PPENDIX I

Given problem (6), we now describe the SDDP approach which is a state-of-the-art method for solving multi-stage stochastic optimization. In the offline training algorithm, the post-decision value function $\bar{V}_t(\mathbf{l}_t, \boldsymbol{\nu}_{[t]}, \boldsymbol{\rho}_{[t]})$ is approximated by a collection of hyperplanes. Given this insights, in each iteration of the algorithm, we perform two steps; (i) in a forward pass from t = 0, trial solutions for each stage are generated by solving subproblems (18) using the current estimate of the future expected-cost-to-go function $\hat{V}_{t+1}(\mathbf{l}_{t+1}, \boldsymbol{\nu}_{[t+1]}, \boldsymbol{\rho}_{[t+1]})$; (ii) in a backward pass from $t = T$, each of the $\hat{V}_t(\mathbf{l}_t, \boldsymbol{\nu}_{[t]}, \boldsymbol{\rho}_{[t]})$ functions are then updated by adding cutting planes derived from the optimal actions $\boldsymbol{\pi}_t$ obtained in the forward pass. fter each iteration, the current $\hat{V}_{t+1}(\mathbf{l}_{t+1}, \boldsymbol{\nu}_{[t+1]}, \boldsymbol{\rho}_{[t+1]})$ provides a lower bound on the true optimal expected-cost-to-go function, which is being successively tightened.

---

lgorithm 2 Stochastic Dual Dynamic Programming

For $n = 1, \ldots, N$,

a): initialize $\ _t^1, \mathbf{b}_t^1, t = 1, \ldots, T \quad 1$ and generate M samples $\boldsymbol{\nu}_t^m, \rho_t^m, t = 0, \ldots, T \quad 1, m = 1, \ldots, M$, letting the initial values be $\boldsymbol{\nu}_0^m = \boldsymbol{\nu}_0, \rho_0^m = \rho_t, m = 1, \ldots, M$. lso, let the initial value of the pre-decision state be $\mathbf{l}_1^m = \mathbf{l}_1, m = 1, \ldots, M$.

b): **Forward pass**: For $t = 0, \ldots, T \quad 1$ and $m = 1, \ldots, M$

$$
\begin{aligned}
\hat{V}_t^m(\bar{\mathbf{l}}_t^m, \boldsymbol{\nu}_{[t]}^m, \boldsymbol{\rho}_{[t]}^m) = \max_{\boldsymbol{\pi}_{t\ \ 1} \in \hat{\Pi}_{t\ \ 1}} \ & C_{t+1}(\boldsymbol{\pi}_{t+1}, \bar{\mathbf{l}}_t^m + \boldsymbol{\nu}_t^m, \boldsymbol{\rho}_{[t+1]}^m) + \theta_{t+1} \\
\text{st} \ \ & \bar{\mathbf{l}}_{t+1} = \bar{\mathbf{l}}_t^m + \boldsymbol{\nu}_t^m + R\boldsymbol{\pi}_{t+1}, \quad [\boldsymbol{\omega}_t^m] \\
& \mathbf{l}^{min} \leq \bar{\mathbf{l}}_{t+1} + \boldsymbol{\nu}_{t+1}^m \leq \mathbf{l}^{max}, \\
& \boldsymbol{\pi}^{min} \leq \boldsymbol{\pi}_{t+1} \leq \boldsymbol{\pi}^{max}, \\
& \theta_{t+1} \leq (\ _{t+1}^k)^T (\bar{\mathbf{l}}_{t+1} + \boldsymbol{\nu}_{t+1}^m) + \mathbf{b}_{t+1}^k, k = 1,..,n
\end{aligned} \tag{18}
$$

let $\boldsymbol{\omega}_t^m$ and $\bar{\mathbf{l}}_{t+1}^m$ be the optimal value of dual variable $\boldsymbol{\omega}_t$ and primal variable $\bar{\mathbf{l}}_{t+1}$ in iteration m, respectively.

c): **Backward pass**: For $t = T \quad 1, \ldots, 0$,
update $\ _t^{n+1}$ and $\mathbf{b}_t^{n+1}$ as

$$
\begin{aligned}
\ _t^{n+1} &= \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{\omega}_t^j \\
\mathbf{b}_t^{n+1} &= \frac{1}{m} \sum_{j=1}^{m} [\hat{V}_t^j(\bar{\mathbf{l}}_t^j, \boldsymbol{\nu}_{[t]}^j, \boldsymbol{\rho}_{[t]}^j) \quad (\boldsymbol{\omega}_t^j)^T (\bar{\mathbf{l}}_t^j + \boldsymbol{\nu}_t^j)]
\end{aligned}
$$
