem.
- If f is linear in p, then $z^*(p)$ is a concave function. Therefore, the inner maximization problem in $(\mathcal{RR})$ is convex maximization and is difficult for most $\mathcal{U}$.

Relative Robustness: A variant

Instead of regret, we measure the distance to the optimal solution (set) from our chosen vector x:

$$d(x, p) = \|x - x^*(p)\|.$$

For a given x we consider the maximum distance function:

$$D(x) := \max_{p \in \mathcal{U}} d(x, p) = \max_{p \in \mathcal{U}} \|x - x^*(p)\|.$$

Now we seek x that

$$(\mathcal{RR})_2 \min_{x: G(x) \in K} \max_{p \in \mathcal{U}} \|x - x^*(p)\|.$$

- An attractive model if we have time to revise (slightly?) our x once p is revealed and want to choose an x that will not need much perturbation.
- $x^*(p)$ is a function even more difficult to work with than $z^*(x)$. There are not many results using this notion of robustness.

Outline

Adjustable Robustness

- We consider a multi-period uncertain optimization problem where uncertainty is revealed progressively through periods.
- A subset of the decision variables can be chosen after these parameters are observed in a way to correct the sub-optimality of the decisions made with less information in earlier stages.
- Adjustable robust optimization (ARO) formulations model these decision environments, allowing recourse action.
- Introduced in a recent paper by Ben Tal et al. and Guslitzer's MSc. thesis.

Adjustable Robustness

Consider the two-stage linear optimization problem given below:

$$\min_{\mathbf{x}^1,\mathbf{x}^2} \{\mathbf{c}^\top \mathbf{x}^1 : A^1\mathbf{x}^1 + A^2\mathbf{x}^2 \leq b\}.$$

$\mathbf{x}^1, \mathbf{x}^2$ are the first and second stage decision variables, A^1, A^2, b are the uncertain parameters.
Let $\mathcal{U}$ denote the *uncertainty set* for parameters A^1, A^2, and b. The standard robust counterpart:

$$\min_{\mathbf{x}^1} \{\mathbf{c}^\top \mathbf{x}^1 : \exists \mathbf{x}^2 \ \forall (A^1, A^2, b) \in \mathcal{U} : A^1\mathbf{x}^1 + A^2\mathbf{x}^2 \leq b\}.$$

In contrast, the ARO allows $\mathbf{x}^2$ to depend on the realized values of the uncertain parameters. As a result, the adjustable robust counterpart problem is given as follows:

$$\min_{\mathbf{x}^1} \{\mathbf{c}^\top \mathbf{x}^1 : \forall (A^1, A^2, b) \in \mathcal{U}, \ \exists \mathbf{x}^2 = \mathbf{x}^2(A^1, A^2, b) : A^1\mathbf{x}^1 + A^2\mathbf{x}^2 \leq b\}.$$

Adjustable Robustness

- The feasible set of the second problem is larger than that of the first problem in general and therefore the model is more flexible.
- ARO models can be especially useful when robust counterparts are unnecessarily conservative.
- The price to pay for this additional modeling flexibility appears to be the increased difficulty of the resulting ARO formulations.
- The feasible set of the recourse actions (second-period decisions) depends on both the first-period decisions and the realization of the uncertain parameters leading to difficult problems.
- Positive results are limited to simple uncertainty sets or can be obtained using simplifying assumptions on the structure of recourse actions to uncertain parameters.

Outline

Tools: Conic Optimization Models and Software

Consider a generic linear optimization problem:

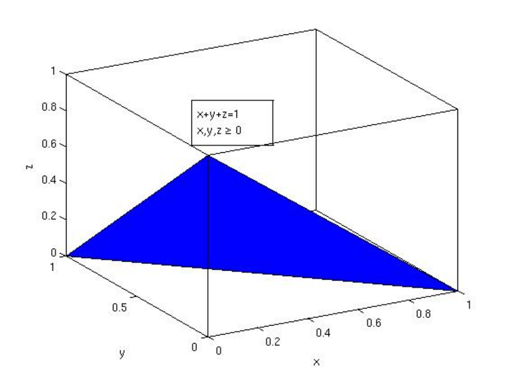
$$(\mathcal{LP}) \qquad \begin{array}{lrcl} \min_x & c^T x & & \\ & Ax & = & b \\ & x & \geq & 0. \end{array}$$

Here $A \in I\!R^{m \times n}$, $b \in I\!R^m$, $c \in I\!R^n$ are given, and $x \in I\!R^n$ is the variable vector to be determined as the solution of the problem. The feasible set is the intersection of the affine set $\{x : Ax = b\}$ with the non-negative orthant $I\!R^n_+ = \{x : x \geq 0\}$. The set $I\!R^n_+$ is a convex cone. In theory, the interior-point machinery will work when $I\!R^n_+$ is replaced by an arbitrary convex cone, say K. We then get the following conic optimization problem:

$$(\mathcal{LP}) \qquad \begin{array}{lrcl} \min_x & c^T x & & \\ & Ax & = & b \\ & x \in K & & \end{array}$$

Conic Optimization: Special Cases

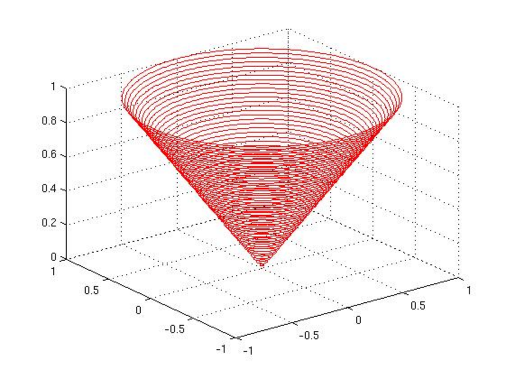
Linear Optimization ($K = \mathbb{R}^n_+$)



Conic Optimization: Special Cases

Quadratically constrained optimization using

$$K_q = \{x = (x_0, x_1) \in I\!R \times I\!R^n : x_0 \geq \|x_1\|\}$$

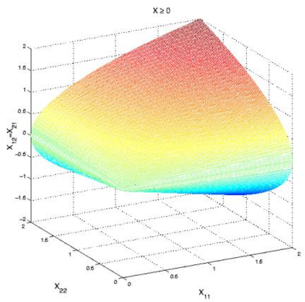


Conic Optimization: Special Cases

Semidefinite Optimization

$$K_s = \{X \in \Re^{n \times n} : X \succeq 0\}$$

Notation: $X \succeq 0$ means X is sym. and positive semidefinite.



Conic Optimization Software

- Products that solve special cases of conic optimization problems: too many to cite. There is now even a conic feature in Excel Solver.
- SeDuMi (Sturm): can solve LPs, SOCPs, SDPs
- SDPT3 (Todd, Toh, T.): can solve LPs, SOCPs, SDPs

An Example

Consider the following single-constraint linear program where the objective function is certain but the constraint coefficients are uncertain:

$$\min c^T x \text{ s.t. } a^T x + b \geq 0, \forall [a; b] \in \mathcal{U}$$

where the uncertainty set is ellipsoidal:

$$\mathcal{U} = \{[a; b] = [a^0; b^0] + \sum_{j=1}^{k} u_j [a^j; b^j], \|u\| \leq 1\}.$$

For a fixed x the robust constraint is satisfied if and only if

$$0 \leq \min_{[a;b] \in \mathcal{U}} a^T x + b \equiv \min_{u: \|u\| \leq 1} \alpha + u^T \beta,$$

where $\alpha = (a^0)^T x + b^0$ and $\beta = (\beta_1, \ldots, \beta_k)$ with $\beta_j = (a^j)^T x + b^j$.

An Example, continued

$$\min_{[a;b]\in\mathcal{U}} a^T x + b \equiv \min_{u:\|u\|\leq 1} \alpha + u^T \beta,$$

where $\alpha = (a^0)^T x + b^0$ and $\beta = (\beta_1, \ldots, \beta_k)$ with $\beta_j = (a^j)^T x + b^j$.
It is easy to see that the minimum is achieved at $u^* = -\frac{\beta}{\|\beta\|}$.
Thus, the robust constraint is equivalent to

$$(a^0)^T x + b^0 \geq \sqrt{\sum_{j=1}^{k} ((a^j)^T x + b^j)^2}.$$

This can be rewritten as a second-order cone constraint. Easily generalizes to multiple-constraint LPs.

S-procedure

Lemma: Let $F_i(x) = x^T A_i x + 2 + b_i^T x + c_i$, $i = 0, 1, \ldots, p$ be quadratic functions of $x \in \Re^n$. Then,

$$F_i(x) \geq 0, i = 1, \ldots, p \Rightarrow F_0(x) \geq 0$$

if there exist $\lambda_i \geq 0$ such that

$$\begin{bmatrix} A_0 & b_0 \\ b_0^T & c_0 \end{bmatrix} - \sum_{i=1}^{p} \lambda_i \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix} \succeq 0.$$

If $p = 1$, converse also holds as long as $\exists x_0$ s.t. $F_1(x_0) > 0$.
This lemma is useful for robust optimization problems where the uncertainty set is an ellipsoidal set.

Use of the S-procedure

Consider the following convex-quadratically constrained problem where the objective function is certain but the constraint coefficients are uncertain:

$$\min c^T x \text{ s.t. } -x^T(aa^T)x + 2b^T x + \gamma \geq 0, \forall [a; b; \gamma] \in \mathcal{U}$$

where the uncertainty set is ellipsoidal:

$$\mathcal{U} = \{[a; b; \gamma] = [a^0; b^0; \gamma^0] + \sum_{j=1}^{k} u_j [a^j; b^j; \gamma^j], \|u\| \leq 1\}.$$

Since membership in $\mathcal{U}$ is determined via a quadratic inequality ($u^T I u \leq 1$), and we want this inequality to imply another quadratic inequality, this is a tailor-made problem for the S-procedure.

The robust constraint can now be written as a positive semidefiniteness constraint.

While positive semidefiniteness constraint is more “complicated” than a quadratic constraint, effectively this procedure reduces a semi-infinite optimization problem to a finite optimization problem.

Use of Duality and Saddle-Point Conditions

Recall the robust mean-variance optimization problem:

$$\max_{x \in \mathcal{X}} \{ \min_{(\mu,\Sigma) \in \mathcal{U}} \mu^T x - \lambda x^T \Sigma x \}.$$

There is a *dual* of the robust MVO problem:

$$\min_{(\mu,\Sigma) \in \mathcal{U}} \{ \max_{x \in \mathcal{X}} \mu^T x - \lambda x^T \Sigma x \}.$$

Given that $f(x, \mu, \Sigma) = \mu^T x - \lambda x^T \Sigma x$ is concave in x and convex (linear, in fact) in (μ, Σ), if $\mathcal{X}$ and $\mathcal{U}$ are nonempty optimal values of these problems coincide and there exists a saddle point $(x^*, (\mu^*, \Sigma^*))$ such that

$$f(x, \mu^*, \Sigma^*) \leq f(x^*, \mu^*, \Sigma^*) \leq f(x^*, \mu, \Sigma), \ \forall x \in \mathcal{X}, (\mu, \Sigma) \in \mathcal{U}.$$

The Algorithm

Saddle-Point Algorithm (SP Algorithm)

1. *Initialization:*
   Choose $\alpha > 0$ *and* $\beta > 0$. *Find a* $t_0 > 0$ *and* $(x_0, \mu_0, \Sigma_0) \in \mathcal{X}_R^0 \times \mathcal{U}^0$ *that satisfies* $\eta(\phi_{t_0}, x_0, \mu_0, \Sigma_0) \leq \beta$. *Set* $k = 0$.
2. *Iteration:* **while** $t_k < M$, *set*
   $$t_{k+1} = (1+\alpha)t_k.$$
   Take a full Newton step:
   $$(x_{k+1}, \mu_{k+1}, \Sigma_{k+1}) = (x_k, \mu_k, \Sigma_k) - \left[\nabla^2 \phi_{t_{k+1}}(x_k, \mu_k, \Sigma_k)\right]^{-1} \nabla \phi_{t_{k+1}}(x_k, \mu_k, \Sigma_k).$$
   Set $k = k+1$.
   end

Above, η is the "Newton decrement" and measures the proximity of our iterates to the central path, ϕ_t is the saddle-barrier function. α and β need to satisfy certain conditions to ensure polynomial convergence.