Problem 10.1: ADMM for Stochastic Linear Optimization Problems

Consider the following two-stage stochastic linear optimization problem

$$\zeta = \min_x . \ \left\{ c^\top x + \mathcal{Q}(x) : x \in X \right\}, \tag{1}$$

with the variables $x \in \mathbb{R}^{n_x}$ and known first-stage costs $c \in \mathbb{R}^{n_x}$. The set X consists of linear constraints on the variables x. The function $\mathcal{Q} : \mathbb{R}^{n_x} \to \mathbb{R}$ is the *expected recourse value*

$$\mathcal{Q}(x) = \mathbf{E}_\xi \left[\min_y . \ \left\{ q(\xi)^\top y : W(\xi) y = h(\xi) - T(\xi) x, \ y \in Y(\xi) \right\} \right] \tag{2}$$

with variables $y \in \mathbb{R}^{n_y}$. Values of the vectors $q(\xi) \in \mathbb{R}^{n_y}, h(\xi) \in \mathbb{R}^n$; matrices $W(\xi) \in \mathbb{R}^{n \times n_y}$, $T(\xi) \in \mathbb{R}^{n \times n_x}$; and the set $Y(\xi)$ all depend on realizations of a random variable ξ.

Suppose that ξ is associated with a discrete distribution indexed by a finite set $\mathcal{S}$, consisting of realizations $\xi_1, \dots, \xi_{|S|}$, corresponding to realization probabilities $p_1, \dots, p_{|S|}$. Each realization ξ_s of ξ is called a *scenario* and encodes realizations observed by the random elements

$$(q(\xi_s), h(\xi_s), W(\xi_s), T(\xi_s), Y(\xi_s))$$

To simplify notation, we refer to this collection of random elements respectively as

$$(q_s, h_s, W_s, T_s, Y_s)$$

For each scenario $s \in S$, the set Y_s consists of linear constraints on the variables $y_s \in \mathbb{R}^{n_y}$. We can reformulate problem (1) as an *equivalent deterministic problem*

$$\zeta = \min_{x,y} . \ \left\{ c^\top x + \sum_{s \in \mathcal{S}} p_s q_s^\top y_s : (x, y_s) \in K_s, \ \forall s \in S \right\}, \tag{3}$$

where

$$K_s = \left\{ (x, y_s) : W_s y_s = h_s - T_s x, \ x \in X, y_s \in Y_s \right\}.$$

Problem (3) has a decomposable structure that can be exploited. To induce this structure, let us introduce scenario-dependent copy variables x_s of the first-stage variable x for each scenario $s \in \mathcal{S}$. Using these copy variables, we can reformulate (3) as

$$\zeta = \min_{x,y,z} . \ \left\{ \sum_{s \in \mathcal{S}} p_s (c^\top x_s + q_s^\top y_s) : (x_s, y_s) \in K_s, x_s = z, \ \forall s \in \mathcal{S}, z \in \mathbb{R}^{n_x} \right\}. \tag{4}$$

The variable $z \in \mathbb{R}^{n_x}$ is a common global variable, and the constraints $x_s = z$ for all $s \in \mathcal{S}$ enforce *nonanticipativity* for the first-stage decisions: all first-stage decisions x_s must be the same (z) for each scenario $s \in \mathcal{S}$ in the final solution.

Relaxing the nonanticipativity constraints $x_s = z$ for all $s \in \mathcal{S}$ in (4) in Lagrangian fashion yields the following *augmented Lagrangian dual function*

$$\phi(\mu) = \min_{x,y,z} . \sum_{s \in \mathcal{S}} \left[p_s (c^\top x_s + q_s^\top y_s) + \mu_s^\top (x_s - z) + p_s \frac{\rho}{2} ||x_s - z||_2^2 \right] \tag{5}$$

$$\text{subject to: } (x_s, y_s) \in K_s, \ \forall s \in \mathcal{S} \tag{6}$$

By defining $v_s = \mu_s / p_s$ for all $s \in \mathcal{S}$, we can rewrite (5) – (6) as

$$\phi(v) = \min_{x,y,z} . \sum_{s \in \mathcal{S}} p_s L_s^\rho (x_s, y_s, z, v_s) \tag{7}$$

$$\text{subject to: } (x_s, y_s) \in K_s, \ \forall s \in \mathcal{S} \tag{8}$$

where $L_s^\rho(x_s, y_s, z, v_s)$, defined for each $s \in S$, is the *augmented Lagrangian*

$$L_s^\rho(x_s, y_s, z, v_s) = c^\top x_s + q_s^\top y_s + v_s^\top (x_s - z) + \frac{\rho}{2}||x_s - z||_2^2 \tag{9}$$

Derive the ADMM iterations for solving the problem (7) – (8) in a distributed fashion for each scenario $s \in \mathcal{S}$ separately.

Solution.

Since z is unconstrained in (7) – (8), the value of $\phi(v)$ can be made arbitrarily small unless

$$v_s^\top z = 0, \quad \text{for all } s \in \mathcal{S} \tag{10}$$

Therefore, to ensure that $\phi(v) > -\infty$, it is necessary that the condition (10) holds either by assumption or construction. Thus, the term $v_s^\top z$ vanishes from (9) for all $s \in S$.

Setting $v_s^\top z = 0$, for all $s \in S$ according to (10), we can rewrite (9) as

$$L_s^\rho(x_s, y_s, z, v_s) = (c + v_s)^\top x_s + q_s^\top y_s + \frac{\rho}{2}||x_s - z||_2^2. \tag{11}$$

In this case, the ADMM update step of (x_s, y_s) for all $s \in \mathcal{S}$ is of the form

$$\begin{aligned}(x_s^{k+1}, y_s^{k+1}) &= \underset{(x_s,y_s)\in K_s}{\text{argmin}} \; L_s^\rho(x_s, y_s, z^k, v_s^k) \\ &= \underset{(x_s,y_s)\in K_s}{\text{argmin}} \left\{(c + v_s^k)^\top x_s + q_s^\top y_s + \frac{\rho}{2}||x_s - z^k||_2^2\right\},\end{aligned} \tag{12}$$

which can be done in parallel for each scenario $s \in S$. Thus, computing (x_s^{k+1}, y_s^{k+1}) for each $s \in S$ amounts to solving a quadratic problem with linear constraints defined in $(x_s, y_s) \in K_s$. After updating x_s^{k+1} and y_s^{k+1} for each scenario $s \in S$, the $z-$update is of the form

$$\begin{aligned}z^{k+1} &= \underset{z}{\text{argmin}} \sum_{s\in S} p_s L_s^\rho(x_s^{k+1}, y_s^{k+1}, z, v_s^k) \\ &= \underset{z}{\text{argmin}} \sum_{s\in S} p_s \left[(c + v_s^k)^\top x_s^{k+1} + q_s^\top y_s^{k+1} + \frac{\rho}{2}||x_s^{k+1} - z||_2^2\right]\end{aligned} \tag{13}$$

Taking the gradient of (13) with regard to z and setting it to zero, we get

$$\begin{aligned}\sum_{s\in S} p_s \rho(x_s^{k+1} - z) &= 0 \\ \sum_{s\in S} p_s x_s^{k+1} - z \sum_{s\in S} p_s &= 0\end{aligned} \tag{14}$$

since $\sum_{s\in S} p_s = 1$, we get the following $z-$update from (14):

$$z^{k+1} = \sum_{s\in S} p_s x_s^{k+1} \tag{15}$$

Finally, the dual variables v_s are updated separately for each scenario $s \in S$ using Gradient Descent with a step size ρ as

$$v_s^{k+1} = v_s^k + \rho(x_s^{k+1} - z^{k+1}). \tag{16}$$

These updates can obviously be computed in parallel for each scenario $s \in S$.

To recap, we first update (x_s^{k+1}, y_s^{k+1}) for each scenario $s \in S$ separately (which can be done in parallel) by using (12). Each of these updates corresponds to solving a quadratic problem with linear constraints that can be solved, for instance, using the `Ipopt` solver in `JuMP`. Then, we update z^{k+1} simply by using (15). Finally, we update v_s^{k+1} for each scenario $s \in S$ using (16). These v_s^{k+1} updates for all $s \in S$ can also be computed in parallel.

The squared primal residual norm in this case is $p_s||r_s^{k+1}||_2^2 = p_s||x_s^{k+1} - z^{k+1}||_2^2$ for all $s \in S$ and the squared dual residual norm becomes $p_s||s^{k+1}||_2^2 = p_s||z^{k+1} - z^k||_2^2$. Summing these two yields

$$\begin{aligned}
&\sum_{s\in S} p_s \left[||x_s^{k+1} - z^{k+1}||_2^2 + ||z^{k+1} - z^k||_2^2\right] \\
=&\sum_{s\in S} p_s \left[(x_s^{k+1} - z^{k+1})^\top (x_s^{k+1} - z^{k+1}) + (z^{k+1} - z^k)^\top (z^{k+1} - z^k)\right] \\
=&\sum_{s\in S} p_s \left[||x_s^{k+1}||_2^2 - 2(x_s^{k+1})^\top z^{k+1} + ||z^{k+1}||_2^2 + ||z^{k+1}||_2^2 - 2(z^{k+1})^\top z^k + ||z^k||_2^2\right] \\
=&\sum_{s\in S} p_s \left[||x_s^{k+1}||_2^2 - 2||z^{k+1}||_2^2 + 2||z^{k+1}||_2^2 - 2(z^{k+1})^\top z^k + ||z^k||_2^2\right] \\
=&\sum_{s\in S} p_s \left[||x_s^{k+1}||_2^2 - 2(x_s^{k+1})^\top z^k + ||z^k||_2^2\right] \\
=&\sum_{s\in S} p_s ||x_s^{k+1} - z^k||_2^2
\end{aligned}$$

However, in Assignment 2, we will use as a stopping criterion the following non-squared sum of primal and dual residuals terms multiplied by ρ, which can make convergence smoother:

$$\sum_{s\in S} p_s \rho ||x_s^{k+1} - z^k||_2 \tag{17}$$

The sum terms in the stopping criterion (17) can be computed in parallel for each $s \in S$ after updating x_s^{k+1}. This is detailed in skeleton code of Assignment 2. The algorithm stops when

$$\sum_{s\in S} p_s \rho ||x_s^{k+1} - z^k||_2 < \epsilon$$

for some pre-defined tolerance $\epsilon > 0$.