SCISPACE
formerly Typeset

Open access • Journal Article • DOI:10.1007/S10107-019-01368-1

MIDAS: A mixed integer dynamic approximation scheme — Source link

Andy Philpott, F. Wahid, Joseph Frédéric Bonnans

Institutions: University of Auckland, École Polytechnique

Published on: 01 May 2020 - Mathematical Programming (Springer Berlin Heidelberg)

Topics: Dynamic programming, Monotonic function, Integer (computer science) and Stochastic optimization

Related papers:

- Multi-stage stochastic optimization applied to energy planning
- Analysis of stochastic dual dynamic programming method
- Stochastic dual dynamic integer programming
- Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs
- On the convergence of stochastic dual dynamic programming and related methods

Share this paper:

View more about this paper here: https://typeset.io/papers/midas-a-mixed-integer-dynamic-approximation-scheme-578m8xwqsy

HAL open science

MIDAS: A Mixed Integer Dynamic Approximation Scheme

Andy Philpott, Faisal Wahid, Frédéric Bonnans

To cite this version:

Andy Philpott, Faisal Wahid, Frédéric Bonnans. MIDAS: A Mixed Integer Dynamic Approximation Scheme. [Research Report] Inria Saclay Ile de France. 2016, pp.22. hal-01401950

HAL Id: hal-01401950

https://hal.inria.fr/hal-01401950

Submitted on 24 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MIDAS: A Mixed Integer Dynamic Approximation Scheme *

Andy Philpott[†], Faisal Wahid[‡], Frédéric Bonnans[§]

June 16, 2016

Abstract

Mixed Integer Dynamic Approximation Scheme (MIDAS) is a new sampling-based algorithm for solving finite-horizon stochastic dynamic programs with monotonic Bellman functions. MIDAS approximates these value functions using step functions, leading to stage problems that are mixed integer programs. We provide a general description of MIDAS, and prove its almost-sure convergence to an ε-optimal policy when the Bellman functions are known to be continuous, and the sampling process satisfies standard assumptions.

1 Introduction

In general, multistage stochastic programming problems are extremely difficult to solve. If one discretizes the random variable using a finite set of outcomes in each stage and represents the process as a scenario tree, then the problem size grows exponentially with the number of stages and outcomes [9]. On the other hand, if the random variables are stagewise independent then the problem can be formulated as a dynamic programming recursion, and attacked using an approximate dynamic programming algorithm.

When the Bellman functions are known to be convex (if minimizing) or concave (if maximizing) then they can be approximated by cutting planes. This is the basis of the popular Stochastic Dual Dynamic Programming (SDDP) algorithm originally proposed by Pereira and Pinto [7]. This creates a sequence of cutting-plane outer approximations to the Bellman function at each stage by evaluating stage problems on independently sampled sequences of random outcomes. A comprehensive recent summary of this algorithm and its properties is provided by [10]. A number of algorithms that are variations of this idea have been proposed (see e.g. [3, 4, 6]).

The first formal proof of almost-sure convergence for SDDP-based algorithms was provided by Chen and Powell in [3] for their CUPPS algorithm. Their proof however relied on a key unstated assumption

*This research was carried out with financial support from PGMO, EDF and Meridian Energy. The authors wish to acknowledge the contributions of Anes Dallagi and Cedric Gouvernet of EDF for discussions on an earlier version of this paper.

[†]University of Auckland

[‡]University of Auckland

[§]Inria & École Polytechnique

identified by Philpott and Guan in [8], who provided a new proof of almost-sure convergence for problems with polyhedral Bellman functions without requiring this assumption. Almost-sure convergence for general convex Bellman functions was recently proved using a different approach by Girardeau et al. [5].

In order to guarantee the validity of the outer approximation of the Bellman functions, SDDP-based methods require these functions to be convex (if minimizing) or concave (if maximizing). However, there are many instances of problems when the optimal value of the stage problem is not a convex (or concave) function of the state variables. For example, if the stage problem involves state variables with components appearing in both the objective function and the right-hand side of a convex optimization problem then the optimal value function might have a saddle point. More generally, if the stage problems are not convex then we cannot guarantee convexity of the optimal value function. This will happen when stage problems are mixed-integer programs (MIPS), or when they incorporate nonlinear effects, such as modeling head effects in hydropower production functions [2].

Our interest in problems of this type is motivated by models of hydroelectricity systems in which we seek to maximize revenue from releasing water through generating stations on a river chain. This can be modeled as a stochastic control problem with discrete-time dynamics:

$$x_{t+1} = f_t(x_t, u_t, \xi_t),\ x_1 = \bar{x},\ t = 1, 2, \ldots, T.$$

In this problem, $u_t \in U(x_t)$ is a control and ξ_t is a random noise term. We assume that f_t is a continuous function of its arguments. The control u_t generates a reward $r_t(x_t, u_t)$ in each stage and a terminal reward $R(x_{T+1})$, where r_t and R are continuous and monotonic increasing in their arguments. Given initial state $\bar{x}$, we seek an optimal policy yielding $V_1(\bar{x})$, where

$$\begin{aligned} V_t(x) &= \mathbb{E}_{\xi_t}\left[\max_{u \in U(x)} \{r_t(x, u, \xi_t) + V_{t+1}(f_t(x, u, \xi_t))\}\right] \\ V_{T+1}(x) &= R(x). \end{aligned}$$

Here $V_t(x)$ denotes the maximum expected reward from the beginning of stage t onwards, given the state is x, and we take action u_t after observing the random disturbance ξ_t. We assume that $U(x)$ is sufficiently regular so that V_t is continuous if V_{t+1} is.

For example, in a single-reservoir hydro-scheduling problem, $x = (s, p)$ might represent both the reservoir stock variable s and a price variable p, and $u = (v, l)$ represents the reservoir release v through a generator and reservoir spill l. In this case the dynamics might be represented by

$$\begin{bmatrix} s_{t+1} \\ p_{t+1} \end{bmatrix} = \begin{bmatrix} s_t - v_t - l_t + \omega_t \\ \alpha_t p_t + (1 - \alpha_t) b_t + \eta_t \end{bmatrix},$$

where ω_t is (random) reservoir inflow, and η_t is the error term for an autoregressive model of price, so $\xi_t = [\ \omega_t \quad \eta_t\ ]^\top$. Here we might define

$$r_t(s, p, v, l, \omega_t, \eta_t) = pg(v),$$

giving the revenue earned by released energy $g(v)$ sold at price p, and $U(x) = U_0 \cap [0, x]$. For such a model, it is easy to show (under the assumption that we can spill energy without penalty) that $V_t(s, p)$

is continuous and monotonic increasing in s and p. On the other hand $V_t(s,p)$ is not always concave which makes the application of standard SDDP invalid.

A number of authors have looked to extend SDDP methods to deal with non-convex stage problems. The first approach replaces the non-convex components of the problem with convex approximations, for example using McCormick envelopes to approximate the production function of hydro plants as in [2]. The second approach convexifies the value function, e.g. using Lagrangian relaxation techniques. A recent paper [12] by Thomé et al. proposes a Lagrangian relaxation of of the SDDP subproblem, and then uses the Lagrange multipliers to produce valid Benders cuts. A similar approach is adopted by Steeger and Rebennack in [11]. Abgottspon et al [1] introduce a heuristic to add locally valid cuts that enhance a convexified approximation of the value function.

In this paper we propose a new extension of SDDP called Mixed Integer Dynamic Approximation Scheme (MIDAS). MIDAS uses the same algorithmic framework as SDDP but, instead of using cutting planes, MIDAS uses step functions to approximate the value function. The approximation requires an assumption that the value function is monotonic in the state variables. In this case each stage problem can be solved to global optimality as a MIP. We show that if the true Bellman function is continuous then MIDAS converges almost surely to a $(T+1)\varepsilon$-optimal solution for a problem with T stages.

The rest of the paper is laid out as follows. We begin by outlining the approximation of $V_t(x)$ that is iuswed by MIDAS. In section 3 we prove the convergence of MIDAS to a $(T+1)\varepsilon$-optimal solution to a multistage deterministic optimization problem. Lastly, in section 4, we extend our proof to a sampling-based algorithm applied to the multistage stochastic optimization problem, and demonstrate its almost-sure convergence. A simple hydro-electric scheduling example illustrating the algorithm is presented in section 5. We conclude the paper in section 6 by discussing the potential value of using MIDAS to solve stochastic multistage integer programs.

2 Approximating the Bellman function

The MIDAS algorithm approximates each Bellman function $V_t(x)$ by a piecewise constant function $Q_t^H(x)$, which is updated in a sequence (indexed by H) of forward and backward passes through the stages of the problem being solved. Since the terminal value function $R(x)$ is continuous, the Bellman functions $V_t(x)$, $t = 1, 2, \ldots, T+1$ will be continuous. Moreover since X is compact we have

Proposition 1. *There exists $\varepsilon > 0$ and $\delta > 0$ such that for every $t = 1, 2, \ldots, T+1$, and x,y with $\|x - y\|_\infty < \delta$,*

$$|V_t(x) - V_t(y)| < \varepsilon.$$

From now on we shall choose a fixed $\delta > 0$ and $\varepsilon > 0$ so that Proposition 1 holds. For example if $V_t(x)$, $t = 1, 2, \ldots, T+1$ can be shown to be Lipschitz with constant K, then we can choose $\delta > 0$ and $\varepsilon = K\delta$.

MIDAS updates the value function at points x^h that are the most recently visited values of the state variables computed using a forward pass of the algorithm. Consider a selection of points $x^h \in X$,

$h = 1, 2, \ldots, H$, with $\left\|x^{h_i} - x^{h_j}\right\|_\infty > \delta$, $h_i \neq h_j$. Suppose that we have computed values

$$q^h = Q(x^h),\ h = 1, 2, \ldots, H, \tag{1}$$

where Q is any monotonic increasing upper semi-continuous function. Since X is compact, we can define $M = \max_{x \in X} Q(x)$. For each $x \in X$, we define $\mathcal{H}_\delta(x) \subseteq \{1, 2, \ldots, H\}$ by

$$\mathcal{H}_\delta(x) = \{h' : x_i^{h'} > x_i - \delta,\ i = 1, 2, \ldots, n\},$$

and

$$Q^H(x) = \min\left\{M, \min\left\{q^{h'} : h' \in \mathcal{H}_\delta(x)\right\}\right\}. \tag{2}$$

We call $\mathcal{H}_\delta(x)$ the *supporting indices* of Q^H at x. Observe that if $x_1 \leq x_2$ then $\mathcal{H}_\delta(x_2) \subseteq \mathcal{H}_\delta(x_1)$, so Q^H is monotonic increasing. Since the sets of supporting indices are also nested by inclusion as H increases, it is easy to see that for every H,

$$Q^{H+1}(x) \leq Q^H(x), \quad x \in X, \tag{3}$$

and $h \in \mathcal{H}_\delta(x^h)$ implies

$$Q^H(x^h) \leq q^h, h = 1, 2, \ldots, H. \tag{4}$$

Suppose that there was some $h' \in \mathcal{H}_\delta(x^h)$ with $q^{h'} < q^h$, so $Q^H(x^h) < q^h$. Although $x_i^{h'} > x_i^h - \delta$, $i = 1, 2, \ldots, n$, this does not violate monotonicity since $\delta > 0$. However when $q^h = R(x^h)$ then there is a bound on the difference between $Q^H(x^h)$ and q^h given by the following lemma.

Lemma 1. *Suppose*

$$q^h = R(x^h),\ h = 1, 2, \ldots, H. \tag{5}$$

Then for any $x \in X$,

$$Q^H(x) \geq R(x) - \varepsilon,$$

and for any $h = 1, 2, \ldots, H$,

$$Q^H(x^h) \geq q^h - \varepsilon.$$

Proof. Consider $x \in X$. Observe that $h \in \mathcal{H}_\delta(x)$ implies $x_i < x_i^h + \delta$, $i = 1, 2, \ldots, n$, so there is some $y \geq x$ with $\left\|y - x^h\right\|_\infty < \delta$ and

$$R(x) \leq R(y) \leq R(x^h) + \varepsilon$$

by virtue of continuity. Recalling (5) and (2) yields the first inequality, from which the second follows immediately upon substituting (5). ■

Two examples of the approximation of $Q(x)$ by $Q^H(x)$ are given in Figure 1 and Figure 2 .

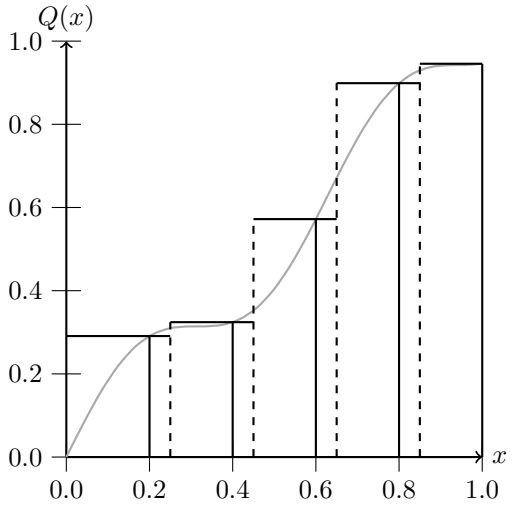

Figure 1: Approximation of $Q(x) = x + 0.1\sin(10x)$ shown in grey, by piecewise constant $Q^H(x)$, shown in black. Here $\delta = 0.05$ and $x^h = 0.2, 0.4, 0.6, 0.8.$.

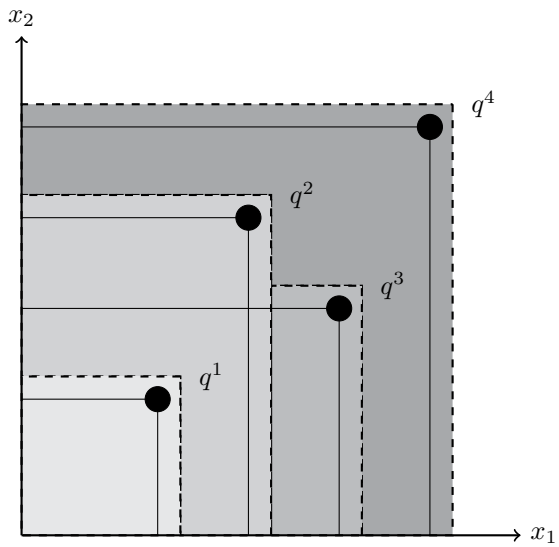

Figure 2: Contour plot of $Q_H(x)$ when $H = 4$. The circled points are x^h, $h = 1, 2, 3, 4$. The darker shading indicates increasing values of $Q^H(x)$ (which equals $Q(x^h)$ in each region containing x^h.)

The definition of Q_H in terms of supporting indices is made for notational convenience. In practice $Q_H(x)$ is computed using mixed integer programming (hence the name MIDAS). We propose one possible formulation for doing this in the Appendix.

3 Multistage optimization problems

In Section 2 we discussed how MIDAS approximates the value function. In this section we describe the MIDAS algorithm for a deterministic multistage optimization problem, and prove its convergence.

Given an initial state $\bar{x}$, we consider a deterministic optimization problem of the form:

$$\begin{array}{llrll} \text{MP:} & \max\limits_{x,u} & \sum\limits_{t=1}^{T} r_t(x_t,u_t) & + & R(x_{T+1}) \\ \text{s.t.} & & x_{t+1} = f_t(x_t,u_t), & & t=1,2,\ldots,T \\ & & x_1 = \bar{x}, & & \\ & & u_t \in U_t(x_t), x_t \in X_t, & & t=1,2,\ldots,T. \end{array}$$

This gives a dynamic programming formulation

$$\begin{aligned} V_t(x) &= \max_{u\in U(x)} \{r_t(x,u) + V_{t+1}(f_t(x,u))\}, \\ V_{T+1}(x) &= R(x), \end{aligned}$$

where we seek a policy that maximizes $V_1(\bar{x})$. We assume that $r_t(x,u)$ and $f_t(x,u))$ are continuous and increasing in x and for every t, $V_t(x)$ is a monotonic increasing continuous function of x, so that Proposition 1 holds.

The deterministic MIDAS algorithm (Algorithm 1) applied to MP is as follows.

Algorithm 1 Deterministic MIDAS

1. *Set* $H=1$, $Q_t^H(x) = M$, an upper bound on $V_t(x)$, $t=1,2,\ldots,T+1$.
2. Forward pass: Set $x_1^H = \bar{x}$. For $t=1$ to T,
   (a) Solve $\max\limits_{u\in U(x_t^H)} \left\{r_t(x_t^H,u) + Q_{t+1}^H(f_t(x_t^H,u))\right\}$ to give u_t^H;
   (b) If $\left\|f_t(x_t^H,u_t^H) - x_{t+1}^h\right\|_\infty < \delta$ for $h<H$ then set $x_{t+1}^H = x_{t+1}^h$, else set $x_{t+1}^H = f_t(x_t^H,u_t^H)$.
3. If there is some $h<H$ with $x_{T+1}^H = x_{T+1}^h$ then stop.
4. Backward pass: Update $Q_{T+1}^H(x)$ to $Q_{T+1}^{H+1}(x)$ by adding $q_{T+1}^H = R(x_{T+1}^H)$ at point x_{T+1}^H. Then for every $t=T$ down to 1 do
   (a) Solve $\varphi = \max\limits_{u\in U(x_t^H)} \left\{r_t(x_t^H,u) + Q_{t+1}^{H+1}(f_t(x_t^H,u))\right\}$;
   (b) Update $Q_t^H(x)$ to $Q_t^{H+1}(x)$ by adding $q_t^H = \varphi$ at point x_t^H.
5. Increase H by 1 and go to step 2.

Observe that the first forward pass of Algorithm 1 follows a myopic strategy. We now show that this algorithm will converge in step 3 after a finite number of iterations to a solution $(u_1^H, u_2^H, \ldots, u_T^H)$ that is a $(T+1)\varepsilon$-optimal solution to the dynamic program starting in state $\bar{x}$. First observe that the sequence $\{(x_1^H, x_2^H, \ldots, x_T^H)\}$ lies in a compact set, so it has some convergent subsequence. It follows that we can find H and $h \leq H$ with $\left\| x_t^{H+1} - x_t^h \right\|_\infty < \delta$ for every $t = 1, 2, \ldots, T+1$. We demonstrate this using the following two lemmas.

Lemma 2. *Let $N_\delta(S) = \{x : \|x - s\|_\infty < \delta, s \in S\}$. Then any infinite set S lying in a compact set contains a finite subset Z with $S \subseteq N_\delta(Z)$.*

Proof. Consider an infinite set S lying in a compact set X. Since S is a subset of the closed set X, its closure $\bar{S}$ lies in X, and so $\bar{S}$ is compact. Now for some fixed $\delta > 0$, it is easy to see that $N_\delta(S)$ is a open covering of $\bar{S}$, since any limit point of S must be in $N_\delta(\{s\})$ for some $s \in S$. Since $\bar{S}$ is compact, there exists a finite subcover, i.e. a finite set $Z = \{z_1, z_2, \ldots, z_N\} \subseteq S$ with $S \subseteq \bar{S} \subseteq \cup_{k=1}^N N_\delta(z_k)$. ■

Lemma 3. *For every $t = 1, 2, \ldots, T+1$ and for all iterations H,*

$$Q_t^H(x) \geq V_t(x) - (T + 2 - t)\varepsilon.$$

Proof. We prove the result by induction. First observe that $V_{T+1}(x) = R(x)$ in (1), so Lemma 1 yields

$$Q_{T+1}^H(x) \geq V_{T+1}(x) - \varepsilon,$$

for all iterations H, which shows the result is true for $t = T+1$. Now suppose for some $t \leq T+1$ that (3) holds for every iteration H and every $x \in X$, and let

$$u_{t-1}^* \in \arg\max\{r_{t-1}(x_{t-1}^h, u) + V_t(f_{t-1}(x_{t-1}^h, u))\}.$$

It follows that for $h = 1, 2, \ldots, H$,

$$\begin{aligned} q_{t-1}^h &= \max_{u \in U(x_{t-1}^h)} \left\{ r_{t-1}(x_{t-1}^h, u) + Q_t^h(f_{t-1}(x_{t-1}^h, u)) \right\} \\ &\geq r_{t-1}(x_{t-1}^h, u_{t-1}^*) + Q_t^h(f_{t-1}(x_{t-1}^h, u_{t-1}^*)) \\ &\geq r_{t-1}(x_{t-1}^h, u_{t-1}^*) + V_t(f_{t-1}(x_{t-1}^h, u_{t-1}^*)) - (T + 2 - t)\varepsilon \\ &= V_{t-1}(x_{t-1}^h) - (T + 2 - t)\varepsilon \end{aligned}$$

Now consider an arbitrary $x \in X$, and suppose $h \in \mathcal{H}_\delta(x)$. Then $x < x_{t-1}^h + \delta\mathbf{1}$, so

$$V_{t-1}(x) \leq V_{t-1}(x_{t-1}^h) + \varepsilon,$$

giving

$$q_{t-1}^h \geq V_{t-1}(x) - (T + 2 - t)\varepsilon - \varepsilon.$$

By definition

$$Q_{t-1}^H(x) = \min \left\{ q_{t-1}^h : h \in \mathcal{H}_\delta(x) \right\},$$

so (3) holds for $t - 1$. ■

Lemma 4. *Suppose there is some $h < H$ with $\left\|x^H_{T+1} - x^h_{T+1}\right\|_\infty < \delta$. Then the sequence of actions $(u^H_1, u^H_2, \ldots, u^H_T)$ is a $(T+1)\varepsilon$-optimal solution to the dynamic program starting in state $\bar{x}$.*

Proof. Suppose there is some $h < H$ with $\left\|x^H_{T+1} - x^h_{T+1}\right\|_\infty < \delta$. Then Algorithm 2 stops in Step 3. At this point we have a sequence of actions $(u^H_1, u^H_2, \ldots, u^H_T)$, and a sequence of states $(\bar{x} = x^H_1, x^H_2, \ldots, x^H_T, x^H_{T+1})$, with $x^H_{t+1} = f_t(x^H_t, u^H_t)$ and

$$u^H_t \in \arg\max_{u \in U(x^H_t)} \left\{ r_t(x^H_t, u) + Q^H_{t+1}(f_t(x^H_t, u)) \right\}.$$

We show by induction that for every $t = 1, 2, \ldots, T$

$$r_t(x^H_t, u^H_t) + V^H_{t+1}(f_t(x^H_t, u^H_t)) \geq V_t(x^H_t) - (T + 2 - t)\varepsilon, \tag{6}$$

$$Q^H_{t+1}(x^H_{t+1}) \leq V_{t+1}(x^H_{t+1}) + \varepsilon. \tag{7}$$

When $t = T$, since $h < H$, we have by (3)

$$\begin{aligned} Q^H_{T+1}(x^H_{T+1}) &\leq Q^h_{T+1}(x^H_{T+1}) \\ &= Q^h_{T+1}(x^h_{T+1}) \\ &= V_{T+1}(x^h_{T+1}) \\ &\leq V_{T+1}(x^H_{T+1}) + \varepsilon \end{aligned}$$

where the last inequality holds because $\left\|x^H_{T+1} - x^h_{T+1}\right\|_\infty < \delta$. Now let

$$u^* \in \arg\max_{u \in U(x^H_T)} \left\{ r_T(x^H_T, u) + V_{T+1}(f_T(x^H_T, u)) \right\}$$

so

$$V_T(x^H_T) = r_T(x^H_T, u^*) + V_{T+1}(f_T(x^H_T, u^*)).$$

Then

$$\begin{aligned} r_T(x^H_T, u^H_T) + Q^H_{T+1}(f_T(x^H_T, u^H_T)) &\geq r_T(x^H_T, u^*) + Q^H_{T+1}(f_T(x^H_T, u^*)) \\ &\geq r_T(x^H_T, u^*) + V_{T+1}(f_T(x^H_T, u^*) - \varepsilon \end{aligned}$$

by Lemma 3. Also we have just shown that (7) holds for $t = T$, so

$$r_T(x^H_T, u^H_T) + V_{T+1}(x^H_{T+1}) \geq V_T(x^H_T) - 2\varepsilon$$

yielding (6) in the case $t = T$.

Now suppose (6) and (7) hold at t. By (4)

$$Q^H_t(x^H_t) \leq q^H_t = r_t(x^H_t, u^H_t) + Q^H_{t+1}(f_t(x^H_t, u^H_t)),$$

so by (7)

$$\begin{aligned} Q^H_t(x^H_t) &\leq r_t(x^H_t, u^H_t) + V_{t+1}(f_t(x^H_t, u^H_t)) + \varepsilon \\ &\leq V_t(x^H_t) + \varepsilon \end{aligned}$$

so (7) holds at $t-1$. And

$$\begin{aligned} r_{t-1}(x_{t-1}^H, u_{t-1}^H) + Q_t^H(f_{t-1}(x_{t-1}^H, u_{t-1}^H)) &\geq r_{t-1}(x_{t-1}^H, u_{t-1}^*) + Q_t^H(f_{t-1}(x_{t-1}^H, u_{t-1}^*)) \\ &\geq r_{t-1}(x_{t-1}^H, u_{t-1}^*) + V_t(f_{t-1}(x_{t-1}^H, u_{t-1}^*)) \\ &\quad -(T+2-t)\varepsilon \end{aligned}$$

by Lemma 3. Combining with (7) gives

$$r_{t-1}(x_{t-1}^H, u_{t-1}^H) + V_t(x_t^H) \geq V_{t-1}(x_{t-1}^H) - (T+2-(t-1))\varepsilon$$

establishing (6) for $t-2$.

Since $x_1^H = \bar{x}$, It follows by induction that

$$r_1(\bar{x}, u_1^H) + V_2(f_1(\bar{x}, u_1^H)) = V_1(\bar{x}) - (T+1)\varepsilon$$

showing that action u_1^H is the first stage decision of $\{u_t^H, t = 1, 2, \ldots, T\}$ which defines a $(T+1)\varepsilon$-optimal policy. ■

4 Multistage stochastic optimization problems

We extend model MP, described in Section 3, to include random noise ξ_t on the state transition function $f_t(x_t, u_t, \xi_t)$. This creates the following multistage stochastic program MSP.

$$\begin{array}{llll} \text{MSP:} & \max & \mathbb{E}_{\xi_t}\left[\sum_{t=1}^{T} r_t(x_t, u_t) + R(x_{T+1})\right] & \\ & \text{s.t.} & x_{t+1} = f_t(x_t, u_t, \xi_t), & t = 1, 2 \ldots, T, \\ & & x_1 = \overline{x} & \\ & & u_t \in U(x_t), & t = 1, 2 \ldots, T, \\ & & x_t \in X_t, & t = 1, 2 \ldots, T. \end{array}$$

We make the following assumptions:

1. ξ_t is stagewise independent.
2. The set Ω_t of random outcomes in each stage $t = 1, 2, \ldots, T$ is discrete and finite
3. X_t is a compact set.

The particular form of ξ_t we assume yields a finite set of possible sample paths or scenarios, so it is helpful for exposition to view the random process ξ_t as a *scenario tree* with nodes $n \in \mathcal{N}$ and leaves in $\mathcal{L}$, where n represents different future states of the world, and ξ_n denotes the realization of ξ_t in node n. Each node n has a probability $p(n)$. By convention we number the root node $n = 0$ (with

$p(0) = 1$). The unique predecessor of node $n \neq 0$ is denoted by $n-$. We denote the set of children of node $n \in \mathcal{N} \setminus \mathcal{L}$ by $n+$, and let $M_n = |(n+)|$. The depth $d(n)$ of node n is the number of nodes on the path from node n to node 0, so $d(0) = 1$ and we assume that every leaf node has the same depth, say $d_{\mathcal{L}}$. The depth of a node can be interpreted as a time index $t = 1, 2, \ldots, T + 1 = \delta_{\mathcal{L}}$.

The formulation of MSP in $\mathcal{N}$ becomes

$$\begin{array}{llll} \text{MSPT:} & \max & \sum_{n\in\mathcal{N}\setminus\{0\}} p(n) r_n(x_{n-}, u_n) + \sum_{n\in\mathcal{L}} p(n) R\left(x_n\right) & \\ & \text{s.t.} & x_n = f_{n-}(x_{n-}, u_n, \xi_n), & n \in \mathcal{N} \setminus \{0\}, \\ & & x_0 = \overline{x}, & \\ & & u_n \in U(x_n), & n \in \mathcal{N}, \\ & & x_n \in X_n, & n \in \mathcal{N}. \end{array}$$

Observe in MSPT that we have a choice between hazard-decision and decision-hazard formulations that was not relevant in the deterministic problem. To be consistent with most implementations of SDDP, we have chosen a hazard-decision setting. This means that u is chosen in node n after the information from node n is revealed. In general, this could include information about the stage's reward function (e.g. if the price was random) so we write $r_n(x_{n-}, u_n)$. Alternatively if the reward function were not revealed at stage n, then we would write $r_{n-}(x_{n-}, u_n)$ instead. The discussion that follows can be adapted to this latter case with a minor change of notation.

A recursive form of MSPT is:

$$\begin{array}{lcll} V_n(x_n) & = & \sum\limits_{m\in n+} \frac{p(m)}{p(n)} \max_{u\in U(x_n)} \left\{ r_m(x_n, u) + V_m\left(f_n(x_n, u, \xi_m)\right) \right\} & n \in \mathcal{N} \setminus \mathcal{L}, \\ V_n(x_n) & = & R(x_n), & n \in \mathcal{L}. \end{array}$$

where we seek a policy that maximizes $V_0(\bar{x})$.

The MIDAS algorithm applied to the scenario tree gives Algorithm 2 described below:

Algorithm 2 Sampled MIDAS

Set $H = 1$, $Q_n^H(x) = M$, $n \in \mathcal{N}$. While $H < H_{max}$ do

1. Forward pass: Set $x_0^H = \overline{x}$, and $n = 0$. While $n \notin \mathcal{L}$:
   (a) Sample $m \in n+$ to give ξ_m^H;
   (b) Solve $\max_{u \in U(x_n^H)} \left\{ r_m(x_n^H, u) + Q_m^H(f_n(x_n^H, u, \xi_m^H)) \right\}$ to give u_m^H;
   (c) If $\left\| f_n(x_n^H, u_m^H, \xi_m^H) - x_m^h \right\|_\infty < \delta$ for $h < H$ then set $x_m^{H+1} = x_m^h$, else set $x_m^{H+1} = f_n(x_n^H, u_m^H, \xi_m^H)$;
   (d) Set $n = m$.
2. Backward pass: For the particular node $n \in \mathcal{L}$ at the end of step 1 update $Q_n^H(x)$ to $Q_n^{H+1}(x)$ by adding $q_n^{H+1} = Q(x_n^H)$ at point x_n^{H+1}. While $n > 0$
   (a) Set $n = n-$;
   (b) Compute
   $$\varphi = \sum_{m \in n+} \frac{p(m)}{p(n)} \left[\max_{u \in U(x_n^H)} \left\{ r_m(x_n^{H+1}, u) + Q_m^{H+1}(f_n(x_n^{H+1}, u, \xi_m)) \right\} \right]$$
   (c) Update $Q_n^H(x)$ to $Q_n^{H+1}(x)$ by adding $q_n^{H+1} = \varphi$ at point x_n^{H+1};
3. Increase H by 1 and go to step 1.

Observe that Sampled MIDAS has no stopping criterion. However we have the following result.

Lemma 5. *There is some H^* such that for every $H > H^*$, and every $m \in n+$, $n \in \mathcal{N} \setminus \mathcal{L}$,*

$$\left\| f_n(x_n^H, u_m^H, \xi_m^H) - x_m^h \right\|_\infty < \delta \text{ for some } h < H^*.$$

Proof. Let $\tilde{x} = \{x_n, n \in \mathcal{N}\}$ denote the collection of state variables generated by the current policy. Each forward pass H updates values of x_n (from x_n^H to x_n^{H+1}) for nodes n corresponding to the realization of ξ_n^H on the sample path H. For nodes n not on this sample path we let $x_n^{H+1} = x_n^H$, thus defining $\tilde{x}^{H+1}$. This defines an infinite sequence $S = \{\tilde{x}^H\}_{H=1.2....}$ of points, all of which lie in a compact set. By Lemma 2 S contains a finite subsequence Z with $S \subseteq N_\delta(Z)$. Let H^* be the largest index in Z. Suppose it is not true that for every $n \in \mathcal{N}$ and every $H > H^*$ there is some $h \leq H^*$ with $\left\| f_t(x_t^H, u_t^H, \xi_t) - x_{t+1}^h \right\|_\infty < \delta$. Then for some node $n \in \mathcal{N}$, and some $H > H^*$, $\left\| f_n(x_n^H, u_m^H, \xi_m^H) - x_m^h \right\|_\infty \geq \delta$ for every $h \leq H^*$, so $\left\| x_n^{H+1} - x_m^h \right\|_\infty \geq \delta$, giving $\tilde{x}_m^{H+1} \notin N_\delta(Z)$, a contradiction. ■

By Lemma 5, after H^* iterations every sample path produced in step 1 yields state values that have been visited before. It is clear that no additional points will be added in subsequent backward passes and so

the MIDAS algorithm will simply replicate points that it has already computed with no improvement in its approximation of $Q_n(x)$ at each n. If this situation could be identified then we might terminate the algorithm at this point. This then raises the question of whether the solution obtained at this point is a $(T+1)\varepsilon$-optimal solution to MSP. The answer to this latter question depends on the sampling strategy used in step 1a.

Following [8] we assume that this satisfies the Forward Pass Sampling Property.

Forward Pass Sampling Property (FPSP):

For each $n \in \mathcal{L}$, with probability 1

$$\left|\left\{H : \xi_n^H = \xi_n\right\}\right| = \infty.$$

FPSP states that each leaf node in the scenario tree is visited infinitely many times with probability 1 in the forward pass. There are many sampling methods satisfying this property. For example, one sampling method would be independently sampling a single outcome in each stage with a positive probability for each outcome ξ_m in the forward pass, which meets FPSP by the Borel-Cantelli lemma. Another sampling method that satisfies FPSP is to repeat an exhaustive enumeration of each scenario in the forward pass.

Recall that $d(n)$ is the depth of node n. The following result ensures that Q_n^H is a $(T+2-d(n))\varepsilon$-upper bound on V_n.

Lemma 6. *For every $n \in \mathcal{N}$, and for all iterations H,*

$$Q_n^H(x) \geq V_n(x) - (T+2-d(n))\varepsilon.$$

Proof. We prove the result by induction. Consider any leaf node m in $\mathcal{N}$, with depth $d(m) = T+1$. By Lemma 1

$$Q_{T+1}^H(x_m) \geq V_{T+1}(x_m) - \varepsilon.$$

Consider any node n in $\mathcal{N}$ with depth $d(n)$ and state $x = x_\cdot^h$, and suppose the result is true for all nodes with depth greater than $d(n)$. Let m be a child of n and

$$u_m^* \in \arg\max_{u \in U(x_n)} \left\{r_m(x_n, u) + V_m\left(f_n(x_n, u, \xi_m)\right)\right\}.$$

Then for any $h = 1, 2, \ldots,$

$$\begin{aligned}
\max_{u \in U(x(n))} \left\{r_m(x_n, u) + Q_m^h\left(f_n(x_n, u, \xi_m)\right)\right\} &\geq r_m(x_n, u_m^*) + Q_m^h\left(f_n(x_n, u_m^*, \xi_m)\right) \\
&\geq r_m(x_n, u_m^*) + V_m\left(f_n(x_n, u_m^*, \xi_m)\right) \\
&\quad -(T+2-d(m))\varepsilon
\end{aligned}$$

Multiplying both sides by $\frac{p(m)}{p(n)}$ and adding over all children of n gives for any h,

$$q_n^h \geq V_n(x_n) - (T+1-d(n))\varepsilon,$$

where we have replaced $d(m)$ by $d(n)+1$. Now consider an arbitrary $x \in X$. Since $h \in \mathcal{H}_\delta(x)$ implies $x < x_n^h + \delta\mathbf{1}$,

$$V_n(x) \leq V_n(x_n^h) + \varepsilon,$$

giving

$$q_n^h \geq V_n(x) - (T + 1 - d(n))\varepsilon - \varepsilon.$$

By definition

$$Q_n^H(x) = \min\left\{q_n^h : h \in \mathcal{H}_\delta(x)\right\},$$

so

$$Q_n^H(x) \geq V_n(x) - (T + 2 - d(n)\varepsilon$$

as required. ■

Theorem 1. *If step 1a satisfies FPSP then sampled MIDAS converges almost surely to a $(T+1)\varepsilon$-optimal policy of MSP.*

Proof. We prove the result by induction. Consider the sequence $\{u_n^H, n \in \mathcal{N}\}_{H>H^*}$ of actions, and the corresponding sequence $\{x_n^H, n \in \mathcal{N}\}_{H>H^*}$ of states, defined by Lemma 5. With probability one, each x_n^H is some point that has been visited in a previous iteration of sampled MIDAS, and no new points are added. We show by induction that for every $n \in \mathcal{N}$,

$$Q_n^H(x_n^H) \leq V_n(x_n^H) + \varepsilon, \tag{8}$$

and for every $n \in \mathcal{N} \setminus \mathcal{L}$,

$$\sum_{m \in n+} \frac{p(m)}{p(n)} \left(r_m(x_n^H, u_m^H) + V_m^H(f_n(x_n^H, u_m^H, \xi_m))\right) \geq V_n(x_n^H) - (T + 2 - d(n))\varepsilon. \tag{9}$$

Consider any leaf node m in $\mathcal{L}$. By the FPSP this is visited an infinite number of times, so at least once by some iteration $H > H^*$ with probability 1. The collection $\{x_m^H : H = 1, 2, \ldots\} = \{x_m^h, h = 1, 2, \ldots, H^*\}$. For all such h, the terminal value function $V_m(x_m^h)$ is known exactly, so Lemma 1 yields

$$\begin{aligned} Q_m^H(x_m^H) &\leq Q_m^h(x_m^H) \\ &= Q_m^h(x_m^h) \\ &= V_m(x_m^h) \\ &\leq V_m(x_m^H) + \varepsilon \end{aligned}$$

where the last inequality holds because $\left\|x_m^H - x_m^h\right\|_\infty < \delta$, so (8) holds for m in $\mathcal{L}$. Now let

$$u_m^* \in \arg\max_{u \in U(x_n^H)} \left\{r_m(x_n^H, u) + V_m(f_n(x_n^H, u, \xi_m))\right\}$$

so

$$V_n(x_n^H) = \sum_{m \in n+} \frac{p(m)}{p(n)} \{r_m(x_n^H, u_m^*) + V_m(f_n(x_n^H, u_m^*, \xi_m))\}.$$

Then since

$$Q_m^H(f_n(x_n^H, u_m^*, \xi_m)) \geq V_m(f_n(x_n^H, u_m^*, \xi_m) - \varepsilon$$

by Lemma 3, we have

$$\begin{aligned} r_m(x_n^H, u_m^H) + Q_m^H(f_n(x_n^H, u_m^H, \xi_m)) &\geq r_m(x_n^H, u_m^*) + Q_m^H(f_n(x_n^H, u_m^*, \xi_m)) \\ &\geq r_m(x_n^H, u_m^*) + V_m(f_n(x_n^H, u_m^*, \xi_m) - \varepsilon. \end{aligned}$$

Also we have just shown

$$Q_m^H(x_m^H) \leq V_m(x_m^H) + \varepsilon$$

for all $m \in n+$, so

$$r_m(x_n^H, u_m^H) + V_m(f_n(x_n^H, u_m^H, \xi_m)) \geq r_m(x_n^H, u_m^*) + V_m(f_n(x_n^H, u_m^*, \xi_m) - 2\varepsilon.$$

Taking expectations of both sides with conditional probabilities $\frac{p(m)}{p(n)}$ yields (9) when $d(n) = T$.

Now suppose (8) holds for every node with depth $t + 1 \leq T + 1$, and (9) holds at every scenario tree node with depth $t \leq T$. Let n be a node at depth t. By (4)

$$\begin{aligned} Q_n^H(x_n^H) &\leq q_n^H \\ &= \sum_{m \in n+} \frac{p(m)}{p(n)} \max_{u \in U(x_n^h)} \left\{ r_m(x_n^H, u) + Q_m^H\left(f_t(x_t^h, u, \xi_m)\right)\right\} \end{aligned}$$

and (9) gives

$$\max_{u \in U(x_n^H)} \left\{ r_m(x_n^H, u) + Q_m^H\left(f_t(x_t^H, u, \xi_m)\right)\right\} \leq \max_{u \in U(x_n^H)} \left\{ r_m(x_n^H, u) + V_{t+1}\left(f_t(x_n^H, u, \xi_m)\right) + \varepsilon\right\}$$

so

$$\begin{aligned} Q_n^H(x_n^H) &\leq \sum_{m \in n+} \frac{p(m)}{p(n)} \max_{u \in U(x_n^h)} \left\{ r_m(x_n^H, u) + V_{t+1}\left(f_t(x_n^h, u, \xi_m)\right) + \varepsilon\right\} \\ &= V_n(x_n^H) + \varepsilon, \end{aligned}$$

thus showing (6) for every node $n \in \mathcal{N}$ with $d(n) = t - 1$. Furthermore suppose

$$\begin{aligned} \sum_{m \in n+} \frac{p(m)}{p(n)} \left(r_m(x_n^H, u_m^H) + V_m(f_n(x_n^H, u_m^H, \xi_m))\right) &\geq V_n(x_n^H) \\ &\quad -(T + 2 - d(n))\varepsilon. \end{aligned}$$

for any $n \in \mathcal{N}$ with $d(n) = t$. We have

$$\begin{aligned} r_n(x_{n-}^H, u_n^H) + Q_n^H(f_{n-}(x_{n-}^H, u_n^H, \xi_n)) &\geq r_n(x_{n-}^H, u_n^*) + Q_n^H(f_{n-}(x_{n-}^H, u_n^*, \xi_n)) \\ &\geq r_n(x_{n-}^H, u_n^*) + V_n(f_{n-}(x_{n-}^H, u_n^*, \xi_n)) \\ &\quad -(T + 2 - d(n))\varepsilon \end{aligned}$$

by Lemma 3. And we have just shown

$$Q_n^H(x_n^H) \leq V_n(x_n^H) + \varepsilon.$$

So

$$\begin{aligned} r_n(x^H_{n-}, u^H_n) + V_n(f_{n-}(x^H_{n-}, u^H_n, \xi_n)) &\geq r_n(x^H_{n-}, u^*_n) + V_n(f_{n-}(x^H_{n-}, u^*_n, \xi_n)) \\ &\quad -(T+2-d(n)+1)\varepsilon \\ &= r_n(x^H_{n-}, u^*_n) + V_n(f_{n-}(x^H_{n-}, u^*_n, \xi_n)) \\ &\quad -(T+2-d(n-))\varepsilon. \end{aligned}$$

Taking expectations of both sides with conditional probabilities $\frac{p(n)}{p(n-)}$ yields (9) for every $n \in \mathcal{N}$ with $d(n) = t-1$.

Since $x^H_0 = \bar{x}$, It follows by induction that

$$\sum_{m \in 0+} p(m) \left(r_m(\bar{x}, u^H_m) + V_m(f_0(\bar{x}, u^H_m, \xi_m)) \right) = V_0(\bar{x}) - (T+1)\varepsilon$$

showing that actions $\{u^H_m,\ m \in 0+\}$ are the initial decisions of $\{u^H_n, n \in \mathcal{N} \setminus \{0\}\}$ defining a $(T+1)\varepsilon$-optimal policy. ■

In practical implementations of MIDAS, we choose to stop after a fixed number $\hat{H}$ of iterations. Since $Q^{\hat{H}}_1$ gives a $(T+1)\varepsilon$-upperbound on any optimal policy, we can estimate an optimality gap by simulating the candidate policy and comparing the resulting estimate of its expected value (and its standard error) with $Q^{\hat{H}}_1 + (T+1)\varepsilon$.

We also remark that Algorithm 2 simplifies when ξ_t is stagewise independent. In this case the points (x^H_n, q^H_n) can be shared across all nodes having the same depth t. This means that there is a single approximation Q^H_t shared by all these nodes and updated once for all in each backward pass. The almost-sure convergence result for MIDAS applies in this special case, but one might expect the number of iterations needed to decrease dramatically in comparison with the general case.

5 Numerical example

To illustrate MIDAS we apply it to an instance of a single-reservoir hydroelectric scheduling problem, as described in section 1. In the model we consider, the state $x = (s_t, p_t)$ represents both a reservoir stock variable s_t and a price variable p_t with the following dynamics

$$\begin{bmatrix} s_{t+1} \\ p_{t+1} \end{bmatrix} = \begin{bmatrix} s_t - v_t - l_t + \omega_t \\ \alpha_t p_t + (1-\alpha_t) b_t + \eta_t \end{bmatrix},$$

where v_t is reservoir release, l_t is reservoir spill, ω_t is (random) reservoir inflow, and η_t is the error term for an autoregressive model of price that reverts to a mean price b_t. This means that $\xi_t = [\ \omega_t \quad \eta_t\ ]^\top$. We define the reward function as the revenue earned by the released energy $g(v)$ sold at price p,

$$r_t(s, p, v, \omega_t, \eta_t) = pg(v).$$

We approximate the value function by piecewise constant functions in two dimensions, where q^h represents the expected value function estimate at s^h and p^h. Following the formulation in the Appendix, the approximate value function $Q_t^H(s_t, p_t)$ for storage s_t and price p_t is defined as:

$$
\begin{aligned}
Q_t^H(s_t, p_t) = \text{max} \quad & \varphi \\
\text{s.t.} \quad & \\
& \varphi \leq q_t^h + (M - q_t^h)(1 - w_h), \\
& s_t \geq s_t^h z_s^h + \delta_s, \\
& p_t \geq p_t^h z_p^h + \delta_p, \\
& z_s^h + z_p^h = 1 - w_h, \\
& w_h,\ z_s^h,\ z_p^h \in \{0, 1\}.
\end{aligned} \tag{10}
$$

Figure 3 illustrates an example of how the value function is approximated. At $(s_t, p_t) = (175, 17.5)$, the expected value function estimate is 4000.

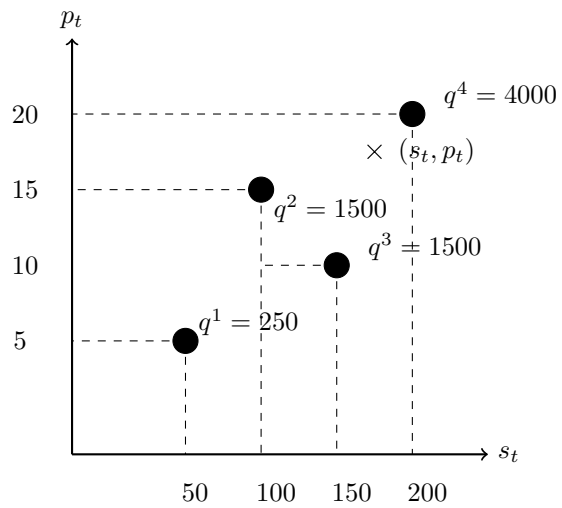

Figure 3: Value function approximation for $H = 4$. $Q_t^H(s_t, p_t)$, as shown by the cross, equals $q^4 = 4000$.

We applied Algorithm 2 to an instance with twelve stages (i.e. $T = 12$). The price is modelled as an autoregressive lag 1 process that reverts to a mean defined by b_t where the noise term η_t is assumed to have a standard normal distribution that is discretized into 10 outcomes. The number of possible price realizations in each stage increases with t, so a scenario tree representing this problem would have 10^{12} scenarios. We chose $\delta_s = 10$ and $\delta_p = 5$, and set the reservoir storage bounds to be between 10 and 200. The generation function $g(v)$ is a convex piecewise linear function of the turbine flow v, which is between 0 and 70. Table 1 summarizes the values of the parameters chosen.

Parameters	Value
T	12
α_t for $t = 1, 2, \ldots, T$	0.5
η_t for $t = 1, 2, \ldots, T$	$\sim$ Norm$(0, 1)$
b_t	$[61.261,\ 56.716,\ 59.159,\ 66.080,$ $72.131,\ 76.708,\ 76.665,\ 76.071,$ $76.832,\ 69.970,\ 69.132,\ 67.176]$
(δ_s, δ_p)	$(10, 5)$
X_δ	$\{x \in \mathbb{R} \ : \ x \in [10, 200]\}$
l for $t = 1, 2, \ldots, T$	0
ω_t for $t = 1, 2, \ldots, T$	0
$U(s_t)$	$\{v \in \mathbb{R} \ : \ v \in [0, \min\{s_t, 70\}]\}$
$g(v)$	$\begin{cases} 1.1v & \text{if } v \in [0, 50], \\ v + 5 & \text{if } v \in (50, 60], \\ 0.5(v - 60) + 65 & \text{if } v \in (60, 70], \\ 0 & \text{otherwise.} \end{cases}$
$\overline{s}$	100
$\overline{p}$	61.261
N	30

Table 1: Model parameters for single reservoir, twelve stage hydroscheduling problem.

We ran MIDAS for 200 iterations. Figure 4 illustrates all of the sampled price scenarios from the forward pass of the MIDAS algorithm. As the price model is mean reverting, it is generating price scenarios around the average price b_t indicated by the solid black line. As illustrated in Figure 5, the upper and lower bounds of the numerical example converge to the value of an optimal policy.

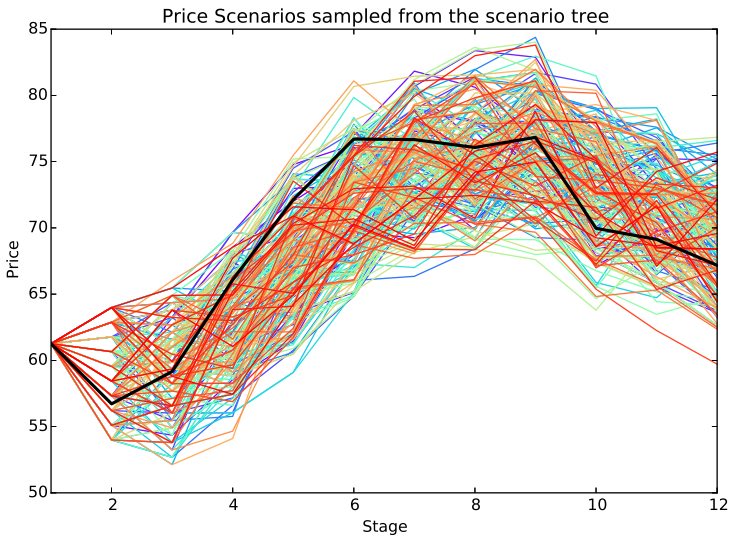

Figure 4: Visited price states of Algorithm 2.

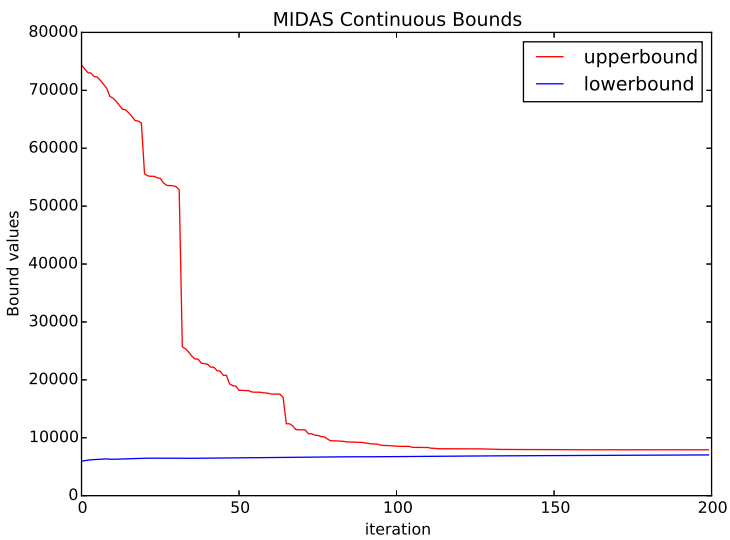

Figure 5: Upper and lower bound values by iteration.

Simulating the candidate policy at iteration 200 produces the prices shown in Figure 6 and the generation schedules illustrated in Figure 7. The simulation shows highest levels of generation are during periods of peak prices (periods 5 to 10). In scenarios where the observed price is higher than average the station generates more power. For example, in one price scenario 44 MWh of energy is generated in period 4 for a price of $\$70.07$, which is higher than the mean price of $\$59.159$.

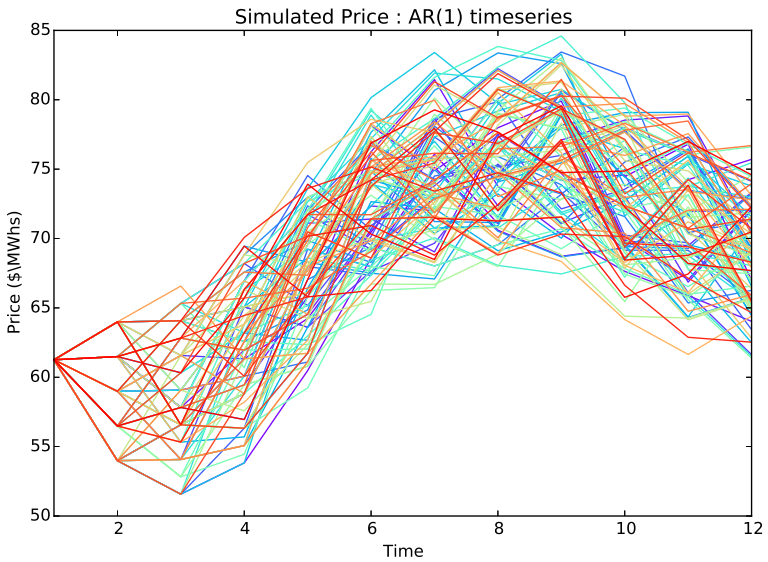

Figure 6: Prices scenarios generated from the policy simulation.

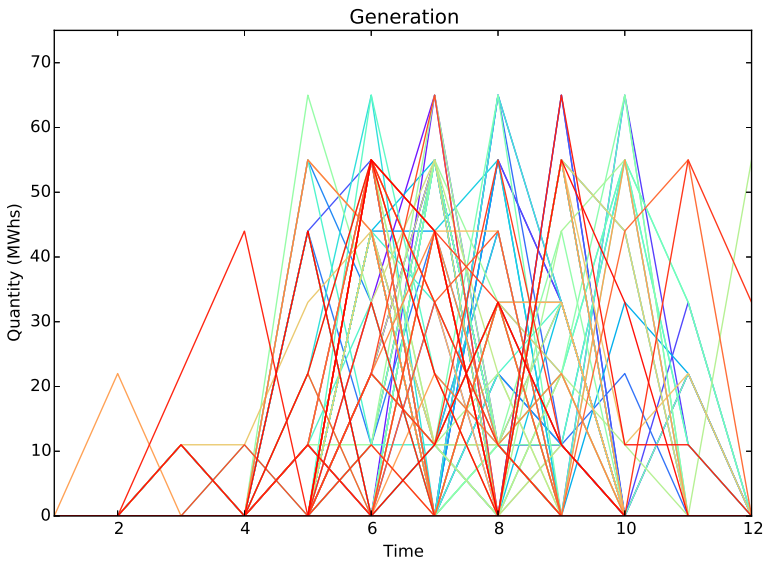

Figure 7: Corresponding offers of the price scenarios from the policy simulation.

6 Conclusion

In this paper, we have proposed a method called MIDAS for solving multistage stochastic dynamic programming problems with nondecreasing Bellman functions. We have demonstrated the almost-sure convergence of MIDAS to a $(T+1)\varepsilon$-optimal policy in a finite number of steps. Based on the numerical example in section 5, we may expect that the number of iterations will decrease as δ increases. Increasing δ increases the distance between distinct states that are visited, hence reducing the number of piecewise constant functions needed to cover the domain of the value function. However, the quality of the optimal policy will depend on the value of δ, since a smaller δ will give a lower ε. Furthermore, since the current formulation treats each stage problem as a MIP, the stage problems will increase in size with each iteration. Therefore it is important to choose an appropriate δ to produce good optimum policies within reasonable computation time.

MIDAS can be extended in several ways. Currently, we approximate the stage problems using a MIP formulation that treats the Bellman functions as piecewise constant. One formulation of such a MIP is given in the Appendix. A more accurate approximation might be achieved using an alternative formulation that for example uses specially ordered sets to yield a piecewise affine approximation. As long as this provides an ε-outer approximation of the Bellman function, we can incorporate it into a MIDAS scheme, which will converge almost surely by the same arguments above.

Although our convergence result applies to problems with continuous Bellman functions, MIDAS can be applied to multistage stochastic integer programming (MSIP) problems. Solving a deterministic equivalent of a MSIP can be difficult, as the scenario tree grows exponentially with the number of stages and the number of outcomes, and MIP algorithms generally scale poorly. However, since solving many small MIPs will be faster than solving a single large MIP, we might expect MIDAS to produce good candidate solutions to MSIP problems with much less computational effort. Our hope is that MIDAS will provide a practical computational approach to solving these difficult multistage stochastic integer programming problems.

Appendix: A MIP representation of $Q^H(x)$

Assume that $X = \{x : 0 \leq x_i \leq K_i,\ i = 1, 2, \ldots, n\}$, and let $Q(x)$ be any upper semi-continuous function defined on X. Suppose for some points x^h, $h = 1, 2, \ldots, H$, we have $Q(x^h) = q^h$. Recall $M = \max_{x \in X} Q(x)$,

$$\mathcal{H}_\delta(x) = \{h' : x_i^{h'} > x_i - \delta,\ i = 1, 2, \ldots, n\},$$
$$Q^H(x) = \min\left\{M, \min\left\{q^{h'} : h' \in \mathcal{H}_\delta(x)\right\}\right\}.$$

For $\delta > 0$, and for any $x \in X_\delta = \{x : \delta \leq x_i \leq K_i,\ i = 1, 2, \ldots, n\}$, define $\bar{Q}^H(x)$ to be the optimal value of the mixed integer program

$$
\begin{array}{llcll}
\text{MIP}(x)\text{: max} & \varphi & & & \\
\text{s.t.} & \varphi & \leq & q^h + (M - q^h)(1 - w_h), & h = 1, 2, \ldots, H, \\
& x_k & \geq & x_k^h z_k^h + \delta, & k = 1, 2, \ldots, n, \\
& \sum_{k=1}^n z_k^h & = & 1 - w_h, & h = 1, 2, \ldots, H, \\
& w_h & \in & \{0, 1\}, & h = 1, 2, \ldots, H, \\
& z_k^h & \in & \{0, 1\}, & k = 1, 2, \ldots, n, \\
& & & & h = 1, 2, \ldots, H.
\end{array}
$$

Proposition 2. *For every $x \in X_\delta$,*

$$\bar{Q}^H(x) = Q^H(x).$$

Proof. For a given point $x \in X_\delta$, consider w_h, z_k^h, $k = 1, 2, \ldots, n$, $h = 1, 2, \ldots, H$ that are feasible for MIP(x). If $w_h = 0$, $h = 1, 2, \ldots, H$, then $\varphi \leq M$ is the only constraint on φ and so $\bar{Q}^H(x) = M$. But $w_h = 0$, $h = 1, 2, \ldots, H$ means that for every such h, $z_k^h = 1$ for some component k giving

$$x_k \geq x_k^h + \delta.$$

Thus

$$H_\delta(x) = \{h : x_k < x_k^h + \delta \text{ for every } k\} = \emptyset.$$

Thus $Q^H(x) = M$ which is the same value as $\bar{Q}^H(x)$.

Now assume that the optimal solution to MIP(x) has $w_h = 1$ for some h. It suffices to show that

$$\bar{Q}^H(x) = \min\{q^h : h \in H_\delta(x)\}.$$

First if $h \in H_\delta(x)$ then $w_h = 1$. This is because choosing $w_h = 0$ implies $z_k^h = 1$ for some k, so for at least one k

$$x_k \geq x_k^h + \delta,$$

so $h \notin H_\delta(x)$.

Now if $h \notin H_\delta(x)$ then any feasible solution to MIP(x) can have either $w_h = 0$ or $w_h = 1$. Observe however that if

$$q^h < \min\{q^{h'} : h' \in H_\delta(x)\}$$

for any such h then choosing $w_h = 1$ for any of these would yield a value of φ strictly lower than the value obtained by choosing $w_h = 0$ for all of them. So $w_h = 0$ is optimal for $h \notin H_\delta(x)$. It follows that $H_\delta(x) = \{h : w_h = 1\}$. Thus the optimal value of MIP(x) is

$$\bar{Q}^H(x) = \min\{q^h : w_h = 1\} = \min\{q^h : h \in H_\delta(x)\} = Q^H(x).$$

■

References

[1] H. Abgottspon, K. Njalsson, M.A. Bucher, and G. Andersson. Risk-averse medium-term hydro optimization considering provision of spinning reserves. In *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pages 1–6. IEEE, 2014.

[2] S. Cerisola, J.M. Latorre, and A. Ramos. Stochastic dual dynamic programming applied to non-convex hydrothermal models. *European Journal of Operational Research*, 218(3):687–697, 2012.

[3] Z. Chen and W.B. Powell. Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *Journal of Optimization Theory and Applications*, 102(3): 497–524, 1999.

[4] C.J. Donohue and J.R. Birge. The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic Operations Research*, 1(1), 2006.

[5] P. Girardeau, V. Leclere, and A.B. Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40(1):130–145, 2014.

[6] M. Hindsberger and A.B. Philpott. Resa: A method for solving multistage stochastic linear programs. *Journal of Applied Operational Research*, 6(1):2–15, 2014.

[7] M.V.F. Pereira and L.M.V.G. Pinto. Multistage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1-3):359–375, 1991.

[8] A.B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

[9] A. Shapiro. On complexity of multistage stochastic programs. *Operations Research Letters*, 34(1): 1 – 8, 2006.

[10] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[11] G. Steeger and S. Rebennack. Strategic bidding for multiple price-maker hydroelectric producers. *IIE Transactions*, 47(9):1013–1031, 2015.

[12] F. Thome, M.V.F. Pereira, S. Granville, and M. Fampa. Non-convexities representation on hydrothermal operation planning using SDDP. Technical report, working paper, available: http://www. psr-inc. com. br/portal/psr/publicacoes, 2013.