CHAPTER 10
OPTIMIZATION UNDER UNCERTAINTY

The central goal of this chapter is to understand how we can formulate and solve optimization problems while accounting for uncertainty in the model parameters. We will discuss a few methods that can be used which include:

- Stochastic optimization
- Robust optimization

General problem description

Let's consider that we have an optimization problem, a general MINLP, and that we have also a set of parameters (θ) that we use in the formulation of this problem.

$$\min_{x,y} f(x, y, \theta)$$

$$s.t. \, g(x, y, \theta) \leq 0$$

In many instances, these parameters will have uncertainty associated with them. That is, it is possible that we do not know for sure what is their value, however, these parameters are neither decision variables, that is I do not know for sure what is the value that they are going to take), but at the same time I do not have any decision power over their value (thus, parameters in this type of problems are variables, just not decision variables). Thing for example in the price of a feedstock, you do not decide what is the value that is going to have, and you do now know for sure what is the exact value that you are going to get. Note that once the value of the parameters is fixed, then we recover an MINLP, NLP, or LP problem.

The question is then how we can deal with the uncertainty related to these parameters. This is the whole topic of optimization under uncertainty, and there are different approaches that we can use, that include:

- Stochastic optimization-> A distribution function for uncertain parameters exists and you try to use this information to reformulate your optimization problem
- Robust optimization-> A worst case approximation to the problem of uncertainty.

It is important to note that these approaches take uncertainty into consideration when solving the optimization problem. When uncertainty is included, more conservative but also more reliable solutions are obtained in comparison with the case where no uncertainty is considered.

A comparison between the main features of these two optimization strategies is shown in the table below:

	Stochastic	Robust
Assumption	Known probability for y	$\mathbf{y}$ bounded, $\mathbf{y} \in Y$
Objective	e.g., $\min_{\mathbf{x}} \mathbb{E}_Y(f(\mathbf{x},\mathbf{y}))$	$\min_{\mathbf{x}} \max_{\mathbf{y}\in Y} f(\mathbf{x},\mathbf{y})$
	Optimality in probabilistic sense	Optimality for the worst case
Constraint	e.g., $\mathbb{P}_Y(\mathbf{c}(\mathbf{x},\mathbf{y}) \le \mathbf{0}) \ge \alpha, \alpha \in (0,1)$	$\mathbf{c}(\mathbf{x},\mathbf{y}) \le \mathbf{0}, \forall \mathbf{y} \in Y$
	Chance for feasibility	Guaranteed feasibility

Stochastic optimization

It was initially developed in the 50's by Dantzig (the father of linear programming). In PSE it took a while before we catch up, the reason is that stochastic programming problems are computationally heavier than their deterministic counterparts (no uncertainty considered). In this approach, we assume that the probability distributions associated with the different scenarios are known a priori. The uncertainties are usually characterized by some probability distribution that can be approximated using a discrete approximation. For example, the realizations of the demand for a product:

S1: Low demand-> occurs with known probability p1

S2: Medium demand-> occurs with known probability p2

S3: High demand-> occurs with known probability p3

If there is more than 1 parameter, scenarios are constructed based on the cartesian product for the realizations of each parameter.

Two state stochastic programming

One of the more intuitive and common ways to deal with uncertainty is using what is known as two-stage stochastic programming, it is a subset of the stochastic programming approaches. To formulate the problem mathematically, let's consider the following problem description. We have:

$\theta \in \Theta$ uncertain parameters with kwon probability distribution

x decision variables that must be fixed before the uncertainty is realized with constraints, where x can be partitioned into real and binaries such that $x^R \in X^R \subseteq R^{n1}, x^B \in X^B \subseteq \{0,1\}^{m1}$. Furthermore, we have a number of constraints associated with these variables $c^U(x^R, x^B) \le 0$.

z decision variables that can be fixed after the uncertainty is realized, where x can be partitioned into real and binaries such that $z^R \in Z^R \subseteq R^{n2}, z^B \in Z^B \subseteq \{0,1\}^{m2}$ with constraints $c^L(x, z, \theta)$.

A separable objective function that is to be minimized such that $f(x, z, \theta) = f^U(x) + f^L(x, z, \theta)$

The two-stage stochastic program can be written as follows, where first stage decisions are made before the uncertainty is revealed, and second stage decisions (corrective actions to hedge against uncertainty) are made once the uncertainty is revealed:

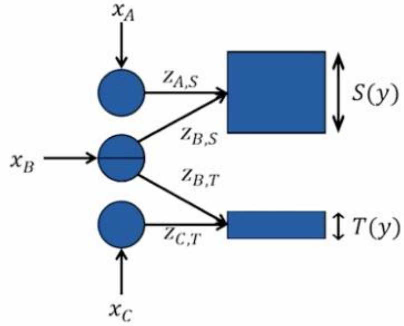
<table>
<tr><th>First stage</th><th>Second stage</th></tr>
<tr><td>$\min_x f^U(x) + E_Y(F(x,\theta))$
$s.t. c^U(x) \leq 0$</td><td>$F(x,z,\theta) = \min_z f^L(z,\theta)$
$s.t. c^L(x,z,\theta) \leq 0$</td></tr>
<tr><td colspan="2">$\min_x f^U(x) + E_Y(f^L(x,z,\theta))$
$s.t. \begin{matrix} c^U(x) \leq 0 \\ c^L(x,z,\theta) \leq 0 \end{matrix}$</td></tr>
</table>

In order for us to solve this problem, we can obtain an equivalent deterministic solution. The fundamental idea behind this approach is to think that it is possible to estimate the expected value, as the weighted sum of the stochastic portion of the objective function over a collection of scenarios ($w \in W$) occurring with probability p_w. The constraints of the problem associated with the second stage must be satisfied at each of the scenarios (therefore the second stage variables z are indexed by scenario). In general, this can be stated as follows:

$$\min f^U(x) + \sum_{w \in W} p_w F^l(x, z_w, \theta_w)$$
$$s.t. \begin{matrix} c^U(x) \leq 0 \\ c^L(x, z_w, \theta_w) \leq 0, \forall w \in \end{matrix}$$

This formulation requires us to know the probability of occurrence of a given scenario, but in principle it allows us to solve the optimization problem. Once fundamental limitation of the formulation is that it is often too large, so special algorithms have been developed to efficiently solve this type of problems while exploiting their numerical features.

Example: Let's consider the following case. In which, I want to decide on the amount that we are going to invest in three processes (x_A, x_B, x_C) that produce two products S and T. The demand for those products (denoted as S and T) is uncertain, and it will depend, on the state of the economy. If we denote z as the amount that we produce by each one of these processes, then we can formulate the problem of interest as follows:



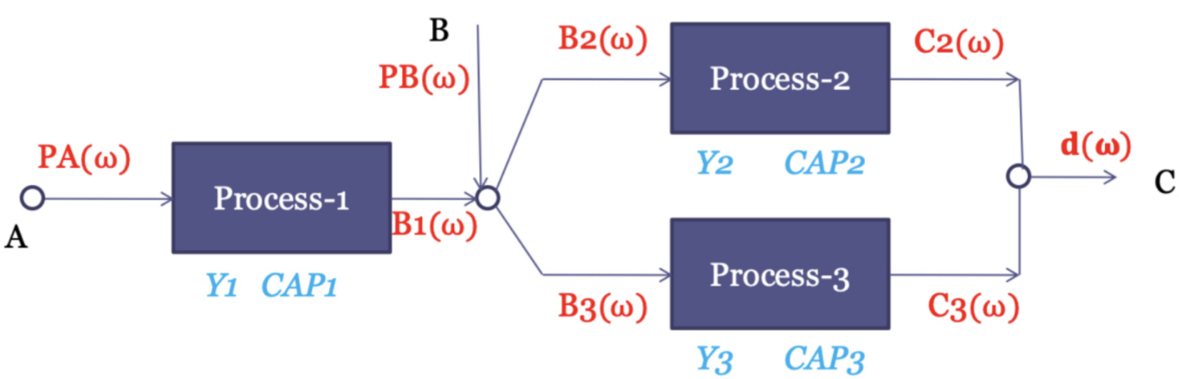
$\min d_A x_A + d_B x_B + d_C x_C + \mathbb{E}(F(x,z,\theta))$ $x_A + x_B + x_C \le x_{max}$ $x_A, x_B, x_C \ge 0$ $F(x,z,\theta) = \min -d_s(z_{A,s} + z_{B,s}) - d_T(z_{B,T} + z_{C,T})$ $s.t.$ $z_{A,s} \le x_A$ $z_{B,s} + z_{B,T} \le x_B$ $s.t.$ $z_{C,T} \le x_B$ $z_{A,S} + z_{B,S} \le S$ $z_{B,T} + z_{C,T} \le T$	$\min d_A x_A + d_B x_B + d_C x_C + \mathbb{E}(-d_s(z_{A,s} + z_{B,s}) - d_T(z_{B,T} + z_{C,T}))$ $x_A + x_B + x_C \le x_{max}$ $x_A, x_B, x_C \ge 0$ $z_{A,s} \le x_A$ $s.t.$ $z_{B,s} + z_{B,T} \le x_B$ $s.t.$ $z_{C,T} \le x_B$ $z_{A,S} + z_{B,S} \le S$ $z_{B,T} + z_{C,T} \le T$

Where d_A, d_B and d_C represent the capital cost investments, and d_s and d_T the prices of each of the products. If we are to find the equivalent deterministic problem, we would obtain the following:

$$\min d_A x_A + d_B x_B + d_C x_C - \sum_{w \in W} p_w \left(d_s(z_{A,s} + z_{B,s}) + d_T(z_{B,T} + z_{C,T}) \right)$$

$$
\begin{gathered}
x_A + x_B + x_C \le x_{max} \\
x_A, x_B, x_C \ge 0 \\
z_{A,s,w} \le x_A \\
s.t. \quad z_{B,s,w} + z_{B,T,w} \le x_B, \forall w \in W \\
s.t. \quad z_{C,T,w} \le x_B, \forall w \in W \\
z_{A,S,w} + z_{B,S,w} \le S_w, \forall w \in W \\
z_{B,T,w} + z_{c,T,w} \le T_w, \forall w \in W
\end{gathered}
$$

Example 2: Consider producing a chemical C which can be manufactured with either process 2 or process 3, both of which use chemical B as raw material. B can be purchased from another company and/or manufactured with process 1 which uses A as a raw material. The demand for chemical C, denoted as d, is the source of uncertainty. The superstructure of the process network is shown in Figure 3, which outlines all the possible alternatives to install this chemical plant. The alternatives include (1) All three processes are selected. (2) A true subset of the three processes are selected. (3) None of the three processes are selected.



The final products can be sold by 25 dollars per kilogram. The fixed cost of investment is 10 dollars for process 1, 15 dollars for process 2, and 15 dollars for process 3. The purchase cost of chemicals A and B is 4.5 dollars per kilogram, and 9.5 dollars per kilogram. With operating costs for all processes been 0.5 dollars per kilogram. U/sing this information the formulation is as follows:

The optimal formulation for this problem is as follows:

$$\max \begin{array}{l} -(10y_1 + 15y_2 + 20y_3 + CAP_1 + 1.5CAP_2 + 2CAP_3) + \\ \mathbb{E}(-4.5PA(w) - 9.5PB(w) - 0.5PA(w) - 0.5B_2(w) - 0.5B_3(w) + 25C_2(w) + 25C_3(w)) \end{array}$$

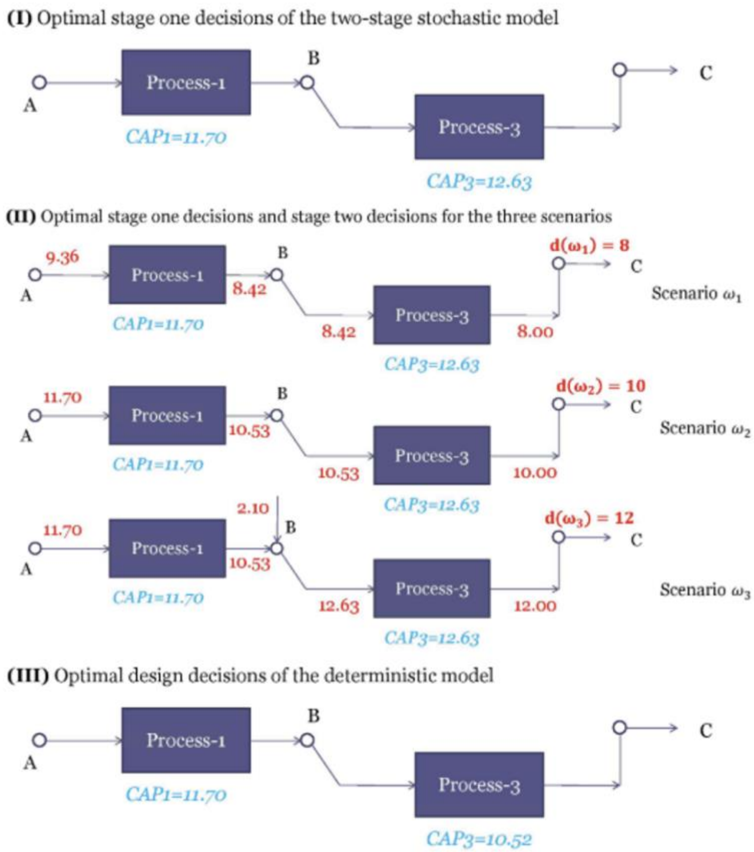
$$s.t. \quad \begin{array}{c} CAP_1 \leq Uy_1 \\ CAP_2 \leq Uy_2 \\ CAP_3 \leq Uy_3 \\ y_2 + y_3 \leq 1 \\ PA_w \leq CAP_1, \forall w \in W \\ B2_w \leq CAP_2, \forall w \in W \\ B3_w \leq CAP_3, \forall w \in W \\ B1(w) = 0.9PA(w), \forall w \in W \\ C2(w) = 0.82B2(w), \forall w \in W \\ C3(w) = 0.95B3(w), \forall w \in W \\ B1(w) + PB(w) = B2(w) + B3(w), \forall w \in W \\ C2(w) + C3(w) = d(w), \forall w \in W \end{array}$$

Suppose we have a 3-scenario problem where the demands d(ω) take values d(ω1) = 8, d(ω2) = 10, d(ω3) = 12, with probabilities τ(ω1) = 0.25, τ(ω2) = 0.5, τ(ω3) = 0.25, respectively. The optimal first-stage decisions are to select processes 1 and 3 with capacities 11.70 and 12.63, respectively. Note that the first-stage decisions are made "here-and-now" and thus are the same for all three scenarios. However, different second-stage decisions are taken for different scenarios as shown. When the demand is low d(ω1) = 8, processes 1 and 3 are not operating at their full capacity. For d(ω2) = 10, process 1 is operating at full capacity but process 3 is not. For d(ω3) = 12, both installed processes are operating at full capacity. The chemical A produced by process 1 is not able to satisfy the requirement of process 3. Therefore, additional chemical B needs to be purchased from other vendors when the demand is high. The expected profit of the stochastic program is 117.22. This optimal value of the stochastic program is called the value of the recourse problem (RP) in the literature (Birge and Louveaux, 2011) (RP=117.22).

Other than using stochastic programming, an alternative approach is to solve the deterministic model where the demand is fixed at its mean value, i.e., set d = 10. The optimal solution for this deterministic model is selecting processes 1 and 3 with capacities being 11.70, and 10.52, respectively. The only difference from the stochastic solution is that the capacity of process 3 becomes lower. The reason is that the deterministic model is "unaware" of the high demand scenario and therefore makes the capacity of process 3 to be just enough to satisfy d = 10. However, if we use the deterministic solution for d = 12, it will result in lost sales. We can fix the first stage solutions to the optimal solutions and evaluate how it performs in the three scenarios by solving each stage two problem separately. An expected profit of 114.20 is obtained. This value is called the expected result of using the expected solution (EEV). One quantitative metric to evaluate the additional value created by stochastic programming compared with solving the deterministic model at mean value is a concept called the value of the stochastic solution (VSS) (Birge and Louveaux, 2011). If the problem is a maximization problem, VSS is defined as:

$$VSS = RP - EEV$$

Therefore, the value of the stochastic solution is 117.22 − 114.20 = 3.02 for the process network problem.

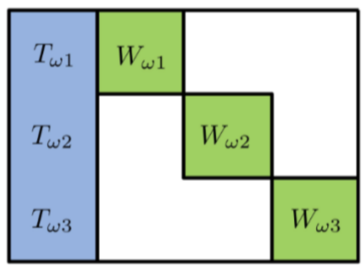


Algorithms to solve stochastic optimization problems

In principle, one could try to solve the deterministic equivalent of the stochastic programming and obtain the solution to the problem. This approach works well but becomes computationally intractable as the number of scenarios increases, note that each scenario considered adds several variables to your formulation. As a response to this challenge, the PSE community has been developing algorithms to efficiently solve 2-stage stochastic programs. Most of these algorithms exploit the structure of the problem, often decomposing the original problem into a collection of smaller subproblems. Different algorithm for different types of problems are summarized below:

- Mixed integer stochastic programs
    - Benders decomposition: This algorithm is one of the most classical examples of how to exploit the structure of a problem to solve if more efficiently. In this approach, the first stage decisions are typically called “complicating variables”, in the sense that if they are fixed, then it is possible to decompose the problem into $|W|$ subproblems one per each scenario. This is understood by writing the general MILP stochastic program as follows, note that if x is fixed then a collection of w subproblems appear:

$$\min c^T x + \sum_{w \in W} p_w d_w^T z_w$$
$$s.t. \begin{array}{c} c^U(x) = Ax \leq 0 \\ c^L(x, z_w, \theta_w) = W_w z_w + T_w x \leq h_w, \forall w \in W \end{array}$$



      We will not explore the details of the algorithm, but it relies on solving systematically a master problem and the collection of subproblems, until convergence is reached.
    - Lagrange decomposition
- Mixed integer nonlinear programs
    - Convex relaxation
        - Generalized Benders decomposition
    - Nonconvex relaxation
        - Generalized Benders decomposition for non-convex problems

Generating probability distributions for uncertain parameters

The sources of data for generating scenarios can come from historical data, time-series model, or expert knowledge.

Robust optimization

As we mentioned before, robust optimization is optimization for the worst case scenario, typically it can be represented as a case of semi-infinite optimization. This type of formulation can be stated as follows:

$$\min_{x \in X} f(x)$$

$$s.t. \, c(x, \theta) \leq 0, \forall \theta \in \Theta$$

If $\theta \in \Theta$ is a continuous interval, then, this type of problem is called semi-infinite, because you have a finite number of variables, but you have an infinite number of constraints. This is because you need to satisfy the constraint for every singe possible value of θ. In general, we will say that a point is feasible if it satisfy the constraint for all possible values of $\theta \in \Theta$. This can be reformulated as follows:

$$\min_{x \in X} f(x)$$

$$s.t. \max_{\theta \in \Theta} c(x, \theta) \leq 0, \forall \theta \in \Theta$$