

Adaptive robust optimization

From optimization

Author: Woo Soo Choe (ChE 345 Spring 2015)

Steward: Professor You, Dajun Yue

Contents

- 1 Introduction
- 2 Model Formulation
- 3 Methodology
- 4 Example
- 5 Application
- 6 Conclusion
- 7 Reference

Introduction

Traditionally, robust optimization has solved problems based on static decisions which are predetermined by the decision makers. Once the decisions were made, the problem was solved and whenever a new uncertainty was realized, the uncertainty was incorporated to the original problem and the entire problem was solved again to account for the uncertainty.[1] In order to address deal with the issues of uncertainties, several attempts were made in the field of optimization. While numerous technique exists, one of the widely studied way of dealing with uncertainty has been utilizing Stochastic Optimization method. In Stochastic Approach, the uncertainty is handled by assigning probability distribution to the uncertainty. Stochastic Optimization has proven its usefulness in certain areas, but this approach has couple drawbacks. First, while one can randomly assign probability distribution to make the model work, in a real life application, it is difficult to come up with an accurate probability distribution. Second, the Stochastic approach does not emphasize heavily on minimizing the cost of the worst case scenario, for the people who are making investment or company decisions need an Optimization technique that will yield conservative result and account for the uncertainties.

Adaptive Robust Optimization, currently led by Aharon Ben-Tal and dimitris Bertsimas, is an improved version of the traditional static robust optimization. Instead of assigning probability distribution to handle uncertainty, Adaptive Robust Optimization handles uncertainty by treating it as a function of ellipsoid, polyhedron, or any other ways that might best serve a specific case of interest. Furthermore, it utilizes the decisions made in the first stage to come up with a solution, which is used to arrive at the final answer even under uncertainties. Even though Adaptive Robust Optimization is a relatively new field, its capability as a way of solving a frequently asked questions in business and other real life application has proven the method useful. This wiki-page was created to introduce the topic of Adaptive Robust Optimization to fellow students with the hope of enriching ChemE 345 experience beyond the scope of what was covered in class.

Model Formulation

Adaptive Robust Optimization implements different techniques to improve on the original static robust optimization by incorporating multiple stages of decision into the algorithm. Currently, in order to minimize the complexity of algorithm, most of the studies on adaptive robust optimization have focused on two-stage problems. Generally, Adaptive Robust Optimization may be formulated in various different forms but for simplicity, Adaptive Robust Optimization in convex case was provided.[2]

$$\max_{x \in S} f(x) + \max_{b \in B} Q(x, b)$$

In the equation x is the first stage variable and y is the second stage variable, where S and Y are all the possible decisions, respectively. b represents a vector of data and when B represents uncertainty set.

In order for the provided convex case formulation to work, the case must satisfy five conditions:

1. S is a nonempty convex set
2. $f(x)$ is convex in x
3. Y is a nonempty convex set
4. $h(y)$ is convex in y
5. For all $i=1, \dots, n$, $H_i(x, y, b)$ is convex in (x, y) , $\forall b \in B$

Clearly, not every Adaptive Robust Optimization problem may be solved using exactly one model. However, key features that need to be present in a model of Adaptive Robust Optimization are the variables which respectively represent the multiple stages, uncertainty sets whether in ellipsoidal form, polyhedral form, or other novel way, and general layout of the problem which solves for the minimum loss at the worst case scenario. Furthermore, another key feature is that second stage variables are not known. Another form of Adaptive Robust Optimization formulation is provided below.[3]

$$\begin{aligned} \min_x \quad & c^T x + \max_{d \in \mathbb{D}} \min_{y \in \Omega} b^T y \\ \text{s.t.} \quad & Fx \leq f \\ & \Omega(x, d) = \{y : Hy \leq h, Ax + By \leq g, Jy = d\} \\ & \mathbb{D} = \{d : Dd \leq k\} \end{aligned}$$

Similarly as in the first formulation provided, x and y represent the first stage variable and the second stage variable respectively. In this case the, \mathbb{D} is the polyhedron uncertainty set of demand d and Ω represents the uncertainty set for the second stage variable y . In this case, H , A , B , g , J , D , and k are numerical parameters which could represent different parameters under different circumstances.

Methodology

In order to investigate how Adaptive Robust Optimization problem, numerous techniques may be used. However, given the scope of this page, only two of the techniques will be introduced. The two algorithms are Bender's Decomposition and Trilevel Optimization. When using Benders Decomposition approach, the algorithm essentially breaks down the original problem into the outer and inner problems. Once the problem is divided into two parts, the outer problem is solved using the Benders Decomposition and the inner problem is solved using the Outer Approximation. The detailed steps are as follows.[3]

Benders Decomposition

The Outer Problem: Benders Decomposition

Step 1: Initialize, by denoting the lower bound as $LB = -\infty$ and the upper bound as $UB = \infty$ and set the iteration count as $C = 0$. Then choose the termination tolerance ϵ .

Step 2: Solve the master problem

$$\begin{aligned} \max_{x, \zeta} \quad & c^T x + \zeta \\ \text{s.t.} \quad & Fx \leq f \\ & \zeta \geq -h^T \alpha_l + (Ax - g)^T \beta_l + d_l^T \lambda_l, \forall l \leq C \end{aligned}$$

In this case, (x_c, ζ_c) denote the optimum solution.

Step 3: Update the lower bound $LB = c^T x_c + \zeta_c$

Step 4: Increase C , the iteration count by 1

Step 5: Solve $I(x_c)$, the inner problem and denote the optimal solution as $(d_c, \alpha_c, \beta_c, \lambda_c)$. Update $UB = \min(UB, c^T x_c + I(x_c))$, where UB stands for the upper bound.

Detailed procedure of Step 5 is as follows.

```

if  $UB - LB \geq \epsilon$  then
    Go to step 2
else
    Calculate  $y_c$ , the dispatch variable given  $x_c$  and  $d_c$ 
end

```

The Inner Problem : Outer Approximation

Step 1: Initialize by using the commitment decision from the outer problem x_c from the outer problem. Then, find an initial uncertainty realization $d_1 \in \mathbb{D}$, set the lower bound $LOA = -\infty$ and the upper bound $UOA = \infty$, set iteration count $j=1$ and then termination tolerance which is denoted as θ

Step 2: Solve the sub-problem below.

$$\begin{aligned} S(x_c, d_j) \quad &= \max -h^T \alpha + (Ax_c - g)^T \beta + d_j^T \lambda \\ \text{s.t.} \quad & -H^T \alpha - B^T \beta + J^T \lambda = b \\ & \alpha \geq 0, \beta \geq 0 \end{aligned}$$

In the inner problem, the optimal solution is denoted as $(\alpha_j, \beta_j, \lambda_j)$. Furthermore, define $d_j^T \lambda_j$ as

$L_j(d_j, \lambda_j) + (d - d_j)^T \lambda_j + (\lambda - \lambda_j)^T d_j$. Then, update $LOA = \max(S(x_c, d_j), LOA)$

Step 3: Solve the master problem

$$\begin{aligned} U(d_j, \lambda_j) \quad &= \max -h^T \alpha + (Ax_c - g)^T \beta + \zeta \\ \text{s.t.} \quad & \zeta \leq L_i(d_i, \lambda_i), \forall i \leq j \\ & -H^T \alpha - B^T \beta + J^T \lambda = b \\ & Dd \leq k \\ & \alpha \geq 0, \beta \geq 0 \end{aligned}$$

Increase the iteration of j by 1. While the optimal solution is denoted as $(d_j, \alpha_j, \beta_j, \lambda_j, \zeta)$, update the upper bound as $UOA = \min(UOA, U(d_j, \lambda_j))$

```

if  $UB - LB \geq 0$  then

```

```

Go to Step 2
else
Return optimal solution as the output
end

```

As seen from the algorithms, Benders Decomposition divides an Adaptive Robust Optimization problem into outer and inner problems and solves them using two algorithms. While this may cause some confusion for ones who have no previous exposures to Benders Decomposition, the approach to solving the outer problem is also called the Benders Decomposition and the approach to solving the inner problem is called the Outer Approximation method. Fundamentally, this algorithm works by first solving the outer problem until $UB - LB \geq \epsilon$ condition is met and then use the x, ζ from the outer problem to plug into the inner problem and solve for the optimum solution until $UB - LB \geq 0$ condition is met. This method has an advantage over traditional Robust Optimization in a sense that it does not sacrifice as much optimality in the solution at the cost of obtaining a conservative answer. Unfortunately, Benders Decomposition method has three problems. First problem is the fact that the master problem relies on the dual variables of the inner and outer problems, which means that the sub-problems cannot have integer variables. Second problem is that the solution does not guarantee a global optimal solution, and it means that the algorithm may not return the absolute worst case scenario before returning the solution. Third problem is that it takes a long time to compute the answer and this might pose a problem when solving a large scale problem. In order to resolve this issue, another algorithm called Trilevel Optimization was proposed by Bokan Chen of Iowa University. Before iterative Trilevel Optimization algorithm applied, the problem needs to be reformulated in an appropriate form as shown below.

$$\begin{aligned}
 & \min_x c^T x + b^T y \\
 & s.t. \quad Fx \leq f \\
 & \quad \max_d \quad b^T y \\
 & \quad s.t. \quad Dd \leq k \\
 & \min_y b^T y \\
 & s.t. \quad Ax + By \leq g \\
 & \quad Hy \leq h \\
 & \quad Jy = d
 \end{aligned}$$

Equation $Fx \leq f$ represents the first constraints on the first stage commitment variables and equation $Dd \leq k$ represents the uncertainty set of demand. Equation $Ax + By \leq g$ represents the constraints that couples the commitment and dispatch variables. $Hy \leq h$ constrains the dispatch variable and $Jy = d$ is the constraint that couples the demand variables and the dispatch variables. Then, the reformulated model can be further refined into the following model.

$$\begin{aligned}
 & \min_{x, \phi} c^T x + \phi \\
 & s.t. \quad Fx \leq f \\
 & \quad \phi \geq b^T y, \forall y \in \mathbb{Y}_D
 \end{aligned}$$

When $\mathbb{Y}_D = \{y | Ax + By \leq g, Hy \leq h\} \cap \{y | Jy = d, d \in \mathbb{D}\}$. Assuming $\Omega \subset \mathbb{D}$, we have $\mathbb{Y}_\Omega = \{y | Ax + By \leq g, Hy \leq h\} \cap \{y | Jy = d, d \in \Omega\}$. This implies $\mathbb{Y}_\Omega \subset \mathbb{Y}_D$. This relaxes the problem into the following form.

$$\begin{aligned} \min_{x,\phi} \quad & c^T x + \phi \\ \text{s.t.} \quad & Fx \leq f \\ & \phi \geq b^T y, \forall y \in \mathbb{Y}_{\neq} \end{aligned}$$

This allows the trilevel problem to be split into the master problem and a sub-problem. Following is the relaxation of the master problem M of the trilevel problem as follows.

$$\begin{aligned} \min_{x,\phi} \quad & c^T x + \phi \\ \text{s.t.} \quad & Fx \leq f \\ & \phi \geq b^T y, \forall y \in \mathbb{Y}_{\mathbb{D}} \end{aligned}$$

The master problem M is a relaxation of the trilevel problem as follows:

$$\begin{aligned} \min_{x,\phi} \quad & c^T x + \phi \\ \text{s.t.} \quad & Fx \leq f \\ & \phi \geq b^T y^i, \forall i = 1, \dots, |\Omega| \\ & Hy^i \leq h, \forall i = 1, \dots, |\Omega| \\ & Ax + By^i \leq g, \forall i = 1, \dots, |\Omega| \\ & Jy^i = d^i, \forall i = 1, \dots, |\Omega| \end{aligned}$$

Following is the bilevel sub-problem which yields the dispatch cost under the worst-case scenario

$$\begin{aligned} \max_d \quad & b^T y \\ \text{s.t.} \quad & Dd \leq k \\ & \min_y b^T y \\ & \text{s.t. } Hy \leq h \\ & By \leq g - Ax \\ & Jy = d \end{aligned}$$

An Iterative Algorithm For The Trilevel Optimization Problem Optimization Problem

Step 1: Initialize by denoting lower bound as $LB = -\infty$ and the upper bound as $UB = \infty$. Then create and empty set Ω .

Step 2: Solve the master problem M. Where the solution of the problem is (x^M, ζ^M, y^M) . Then update the lower bound of the algorithm $LB = \max(LB, c^T x^M + \zeta^M)$.

Step 3: Solve the sub-problem $S(x^M)$. The solution to the problem is (y^S, d^S) . Update the upper bound which is $UB = \min(UB, c^T x^M + b^T y^S)$ and the set $\Omega = d^S \cap \Omega$.

if $UB - LB \gg 0$ then

Go to Step 2

else

Find $\text{argmax}_i b^T y^i$. Calculate the total cost $\zeta = c^T x^M + b^T y^i$, return the optimal solution as x^M, d^i, y^i, ζ

end

Example

In order to illustrate how Adaptive Robust Optimization works, a numerical example is given in this section. This example involves 3 factories and 5 customers and a detailed information is provided through the table below.

Factory	Customers					Fixed costs
	1	2	3	4	5	
1	2	3	4	5	7	2
2	4	3	1	2	6	3
3	5	4	2	1	3	3

[4]

Before solving the problem, the basic set up is as follows.

$$v_j = \min_{i \in O(y)} \{c_{ij}\}$$

$$w_{ij} \begin{cases} 0 & i \in O(y) \\ \max_{i \in C(y)} \{(v_j - c_{ij}), 0\} & i \in C(y) \end{cases}$$

In this case, v_{ij} are the dual variables associated with the demand constraints and w_{ij} represent the dual variables associated with the setup constraints. Furthermore the dual variable can be represented as u and it means the combination of (v, w) . From the proposition, the following Benders cut is derived.

$$\beta_y(y) = u(b - By) + f^t y$$

For this specific problem, the Benders cut can be rewritten as follows.

$$\beta_y(y) = \sum_{i=1}^m v_j + \sum_{i=1}^n (f_i - \sum_{j=1}^m w_{ij}) y_i$$

Returning to the problem, we denote factory 1, factory 2, and factory 3 as y_1, y_2, y_3 and to start the problem, we only assume factory 1 is open and in this case, $v_j = \min_{i \in O(y)} \{c_{ij}\}, j = 1, \dots, m$ would become

$$v_j = (2, 3, 4, 5, 7). \text{ Based on the proposition, } w_{ij} \text{ may be found as follows.}$$

$$w_{1j} = 0$$

$$w_{2j} = (0, 0, 3, 3, 1)$$

$$w_{3j} = (0, 0, 2, 4, 4)$$

Then, solving Benders Cut, we get the following result.

$$\beta_y(y) = \sum_{i=1}^m v_j + \sum_{i=1}^n (f_i - \sum_{j=1}^m w_{ij}) y_i$$

$$\beta_y(y) = 2 + 3 + 4 + 5 + 7 + (2 - 0)y_1 + (3 - (3 + 3 + 1))y_2 + (3 - (2 + 4 + 4 + 4))y_3$$

$$\beta_y(y) = 21 + 2y_1 - 4y_2 - 7y_3$$

From this, the upper bound on the solution, 23, obtained. Then, the Benders cut is used to solve the master problem and by inserting the Benders cut into the master problem, we get the problem in the following form.

$\min z$

$$s.t. \quad z \geq 21 + 2y_1 - 4y_2 - 7y_3$$

$$y \in \mathbb{B}^3$$

In the above problem, the optimal solution is $y=(0,1,1)$, meaning it is best to keep factory 1 closed and open factories 2 and 3. This yield a solution of 10, which becomes new y and one more iteration of the algorithm may be done with this. When the v_j and w_{ij} are found again with the solution, following values are obtained

$$v_j = (4, 3, 1, 1, 3)$$

$$w_{1j} = (2, 0, 0, 0, 0)$$

$$w_{2j} = w_{3j} = 0$$

Then the Benders cut was calculated again as follows.

$$\beta_y(y) = \sum_{i=1}^m v_j + \sum_{i=1}^n (f_i - \sum_{j=1}^m w_{ij})y_i$$

$$\beta_y(y) = (4 + 3 + 1 + 1 + 3) + (2 - 2)y_1 + (3 - 0)y_2 + (3 - 0)y_3$$

$$\beta_y(y) = 12 + 3y_2 + 3y_3$$

From this, we get a new upper bound which is 18 and the master problem looks like:

min z

$$s.t. \quad z \geq 21 + 2y_1 - 4y_2 - 7y_3$$

$$z \geq 12 + 3y_2 + 3y_3$$

$$y \in \mathbb{B}^3$$

As the solution to the problem, we get $y = (0, 0, 1)$. Then, we repeat the iteration process, which yields:

$$v_j = (5, 4, 2, 1, 3)$$

$$w_{1j} = (3, 1, 0, 0, 0)$$

$$w_{2j} = (1, 1, 1, 0, 0)$$

$$w_{3j} = 0$$

Then the Benders cut becomes:

$$\beta_y(y) = \sum_{i=1}^m v_j + \sum_{i=1}^n (f_i - \sum_{j=1}^m w_{ij})y_i$$

$$\beta_y(y) = (5 + 4 + 2 + 1 + 3) + (2 - 4)y_1 + (3 - 3)y_2 + (3 - 0)y_3$$

$$\beta_y(y) = 15 - 2y_1 + 3y_3$$

Now, there is no better upper bound and the master problem becomes:

min z

$$s.t. \quad z \geq 21 + 2y_1 - 4y_2 - 7y_3$$

$$z \geq 12 + 3y_2 + 3y_3$$

$$z \geq 15 - 2y_1 + 3y_3 \quad y \in \mathbb{B}^3$$

This yields the optimal solution of $y = (1, 0, 1)$ and the new lower bound is 16. when the iteration process is repeated until the upper and lower bound are the same, we obtain the optimal solution value of 16 and come to the decision of opening factories 1 and 3 only.

As seen from the iterative procedure, Trilevel Optimization also breaks an optimization problem into smaller parts and use iterative algorithm to close in the difference between the upper and the lower bound. However, the Trilevel Optimization addresses the issues with the Benders Decomposition approach.

Application

Adaptive Robust Optimization has applications in different fields because it can deal with circumstances where uncertainties are present. In real life, since uncertainties exists everywhere regardless of the field, Adaptive Robust Optimization may be used under any circumstances as long as the models may be accurately formulated. In the field of business, problems such as ones presented in the example section may be solved using Adaptive Robust Optimization, where decision makers need to know the worst possible outcome in order to make the best decision for the company.

Separately, even in the field of electrical engineering Adaptive Robust Optimization may be used. As discussed in

Alvaro Lorca's paper[5], Adaptive Robust Optimization may be used to handle daily operational problem of power systems, in which the nodal net electricity loads are uncertain. On a similar note, Adaptive Robust Optimization may be used in virtually any other field that involves uncertainty.

Conclusion

In the wiki-page, Adaptive Robust Optimization was introduced in a moderate detail. As stated in the introduction section, this approach handles the problems with uncertainties in a more efficient manner than pre-existing static Robust Optimization or Stochastic Optimization. However, because the field of Adaptive Robust Optimization is relatively new, numerous different algorithms are suggested to expand the use of the method. While it was not specifically mentioned in the paper, Adaptive Robust Optimization and other techniques such as Stochastic Approach are not mutually exclusive and may be used together to yield the most realistic solution as possible.

Reference

1. Virginie G. Recent Advance in Robust Optimization: An Overview, Universite Paris Dauphine, 2013
2. Zeng, B. An Exact Algorithm for Two-stage Robust Optimization with Mixed Integer Recourse Problems. Department of Industrial and Management Systems Engineering University of South Florida. 2012
3. Chen B. A new trilevel optimization algorithm for the twostage robust unit commitment problem. Iowa State University. 2013
4. Christensen U. Lecture Note on Benders Decomposition. (accessed June 2016)
5. Lorca, A. (2014). Multistage Adaptive Robust Optimization for the Unit Commitment Problem. Eprints for the Optimization Community

Retrieved from "https://optimization.mccormick.northwestern.edu/index.php?title=Adaptive_robust_optimization&oldid=4686"

-
- This page was last modified on 8 June 2015, at 00:03.
 - This page has been accessed 231,386 times.