Lecture notes for stochastic dual dynamic programming

Oscar Dowson

February 10, 2020

1 Recap: Benders decomposition

Consider

$$\begin{array}{ll} \min\limits_{x,y} & c^\top x + d^\top y \\ \text{s.t.} & Ax = b \\ & Tx + Wy = g \\ & x, y \geq 0, \end{array} \tag{1}$$

where A, T, and W are appropriately sized matrices, and b, c, d, g, x, and y are appropriately sized vectors.

Benders decomposition is an iterative method of solving problem (1) that works by decomposing the problem into a first-stage master problem:

$$\begin{array}{rll} V_1 = \min\limits_{x} & c^\top x + V_2(x) \\ \text{s.t.} & Ax = b \\ & x \geq 0, \end{array} \tag{2}$$

and a second-stage subproblem $V_2(x)$ which depends on the first-stage decision x:

$$\begin{array}{rll} V_2(x) = \min\limits_{\bar{x},y} & d^\top y \\ \text{s.t.} & \bar{x} = x \quad [\nu] \\ & T\bar{x} + Wy = g \\ & y \geq 0. \end{array} \tag{3}$$

We refer to the second-stage function $V_2(x)$ as the *cost-to-go* of the first-stage master problem. Henceforth, we will always assume that the second-stage problem has a feasible and bounded solution for all feasible x in the first-stage problem (i.e., the problem has *relatively complete recourse*).

Note that in problem (3), we have added a constraint $\bar{x} = x$ with dual variable ν. This dual variable is a subgradient of the function $V_2(x)$ with respect to x.

I like to visualize this two-stage problem with the following diagram. This will become useful when we consider more complicated models.

$$\begin{tikzcd} \bigcirc \arrow[r] & \boxed{t=1} \arrow[r] & \boxed{t=2} \end{tikzcd}$$

Figure 1: Two-stage deterministic problem

Then, we approximate the first-stage problem:

$$\begin{array}{rll} V_1^K = \min\limits_{x,\theta} & c^\top x + \theta \\ \text{s.t.} & Ax = b \\ & x \geq 0 \\ & \theta \geq \alpha_k + \beta_k^\top x, \quad k = 1, \ldots, K \\ & \theta \geq -M. \end{array} \tag{4}$$

Note that to ensure a bounded feasible solution, we introduce a lower bound on θ of $-M$, chosen such that it is smaller than the true cost-to-go. The coefficients of the cut generated in iteration k are as follows:

$$\alpha_k = V_2(x^*) - {\nu^*}^\top x^*, \quad \beta_k = \nu^*,$$

where x^* is the solution of the approximated master problem, and ν^* is the value of the second-stage dual variable for the $\bar{x} = x^*$ constraint.

After any iteration K, we can derive lower and upper bounds for the objective of the full problem (1). The objective value V_1^K of problem (4) is a valid lower bound. To obtain an upper bound, we solve problem (4) to obtain a solution x^*, and then solve problem (3) at x^* to obtain a solution y^*. The valid upper bound is $c^\top x^* + d^\top y^*$. The Benders decomposition algorithm has converged when the lower bound is equal to the upper bound.

2 Recap: Two-stage stochastic program

We assume we have a random variable drawn from finite sample space Ω with probability p_ω for each $\omega \in \Omega$. We call each ω a *scenario*.

We formulate directly the subproblems:

$$\begin{aligned} V_1 = \min_x \quad & c^\top x + \mathbb{E}[V_2(x,\omega)] \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{5}$$

and a second-stage subproblem $V_2(x,\omega)$ which depends on the first-stage decision x:

$$\begin{aligned} V_2(x,\omega) = \min_{\bar{x},y} \quad & d(\omega)^\top y \\ \text{s.t.} \quad & \bar{x} = x \quad [\nu] \\ & T(\omega)\bar{x} + W(\omega)y = g(\omega) \\ & y \geq 0, \end{aligned}$$

where d, T, W, and g may depend on ω.

We call $\mathbb{E}[V_2(x,\omega)]$ the *cost-to-go*.

Since the second stage now depends on the random variable ω, we can extend our earlier diagram to Figure 2. The squiggly line indicates that some information (the random variable ω) is arriving at the start of stage two, and can be observed by the agent prior to them making a decision.

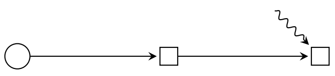

Figure 2: Two-stage stochastic problem

Expanding the expectation term in (5), we get:

$$\begin{aligned} V_1 = \min_x \quad & c^\top x + \sum_{\omega\in\Omega} p_\omega V_2(x,\omega) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

So we could formulate:

$$\begin{aligned} V_1^K = \min_x \quad & c^\top x + \sum_{\omega\in\Omega} p_\omega \theta_\omega \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \\ & \theta_\omega \geq \alpha_{k,\omega} + \beta_{k,\omega}^\top x, \quad \forall \omega \in \Omega, \end{aligned} \tag{multi-cut}$$

and then compute a cut for each realization of ω in every iteration. This is known as *multi-cut*.

Alternatively, we can formulate

$$\begin{aligned} V_1^K = \min_x \quad & c^\top x + \theta \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \\ & \theta \geq \mathbb{E}[\alpha_{k,\omega}] + \mathbb{E}[\beta_{k,\omega}]^\top x. \end{aligned} \tag{single-cut}$$

In practice, we use the single-cut formulation because it has many less constraints and tends to solve quicker.

Homework: Show that the objective of (multi-cut) provides a tighter lower bound than the objective of (single-cut).

2.1 Algorithm

Pseudo-code is given in Algorithm 1.

Algorithm 1: Benders decomposition for two-stage stochastic programs.

```
Set $K = 1$
while not converged do
    // Forward pass
    Solve $V_1^{K-1}$ and obtain solution $x^*$
    // Backward pass
    for $\omega \in \Omega$ do
        Solve $V_2(x^*, \omega)$ and obtain extreme point primal-dual solution pair
        Set $V_\omega^*$ to the optimal objective value
        Set $\nu_\omega^*$ to the optimal dual vector $\nu$
    end
    set $\beta_K = \sum_{\omega \in \Omega} p_\omega \nu_\omega^*$
    set $\alpha_K = \sum_{\omega \in \Omega} p_\omega V_\omega^* - \beta^\top x^*$
    Add the cut $\theta \geq \alpha_K + \beta_K^\top x$ to $V_1^{K-1}$, creating $V_1^K$
    $K \leftarrow K + 1$
end
```

3 Multistage stochastic program

Now consider a problem like the one in Figure 3. All we have done is appended some extra stages to the end of our two-stage graph. In this case, we have a multistage stochastic program with four stages.

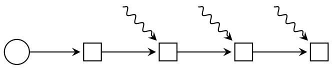

Figure 3: Multi-stage stochastic problem when $T = 4$.

What assumptions are we making?

Assumption 1. *The sample space Ω_t has a finite number of elements at each stage $t = 1, \ldots, T$.*

Assumption 2. *The random variables ω_t are stagewise-independent. That is, ω_i is independent of ω_j for all $i \neq j$.*

These two assumptions enforce independence in the random information is arriving, which we indicate by the squiggly lines entering the nodes in the graph.

Now let's think about writing down the recursive formulation of our problem. As in the two-stage case, the objective of the agent is to minimize the first-stage cost plus the expected cost to go:

$$\begin{aligned} V_1 = \min_{x_1} \quad & c_1^\top x_1 + \mathbb{E}[V_2(x_1, \omega_2)] \\ \text{s.t.} \quad & A_1 x_1 = b_1 \\ & x_1 \geq 0. \end{aligned} \tag{6}$$

However, instead of the second-stage being the end-of-the-line, the objective of the second stage problem, $V_2(x_1, \omega_2)$, is to minimize the cost of the second stage, plus the expected cost-to-go from stage three onward.

Therefore, we formulate a t-stage subproblem $V_t(x_{t-1}, \omega_t)$ which depends on the previous decision x_{t-1}:

$$
\begin{aligned}
V_t(x_{t-1}, \omega_t) = \min_{\bar{x}_{t-1}, x_t} \quad & c_t(\omega_t)^\top x_t + \mathbb{E}[V_{t+1}(x_t, \omega_{t+1})] \\
\text{s.t.} \quad & \bar{x}_{t-1} = x_{t-1} \quad [\nu] \\
& T_t(\omega_t)x_t + W_t(\omega_t)\bar{x}_{t-1} = g_t(\omega_t) \\
& x_t \geq 0,
\end{aligned}
$$

where $V_{T+1}(\cdot, \cdot) = 0$.

Assumption 3. *The problem has relatively complete recourse. That is, $V_t(x_{t-1}, \omega_t)$ has a finite optimal solution for all values of $\omega_t \in \Omega_t$ and feasible values of x_{t-1} in V_{t-1}.*

Importantly, relatively complete recourse means that we do not need to consider Benders feasibility cuts. (This is also a philosophic question: if, in many real-world problems, we have on the order of 10^{50} scenarios, how can we claim to find a feasible policy is we have not explored every scenario and added *all* the necessary feasibility cuts?)

In the two-stage case, we called each realization $\omega \in \Omega$ a scenario. In the multi-stage case, we call a sequence of realizations $\hat{\omega}_2, \ldots, \hat{\omega}_T$ a scenario.

Question How many possible scenarios are there?

Answer

$$
\text{number of scenarios } = \prod_{t=1}^{T} |\Omega_t|
$$

So, if we have 52 stages (one for each week), and 20 possible realizations of the uncertainty each week, then we have 20^{52} possible scenarios!

3.1 Example

A classical example of a multistage stochastic program is the hydro-thermal scheduling problem.

In a simplified version of the hydro-thermal scheduling problem, we assume the agent controls a hydro-electric generator and reservoir. We assume that there are three stages, $t = 1, 2, 3$, representing summer-fall, winter, and spring.

Each time period, they need to choose a generation quantity from thermal g_t, and hydro g_h, in order to meed demand ω_d, which is a stagewise-independent random variable.

The state variable, x_t, is the quantity of water in the reservoir at the start of each time period, and it has a minimum level of 5 units and a maximum level of 15 units. We assume that there are 10 units of water in the reservoir at the start of time, so that $x_0 = 10$.

The state-variable is connected through time by the water balance constraint: $x_t = x_{t-1} - g_h - s + \omega_i$, where x_t is the quantity of water at the end of the time period, x_{t-1} is the quantity of water at the start of the time period, s is the quantity of water spilled from the reservoir, and ω_i is a stagewise-independent random variable that represents the inflow into the reservoir during the time period.

In each stage, the agent incurs the cost of spillage, plus the cost of thermal generation. We assume that the cost of thermal generation is dependent on the stage $t = 1, 2, 3$, and that in each stage, ω is drawn from the set

$$
(\omega_i, \omega_d) = \{(0, 7.5), (3, 5), (10, 2.5)\},
$$

with equal probability.

$$
\begin{aligned}
V_t(x_{t-1}, \omega_t) = \min_{x_t, g_t, g_h, s} \quad & s + t \times g_t + \mathbb{E}[V_{t+1}(x_t, \omega_{t+1})] \\
\text{s.t.} \quad & x_t = x_{t-1} - g_t - s + \omega_i \\
& g_t + g_h = \omega_d \\
& 5 \leq x_t \leq 15 \\
& g_t, g_h, s \geq 0,
\end{aligned}
$$

with initial conditions $x_0 = 10$. The goal of the agent is to minimize $\mathbb{E}[V_1(x_0, \omega_1)]$.

This example seems trivial, but a more complicated version is widely used in practice. In fact, Brazil uses the hydro-thermal scheduling problem to set their national electricity prices! Read more about it at [2].

3.2 Algorithm

V_1 looks like the standard first-stage problem in Benders decomposition. Therefore, using the fact that V_t is convex w.r.t. x_{t-1}, we can form an approximated first-stage problem:

$$\begin{aligned} V_1^K = \min_{x_1,\theta_1} \quad & c_1^\top x_1 + \theta_1 \\ \text{s.t.} \quad & A_1 x_1 = b_1 \\ & x_1 \geq 0 \\ & \theta_1 \geq \alpha_{1,k} + \beta_{1,k}^\top x_1, \quad k = 1,\ldots,K. \end{aligned} \tag{7}$$

However, unlike standard Benders, the "second-stage" problem V_2 *also* has a cost-to-go component. Therefore, we can approximate each t-stage subproblem as follows:

$$\begin{aligned} V_t^K(x_{t-1},\omega_t) = \min_{\bar{x}_{t-1},x_t,\theta_t} \quad & c_t(\omega_t)^\top x_t + \theta_t \\ \text{s.t.} \quad & \bar{x}_{t-1} = x_{t-1} \quad [\nu] \\ & T_t(\omega_t)x_t + W_t(\omega_t)\bar{x}_{t-1} = g_t(\omega_t) \\ & x_t \geq 0 \\ & \theta_t \geq \alpha_{t,k} + \beta_{t,k}^\top x_t, \quad k = 1,\ldots,K, \end{aligned}$$

where $V_{T+1}(\cdot,\cdot) = 0$.

The question, then, is how to generate the cuts. *Stochastic dual dynamic programming* (SDDP) is an algorithm to do so. It is an iterative algorithm with two phases in each iteration:

1. A *forward* pass, which computes a sequence of decision x_t^* based on a single realization of the random variables ω_t from stage $t = 1$ to $T - 1$ (i.e., a scenario). This is equivalent to computing the first-stage decision in standard Benders decomposition.

2. A *backward* pass, which proceeds back up the sequence of stages and values x_t^* visited on the forward pass, and at each stage improves the approximation of the cost-to-go function with a Benders optimality cut.

Pseudo-code is given in Algorithm 2. Note that each forward pass requires $T - 1$ linear programming solves, and each backward pass requires $\sum_{t=2}^{T} |\Omega_t|$ solves.

Algorithm 2: The stochastic dual dynamic programming algorithm.

```
Set $K = 1$
while not converged do
    // Forward pass
    Solve $V_1^{K-1}$ and obtain solution $x_1^*$
    for $t = 2,\ldots,T-1$ do
        Sample $\omega_t$ from $\Omega_t$
        Solve $V_t(x_{t-1}^*,\omega_t)$ and obtain solution $x_t^*$
    end
    // Backward pass
    for $t = T,\ldots,2$ do
        for $\omega_t \in \Omega_t$ do
            Solve $V_t(x_{t-1}^*,\omega_t)$ and obtain extreme point primal-dual solution pair
            Set $V_{\omega_t}^*$ to the optimal objective value
            Set $\nu_{\omega_t}^*$ to the optimal dual vector $\nu$
        end
        set $\beta_{t,K} = \sum_{\omega_t \in \Omega_t} p_{\omega_t} \nu_{\omega_t}^*$
        set $\alpha_{t,K} = \sum_{\omega_t \in \Omega_t} p_{\omega_t} V_{\omega_t}^* - \beta^\top x_{t-1}^*$
        Add the cut $\theta_t \geq \alpha_{t,K} + \beta_{t,K}^\top x_t$ to $V_{t-1}^{K-1}$, creating $V_{t-1}^K$
    end
    $K \leftarrow K + 1$
end
```

3.3 Bounds

Obtaining a lower bound is easy, because it follows directly from Benders decomposition:

$$\underline{V} = V_1^K \leq V_1.$$

Obtaining an upper bound is *much* harder. A feasible solution needs a decision vector x for every stage in *every* possible scenario. Typical models in production (e.g., in Brazil) have 60^{20} scenarios! So is it possible to even compute such a solution? No! However, instead of an exact upper bound, we can compute a *statistical* upper bound using Algorithm 3. This is just a Monte Carlo simulation of the decisions, followed by a standard confidence interval.

Algorithm 3: Upper bounding algorithm for SDDP.

```
for $i = 1, \ldots, N$ do
    $\overline{z}_i = 0$
    Solve $V_1^{K-1}$ and obtain solution $x_1^*$
    for $t = 2, \ldots, T-1$ do
        Sample $\omega_t$ from $\Omega_t$
        Solve $V_t(x_{t-1}^*, \omega_t)$ and obtain solution $x_t^*$
        $\overline{z}_i = \overline{z}_i + c_t^\top x_t^*$
    end
end
$\mu = \frac{1}{N}\sum_{i=1}^N \overline{z}_i$
$\overline{V} = \mu \pm \frac{1.96}{\sqrt{N}}\sqrt{\frac{1}{N-1}\sum_{i=1}^N (\overline{z}_i - \mu)^2}$
```

3.4 Convergence

Unlike Benders, which converges after a finite number of iterations, stochastic dual dynamic programming converges *almost surely* after a finite number of iterations. We won't go into the details; see [3].

In practice, we use a number of stopping rules. These include:

- Running a fixed number of iterations;
- Running a fixed amount of time;
- Waiting until the lower-bound $\underline{V}$ "stalls" so that $\underline{V}^{K_1} \approx \underline{V}_{K_2}$ where $K_1 << K_2$;
- Waiting until the lower-bound is contained within the confidence interval for the upper bound.

Note that the last point is most-often used in the literature, but it has many, many problems. If you perform too few simulations, the variance of the confidence interval will be large and you will terminate early. If the *optimal* policy has a large variance, you may terminate too early. If you perform this check too often, then you will encounter a false positive for convergence.

I prefer the "run it as long as possible" stopping rule.

4 Beyond the basic algorithm

The previous section introduced stochastic dual dynamic programming, and showed how it is derived from Benders decomposition. It is also the way that it is typically presented in the literature because it is notationally convenient. (Even though, I hope you'll agree, there is a lot going on.) However, there is a more general way of formulating the problem.

Recall the constraint:

$$T_t\bar{x}_t + W_t x_t = g_t.$$

What if a column of T_t is 0?

- Changing corresponding $\bar{x}_t$ does not change feasibility of x_t!

- So corresponding ν is always 0.
- So cut coefficients are always 0.

Implication? Split vector x_t into $[u_t \; x_t]$, where u_t are *control* variables that don't appear in stage $t+1$.

Example State x is the quantity of inventory in a warehouse. Control u is quantity of goods to sell each day.

So we can formulate a more general stage problem:

$$\begin{aligned} V_t(x_{t-1}, \omega_t) = \min_{\bar{x}_t, u_t, x_t} \quad & C_t(\bar{x}_t, u_t, \omega_t) + \mathbb{E}[V_{t+1}(x_t, \omega_{t+1})] \\ \text{s.t.} \quad & \bar{x}_t = x_t \quad [\nu] \\ & x_t = T_t(\bar{x}_t, u_t, \omega_t) \\ & u_t \in U_t(\bar{x}_t, \omega_t), \end{aligned}$$

where $V_{T+1}(\cdot, \cdot) = 0$. Note that x_t doesn't appear as an argument to *stage-objective* C_t, but this is just for notational purposes. You can use the *transition function* T_t inside C_t if you wish. We call U_t the *feasibility set*.

This formulation is equivalent to the original formulation if everything is linear. However, SDDP can also be applied so long as the above is a convex program (with some additional assumptions, e.g., Slater).

The above formulation also explains why we introduce the so-called *fishing constraint* of $\bar{x}_t = x_t$; typically, the dimension of x_t is much less than the dimension of the control variables, and so the extra variables and constraints have a small impact on the problem size.

Even with this more general formulation, there are some issues. For example:

1. We assume that there are T stages
2. We assume that stage $t+1$ comes after stage t
3. The stagewise-independence of ω limits the types of stochastic processes we can represent.

To progress, we need to recognise that Figure 3 is a *linear* graph, each node has a subproblem V_t, and information (state variables x) flows between nodes along the arcs. Therefore, we propose to generalize multistage stochastic programming onto a *policy graph*. Read on to [1] for more details.

References

[1] Oscar Dowson. "The Policy Graph Decomposition of Multistage Stochastic Optimization Problems". In: *To appear in Networks* (2020). http://www.optimization-online.org/DB_HTML/2018/11/6914.html.

[2] MEP Maceiral et al. "Twenty years of application of stochastic dual dynamic programming in official and agent studies in Brazil-main features and improvements on the NEWAVE model". In: *2018 Power Systems Computation Conference (PSCC)*. IEEE. 2018, pp. 1–7.

[3] A. B. Philpott and Z. Guan. "On the Convergence of Sampling-Based Methods for Multi-Stage Stochastic Linear Programs". In: *Operations Research Letters* 36 (2008), pp. 450–455.