Analysis of Markov Decision Processes under Parameter Uncertainty Online Companion

Peter Buchholz, Iryna Dohndorf, and Dimitri Scheftelowitsch

Department of Computer Science, TU Dortmund
{peter.buchholz,iryna.dohndorf,dimitri.scheftelowitsch}@cs.tu-dortmund.de

Abstract. Markov Decision Processes (MDPs) are a popular decision model for stochastic systems. Introducing uncertainty in the transition probability distribution by giving upper and lower bounds for the transition probabilities yields the model of Bounded Parameter MDPs (BMDPs) which captures many practical situations with limited knowledge about a system or its environment. In this paper the class of BMDPs is extended to Bounded Parameter Semi Markov Decision Processes (BSMDPs). The main focus of the paper is on the introduction and numerical comparison of different algorithms to compute optimal policies for BMDPs and BSMDPs; specifically, we introduce and compare variants of value and policy iteration.
The paper delivers an empirical comparison between different numerical algorithms for BMDPs and BSMDPs, with an emphasis on the required solution time.

Keywords: (Bounded Parameter) (Semi-)Markov Decision Process, Discounted Reward, Average Reward, Value Iteration, Policy Iteration

1 Algorithms

1.1 Solution methods for MDPs

Algorithm 1 Value iteration for discrete-time MDPs with discounted reward criterion

Require: MDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$, discount factor γ;
1: Specify $\boldsymbol{v}^{(0)} \geq \boldsymbol{0}$, $\epsilon > 0$ and set $k = 0$;
2: **repeat**
3: **for** $i \in \mathcal{S}$ **do**
4: $\boldsymbol{v}^{(k+1)}(i) = \max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a(i) + \gamma \sum_{j\in\mathcal{S}} \boldsymbol{P}^a(i,j)\boldsymbol{v}^{(k)}(j)\right)$;
5: $k = k + 1$;
6: **until** $\left\|\boldsymbol{v}^{(k-1)} - \boldsymbol{v}^{(k)}\right\| < \epsilon\frac{1-\gamma}{2\gamma}$
7: Choose $\boldsymbol{\pi}(i) \in \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a(i) + \gamma \sum_{j\in\mathcal{S}} \boldsymbol{P}^a(i,j)\boldsymbol{v}^{(k)}(j)\right)$ for all $i \in \mathcal{S}$;
8: **return** An ϵ-optimal policy $\boldsymbol{\pi}$, value vector $\boldsymbol{v}^{(k)}$;

Algorithm 2 Value iteration for discrete-time MDPs with expected total reward criterion

Require: MDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$;
1: Specify $\boldsymbol{v}^{(0)} \geq \boldsymbol{0}$, $\epsilon > 0$ and set $k = 0$;
2: **repeat**
3: **for** $i \in \mathcal{S}$ **do**
4: $\boldsymbol{v}^{(k+1)}(i) = \max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a(i) + \sum_{j\in\mathcal{S}} \boldsymbol{P}^a(i,j)\boldsymbol{v}^{(k)}(j)\right)$;
5: $k = k + 1$;
6: **until** $\left\|\boldsymbol{v}^{(k-1)} - \boldsymbol{v}^{(k)}\right\| < \epsilon$
7: $\boldsymbol{\pi}(i) \in \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a(i) + \sum_{j\in\mathcal{S}} \boldsymbol{P}^a(i,j)\boldsymbol{v}^{(k)}(j)\right)$ for all $i \in \mathcal{S}$;
8: **return** An ϵ-optimal policy $\boldsymbol{\pi}$;

Algorithm 3 Policy iteration for discrete-time MDPs with discounted reward criterion

Require: MDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$, discount factor γ
1: Specify $\boldsymbol{\pi}^{(0)} \in \mathit{\Pi}$ some pure initial policy and set $k = 0$;
2: **repeat**
3: (**Policy evaluation**) Solve

$$\boldsymbol{r}^{\boldsymbol{\pi}^{(k)}} = \left(\boldsymbol{I} - \gamma\boldsymbol{P}^{\boldsymbol{\pi}^{(k)}}\right)\boldsymbol{v}^{(k)};$$

4: (**Policy improvement**) Choose $\boldsymbol{\pi}^{(k+1)}$ to satisfy

$$\boldsymbol{\pi}^{(k+1)} = \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a + \gamma\boldsymbol{P}^a\boldsymbol{v}^{(k)}\right)$$

choosing $\boldsymbol{\pi}^{(k+1)}(i) = \boldsymbol{\pi}^{(k)}(i)$ when possible;
5: $k = k + 1$;
6: **until** $\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi}^{(k-1)}$
7: **return** An optimal policy $\boldsymbol{\pi}^* = \boldsymbol{\pi}^{(k)}$;

Algorithm 4 Policy iteration for discrete-time MDPs with expected total reward criterion

Require: MDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$;
1: Specify $\boldsymbol{\pi}^{(0)} \in \Pi$ some pure initial policy and set $k = 0$;
2: **repeat**
3: (**Policy evaluation**) Solve

$$\boldsymbol{r}^{\boldsymbol{\pi}^{(k)}} = \left(\boldsymbol{I} - \boldsymbol{P}^{\boldsymbol{\pi}^{(k)}}\right)\boldsymbol{v}^{(k)};$$

4: (**Policy improvement**) Choose $\boldsymbol{\pi}^{(k+1)}$ to satisfy

$$\boldsymbol{\pi}^{(k+1)} = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}^a + \boldsymbol{P}^a\boldsymbol{v}^{(k)}\right)$$

choosing $\boldsymbol{\pi}^{(k+1)}(i) = \boldsymbol{\pi}^{(k)}(i)$ when possible;
5: $k = k + 1$;
6: **until** $\boldsymbol{\pi}^{(k-1)} = \boldsymbol{\pi}^{(k)}$
7: **return** An optimal policy $\boldsymbol{\pi}^* = \boldsymbol{\pi}^{(k)}$;

1.2 Solution methods for SMDPs

Algorithm 5 Relative value iteration for discrete-time MDPs with average reward criterion

Require: MDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$
1: Specify $\boldsymbol{v}^{(0)} \geq \boldsymbol{0}$, $\epsilon > 0$, set $k = 0$, and choose one state $i_0 \in \mathcal{S}$;
2: **repeat**
3: $\quad \boldsymbol{w}^{(k)} = \boldsymbol{v}^{(k)} - \boldsymbol{e}\boldsymbol{v}^{(k)}(i_0)$;
4: $\quad$ **for** $i \in \mathcal{S}$ **do**
5: $\quad\quad \boldsymbol{v}^{(k+1)}(i) = \max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a(i) + \sum_{j\in\mathcal{S}} \boldsymbol{P}^a(i,j)\boldsymbol{w}^{(k)}(j)\right)$;
6: $\quad k = k + 1$;
7: **until** $\max_{i\in\mathcal{S}} \left(\boldsymbol{v}^{(k+1)}(i) - \boldsymbol{v}^{(k)}(i)\right) - \min_{i\in\mathcal{S}} \left(\boldsymbol{v}^{(k+1)}(i) - \boldsymbol{v}^{(k)}(i)\right) < \epsilon$
8: $\boldsymbol{\pi}(i) \in \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a(i) + \sum_{j\in\mathcal{S}} \boldsymbol{P}^a(i,j)\boldsymbol{w}^{(k)}(j)\right)$ for all $i \in \mathcal{S}$;
9: Set $\bar{\boldsymbol{\pi}} = \boldsymbol{\pi}$, $G = \boldsymbol{w}^{(k)}(1)$ and $\boldsymbol{h} = \boldsymbol{w}$;
10: **return** An ϵ-optimal policy $\bar{\boldsymbol{\pi}}$, average gain G and bias vector $\boldsymbol{h}$;

Algorithm 6 Policy iteration for discrete-time MDPs with average reward criterion

Require: MDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$
1: Specify $\boldsymbol{\pi}^{(0)} \in \Pi$ some pure initial policy and set $k = 0$;
2: **repeat**
3: $\quad$ (**Policy evaluation**) Solve

$$\boldsymbol{r}^{\boldsymbol{\pi}^{(k)}} = \left(\boldsymbol{I} - \boldsymbol{P}^{\boldsymbol{\pi}^{(k)}}\right)\bar{\boldsymbol{g}}^{(k)} + G\mathbb{1}$$

by setting $\bar{\boldsymbol{g}}(i_0) = 0$ for some fixed state $i_0 \in \mathcal{S}$. Compute $\boldsymbol{H}^{(k)} = \left(\boldsymbol{I} - \boldsymbol{P}^{\boldsymbol{\pi}^{(k)}}\right)$. Then $\bar{\boldsymbol{H}}^{(k)}$ is the matrix with the column corresponding to state i_0 replaced by a column of 1's. Solve the linear system

$$\boldsymbol{r}^{\boldsymbol{\pi}^{(k)}} = \bar{\boldsymbol{H}}^{(k)}\boldsymbol{w}$$

where $G^{(k)}$ is the i_0th component of the solution vector $\boldsymbol{w}$ and $\boldsymbol{h}^{(k)}(i) = \boldsymbol{w}(i)$ for $i \neq i_0$;
4: $\quad$ (**Policy improvement**) Choose $\boldsymbol{\pi}^{(k+1)}$ to satisfy

$$\boldsymbol{\pi}^{(k+1)} = \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a + \boldsymbol{P}^a\boldsymbol{h}^{(k)}\right)$$

choosing $\boldsymbol{\pi}^{(k+1)}(i) = \boldsymbol{\pi}^{(k)}(i)$ when possible;
5: $\quad k = k + 1$;
6: **until** $\boldsymbol{\pi}^{(k-1)} = \boldsymbol{\pi}^{(k)}$
7: Set $\bar{\boldsymbol{\pi}}^* = \boldsymbol{\pi}^{(k)}$, $G^* = G^{(k-1)}$, $\boldsymbol{h} = \boldsymbol{h}^{(k-1)}$
8: **return** An optimal policy $\bar{\boldsymbol{\pi}}^*$, the optimal average gain G^* and the deviation vector $\boldsymbol{h}$;

Algorithm 7 Uniformization method for SMDPs with average reward criterion

Require: SMDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$, time vectors $\{\boldsymbol{y}^a\}_{a\in A}$ (average sojourn times in states)

```
Choose η = min_{i∈S} min_{a∈A} y^a(i)/(1 − P^a(i,i));
for a ∈ A do
    s̄^a(i) = r^a(i)/y^a(i);
for a ∈ A do
    for i ∈ S do
        for j ∈ S do
            if i ≠ j then
                Q̄^a(i,j) = η P^a(i,j)/y^a(i):
            else
                Q̄^a(i,j) = 1 + η (P^a(i,j) − 1)/y^a(i);
return Discrete-time MDP (S, A, (Q̄^a)_{a∈A}, {s^a}_{a∈A}), η;
```

Choose $\eta = \min_{i\in\mathcal{S}} \min_{a\in\mathcal{A}} \boldsymbol{y}^a(i)/(1-\boldsymbol{P}^a(i,i))$;
for $a \in \mathcal{A}$ **do**
  $\bar{\boldsymbol{s}}^a(i) = \boldsymbol{r}^a(i)/\boldsymbol{y}^a(i)$;
for $a \in \mathcal{A}$ **do**
  for $i \in \mathcal{S}$ **do**
    for $j \in \mathcal{S}$ **do**
      if $i \neq j$ **then**
        $\bar{\boldsymbol{Q}}^a(i,j) = \eta \frac{\boldsymbol{P}^a(i,j)}{\boldsymbol{y}^a(i)}$:
      else
        $\bar{\boldsymbol{Q}}^a(i,j) = 1 + \eta \frac{\boldsymbol{P}^a(i,j)-1}{\boldsymbol{y}^a(i)}$;
return Discrete-time MDP $(\mathcal{S}, \mathcal{A}, (\bar{\boldsymbol{Q}}^a)_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}})$, η;

Algorithm 8 Value iteration for discrete-time SMDPs with average reward criterion

Require: SMDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$, time vectors $\{\boldsymbol{y}^a\}_{a\in A}$ (average sojourn times in states)
Apply Algorithm 7 to transform the SMDP in an according to the average reward equivalent MDP $\{\mathcal{S}, \mathcal{A}, \{\bar{\boldsymbol{Q}}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}}\}$. Save η;
Use value iteration Algorithm 5 to analyze the MDP;
return An ϵ-optimal policy $\bar{\boldsymbol{\pi}}$, average gain G and bias vector $\boldsymbol{h} = \eta\boldsymbol{h}$;

Algorithm 9 Policy iteration for discrete-time SMDPs with average reward criterion

Require: SMDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \boldsymbol{p})$, time vectors $\{\boldsymbol{y}^a\}_{a\in A}$ (average sojourn times in states)
Apply Algorithm 7 to transform the SMDP in an according to the average reward equivalent MDP $\{\mathcal{S}, \mathcal{A}, \{\bar{\boldsymbol{Q}}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}}\}$. Save η;
Use policy iteration Algorithm 6 to analyze the MDP;
return An optimal policy $\bar{\boldsymbol{\pi}}^*$, average gain G^* and bias vector $\boldsymbol{h} = \eta\boldsymbol{h}$;

Algorithm 10 Transformation method for SMDPs with discounted reward criterion

Require: SMDP $(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a)_{a\in\mathcal{A}}, (\boldsymbol{r}^a)_{a\in A}, \Big((\boldsymbol{p}^{(a,i)}, \boldsymbol{D}_0^{(a,i)})\Big)_{a\in A, i\in\mathcal{S}})$, discount factor β;

1: **for** $a \in \mathcal{A}$ **do**

2: Let $\{(\boldsymbol{p}^{(i)},\ \boldsymbol{D}_0^{(i)})\}_{i\in\mathcal{S}}$ be the set of Phase-type distributions corresponding to the action a;

3: **for** $i \in \mathcal{S}$ **do**

4: Compute $\boldsymbol{s}^a(i) = \boldsymbol{r}^a(i) \int\limits_0^\infty (1 - F^a(i,t))e^{-\beta t}dt$ and for all $j \in \mathcal{S}$ $\bar{\boldsymbol{Q}}^a(i,j) = \boldsymbol{P}^a(i,j) \int\limits_0^\infty f^a(i,t)e^{-\beta t}dt$ with the uniformization based method [1] using the following data.
Compute $\boldsymbol{d}_1^{(i)} = -\boldsymbol{D}_0^{(i)}\mathbb{1}$;
Set $\boldsymbol{P}^{(i)} = \boldsymbol{D}_0^{(i)} - \beta\boldsymbol{I}$ and $\lambda = \max_{\forall i,j\in\mathcal{S}} |\boldsymbol{P}^{(i)}(i,j)|$;
Compute $\boldsymbol{P}^{(i)} = \frac{1}{\lambda}\boldsymbol{P}^{(i)} + \boldsymbol{I}$, $\boldsymbol{d}_1^{(i)} = \frac{1}{\lambda}\boldsymbol{d}_1^{(i)}$ and the time step $\Delta = 1/\lambda$;

5: **return** Discrete-time discounted MDP $(\mathcal{S}, \mathcal{A}, (\bar{\boldsymbol{Q}}^a)_{a\in\mathcal{A}}, (\boldsymbol{s}^a)_{a\in\mathcal{A}})$;

Algorithm 11 Value iteration for discrete-time SMDPs with discounted reward criterion

Require: SMDP $(\mathcal{S}, \mathcal{A}, \{\boldsymbol{P}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{r}^a\}_{a\in A}, \{(\boldsymbol{p}^{(a,i)},\ \boldsymbol{D}_0^{(a,i)})\}_{a\in A, i\in\mathcal{S}})$, discount factor β;

1: Apply transformation Algorithm 10 to transform the SMDP in an according to the discounted reward equivalent MDP $(\mathcal{S}, \mathcal{A}, \{\bar{\boldsymbol{Q}}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}})$;

2: Use value iteration Algorithm 2 to analyze the MDP $\{\mathcal{S}, \mathcal{A}, \{\bar{\boldsymbol{Q}}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}}\}$ according to the expected total reward criterion;

3: **return** An ϵ-optimal policy $\boldsymbol{\pi}$;

Algorithm 12 Policy iteration for discrete-time SMDPs with discounted reward criterion

Require: SMDP $(\mathcal{S}, \mathcal{A}, \{\boldsymbol{P}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{r}^a\}_{a\in A}, \{(\boldsymbol{p}^{(a,i)},\ \boldsymbol{D}_0^{(a,i)})\}_{a\in A, i\in\mathcal{S}})$, discount factor β;

1: Apply transformation Algorithm 10 to transform the SMDP in an according to the discounted reward equivalent MDP $(\mathcal{S}, \mathcal{A}, \{\bar{\boldsymbol{Q}}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}})$;

2: Use policy iteration Algorithm 4 to analyze the MDP $\{\mathcal{S}, \mathcal{A}, \{\bar{\boldsymbol{Q}}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{s}^a\}_{a\in\mathcal{A}}\}$ according to the expected total reward criterion;

3: **return** Optimal policy $\boldsymbol{\pi}^*$;

1.3 Solution methods for BMDPs

Algorithm 13 Interval value iteration for discrete-time BMDPs with discounted reward criterion

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$, discount factor $(\boldsymbol{\gamma}^a_{\updownarrow})_{a\in\mathcal{A}}$, *pessimistic* is *true* when the optimal lower bound has to be computed and *false* when the optimal upper bound has to be computed;

Specify $\boldsymbol{v}^{(0)} \geq \mathbf{0}$, $\boldsymbol{\pi}^{(0)} \geq \mathbf{0}$, $\epsilon > 0$ and set $k = 0$;
$\rightarrow = \downarrow$ if *pessimistic*, otherwise $\rightarrow = \uparrow$
$\gamma^* = \max\left\{\boldsymbol{\gamma}^a_{\rightarrow}(i) \mid (i, a) \in \mathcal{S} \times \mathcal{A}\right\}$
repeat
  for $i \in \mathcal{S}$ **do**
    $[\boldsymbol{v}^{(k+1)}(i),\ \boldsymbol{\pi}^{(k+1)}(i)] = \text{interval\_value}(i,\ \boldsymbol{\pi}^{(k)}(i),\ (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}},\ (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}},\ (\boldsymbol{\gamma}^a_{\rightarrow})_{a\in\mathcal{A}},\ \boldsymbol{v}^{(k)},\ \epsilon,\ \textit{pessimistic})$;
  $k = k + 1$;
until $\left\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\right\| < \epsilon \frac{1-\gamma^*}{2\gamma^*}$
return An ϵ-optimal policy $\boldsymbol{\pi}^{(k)}$, value vector $\boldsymbol{v}^{(k)}$;

function INTERVAL_VALUE(state i, current decision ai, $(\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}$, $(\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}$, $(\boldsymbol{\gamma}^a)_{a\in\mathcal{A}}$, $\boldsymbol{v}$, ϵ, *pessimistic*)
    $w = -1.0e + 12$;
    if *pessimistic* **then**
        $(i_1, i_2, \ldots, i_n) \leftarrow$ *ascending* order of states with respect to states' values $\boldsymbol{v}$;
    else
        $(i_1, i_2, \ldots, i_n) \leftarrow$ *descending* order of states with respect to states' values $\boldsymbol{v}$;
    $\boldsymbol{p} = \boldsymbol{P}^a_{\downarrow}$;
    for $a \in \mathcal{A}$ **do**
        $val = \gamma\, \boldsymbol{p}\, \boldsymbol{v}$;
        if *pessimistic* **then**
            $val = val + \boldsymbol{r}^a_{\downarrow}(i)$;
        else
            $val = val + \boldsymbol{r}^a_{\uparrow}(i)$;
        $r = \boldsymbol{p}\mathbb{1}$;
        for $j \in (i_1, i_2, \ldots, i_n)$ **do**
            if $\boldsymbol{P}^a_{\uparrow}(i,j) > \boldsymbol{P}^a_{\downarrow}(i,j)$; **then**
                $m = \min(\boldsymbol{P}^a_{\uparrow}(i,j) - \boldsymbol{p}(j), 1 - r)$;
                $val = val + \boldsymbol{\gamma}^a(i) \cdot m\boldsymbol{v}(j)$;
                $r = r + m$;
            if $r \geq 1 - 10\epsilon$; **then return** ;
        if $val > w$ **then**
            $w = val$;
            $ai = a$;
    return ai, w;

Algorithm 14 Policy iteration 1 for discrete-time BMDPs with discounted reward criterion

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$, discount factor $(\boldsymbol{\gamma}^a_{\updownarrow})_{a\in\mathcal{A}}$, *pessimistic* is *true* when the optimal lower bound has to be computed and *false* when the optimal upper bound has to be computed;

1: Specify $\boldsymbol{\phi}^{(1)} \in \Pi$ some pure initial policy, $\boldsymbol{v}^{(0)} = \boldsymbol{r}^{\phi(1)}$ and set $k = 1$;
2: **if** *pessimistic* **then**
3: $\quad \boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}^{\phi^{(k)}}_{\downarrow})$;
$\quad \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(k-1)}) = \arg\min_{\boldsymbol{P}\in\boldsymbol{P}^{\phi^{(k)}}_{\updownarrow}} (\boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{v}^{(k-1)})$;
$\quad$ Solve

$$\boldsymbol{r}^{\boldsymbol{\phi}^{(k)}}_{\downarrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(k-1)})\right)\boldsymbol{v}^{(k)};$$

4: $\quad$ **for** $i \in \mathcal{S}$ **do**
5: $\quad\quad$ **for** $a \in \mathcal{A}$ **do**
6: $\quad\quad\quad f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v}^{(k)}) = \min_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}}\left(\boldsymbol{\Gamma}\boldsymbol{P}(i\bullet)\boldsymbol{v}^{(k)})\right)$;
7: $\quad\quad$ Choose $\boldsymbol{\phi}^{(k+1)}(i)$ to satisfy

$$\boldsymbol{\phi}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}^a_{\downarrow}(i) + \boldsymbol{\gamma}^a_{\downarrow}(i) f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v}^{(k)})\right)$$

$\quad\quad$ keeping $\boldsymbol{\phi}^{(k+1)}(i) = \boldsymbol{\phi}^{(k)}(i)$ when possible;
8: $\quad$ **if** $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)}$ **then**
9: $\quad\quad$ Set $\boldsymbol{\phi}^*_{\downarrow} = \boldsymbol{\phi}^{(k+1)}$ and terminate. Otherwise set $k = k+1$ and go to Step 3;
10: **else**
11: $\quad \boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}^{\phi^{(k)}}_{\uparrow})$;
$\quad \boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(k-1)}) = \arg\max_{\boldsymbol{P}\in\boldsymbol{P}^{\phi^{(k)}}_{\updownarrow}} (\boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{v}^{(k-1)})$;
$\quad$ Solve

$$\boldsymbol{r}^{\boldsymbol{\phi}^{(k)}}_{\uparrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(k-1)})\right)\boldsymbol{v}^{(k)};$$

12: $\quad$ **for** $i \in \mathcal{S}$ **do**
13: $\quad\quad$ **for** $a \in \mathcal{A}$ **do**
14: $\quad\quad\quad f^a_{\uparrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v}^{(k)}) = \max_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}}\left(\boldsymbol{P}(i\bullet)\boldsymbol{v}^{(k)})\right)$;
15: $\quad\quad$ Choose $\boldsymbol{\phi}^{(k+1)}(i)$ to satisfy

$$\boldsymbol{\phi}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}^a_{\uparrow}(i) + \boldsymbol{\gamma}^a_{\uparrow}(i) f^a_{\uparrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v}^{(k)})\right)$$

$\quad\quad$ keeping $\boldsymbol{\phi}^{(k+1)}(i) = \boldsymbol{\phi}^{(k)}(i)$ when possible;
16: $\quad$ **if** $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)}$ **then**
17: $\quad\quad$ Set $\boldsymbol{\phi}^*_{\uparrow} = \boldsymbol{\phi}^{(k+1)}$ and terminate. Otherwise set $k = k+1$ and go to Step 11;
18: **return** An optimal policy $\boldsymbol{\phi}^*_{\downarrow}$ if *pessimistic* is *true* and $\boldsymbol{\phi}^*_{\uparrow}$ if *pessimistic* is *false*;

Algorithm 15 Policy iteration 2 for discrete-time BMDPs with discounted reward criterion

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$, discount factor $(\boldsymbol{\gamma}^a_{\updownarrow})_{a\in\mathcal{A}}$, *pessimistic* is *true* when the optimal lower bound has to be computed and *false* when the optimal upper bound has to be computed;

1: Specify $\boldsymbol{\phi}^{(1)} \in \Pi$ some pure initial policy, $\boldsymbol{v}^{(0)} = \boldsymbol{r}^{\phi(1)}$, $\epsilon > 0$ and set $k = 1$, $l = 1$;
2: **if** *pessimistic* **then**
3: **repeat**
4: $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}^{\phi^{(k)}}_{\downarrow})$;
$\boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(l-1)}) = \arg\min_{\boldsymbol{P}\in\boldsymbol{P}^{\phi^{(k)}}_{\updownarrow}}(\boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{v}^{(l-1)})$;
Solve

$$\boldsymbol{r}^{\boldsymbol{\phi}^{(k)}}_{\downarrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(l-1)})\right)\boldsymbol{v}^{(l)};$$

5: **until** $\left\|\boldsymbol{v}^{(l)} - \boldsymbol{v}^{(l-1)}\right\| < \epsilon$;
6: $\boldsymbol{v} = \boldsymbol{v}^{(l)}$;
7: **for** $i \in \mathcal{S}$ **do**
8: **for** $a \in \mathcal{A}$ **do**
9: $f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v}) = \min_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}}\left(\boldsymbol{P}(i\bullet)\boldsymbol{v}\right)$;
10: Choose $\boldsymbol{\phi}^{(k+1)}(i)$ to satisfy

$$\boldsymbol{\phi}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}^a_{\downarrow}(i) + \boldsymbol{\gamma}^a_{\downarrow}(i) f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v})\right)$$

keeping $\boldsymbol{\phi}^{(k+1)}(i) = \boldsymbol{\phi}^{(k)}(i)$ when possible;
11: **if** $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)}$ **then**
12: Set $\boldsymbol{\phi}^*_{\downarrow} = \boldsymbol{\phi}^{(k+1)}$ and terminate. Otherwise set $k = k+1$, $l = 1$, $\boldsymbol{v}^{(0)} = \boldsymbol{v}$ and go to Step 3;
13: **else**
14: **repeat**
15: $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}^{\phi^{(k)}}_{\downarrow})$;
$\boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(l-1)}) = \arg\max_{\boldsymbol{P}\in\boldsymbol{P}^{\phi^{(k)}}_{\updownarrow}}(\boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{v}^{(l-1)})$;
Solve

$$\boldsymbol{r}^{\boldsymbol{\phi}^{(k)}}_{\uparrow} = \left(\boldsymbol{I} - \gamma\boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\boldsymbol{\phi}^{(k)}}_{\updownarrow}, \boldsymbol{v}^{(l-1)})\right)\boldsymbol{v}^{(l)};$$

16: **until** $\left\|\boldsymbol{v}^{(l)} - \boldsymbol{v}^{(l-1)}\right\| < \epsilon$;
17: $\boldsymbol{v} = \boldsymbol{v}^{(l)}$;
18: **for** $i \in \mathcal{S}$ **do**
19: **for** $a \in \mathcal{A}$ **do**
20: $f^a_{\uparrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v}) = \max_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}}\left(\boldsymbol{P}(i\bullet)\boldsymbol{v}\right)$;
21: Choose $\boldsymbol{\phi}^{(k+1)}(i)$ to satisfy

$$\boldsymbol{\phi}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}^a_{\uparrow}(i) + \boldsymbol{\gamma}^a_{\uparrow}(i) f^a_{\uparrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \boldsymbol{v})\right)$$

keeping $\boldsymbol{\phi}^{(k+1)}(i) = \boldsymbol{\phi}^{(k)}(i)$ when possible;
22: **if** $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)}$ **then**
23: Set $\boldsymbol{\phi}^*_{\downarrow} = \boldsymbol{\phi}^{(k+1)}$ and terminate. Otherwise set $k = k+1$, $l = 1$, $\boldsymbol{v}^{(0)} = \boldsymbol{v}$ and go to Step 14;
24: **return** An optimal policy $\boldsymbol{\phi}^*_{\downarrow}$ if *pessimistic* is *true* and $\boldsymbol{\phi}^*_{\uparrow}$ if *pessimistic* is *false*;

Algorithm 16 Interval value iteration for discrete-time BMDPs with average reward criterion

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$, *pessimistic* is *true* when the optimal lower bound has to be computed and *false* when the optimal upper bound has to be computed;

1: Specify $\boldsymbol{v}^{(0)} \geq \boldsymbol{0}$, $\boldsymbol{\pi}^{(0)} \geq \boldsymbol{0}$, $\epsilon > 0$, set $k = 0$, and choose one state $i_0 \in \mathcal{S}$;
2: **repeat**
3: $\quad \boldsymbol{w}^{(k)} = \boldsymbol{v}^{(k)} - \boldsymbol{e}\boldsymbol{v}^{(k)}(i_0)$;
4: $\quad$ **for** $i \in \mathcal{S}$ **do**
5: $\quad\quad [\boldsymbol{v}^{(k+1)}(i),\ \boldsymbol{\pi}^{(k+1)}(i)] = \text{interval\_value}(i, \boldsymbol{\pi}^{(k)}(i), (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}, \mathbb{1}, \boldsymbol{w}^{(k)},$ ϵ, *pessimistic*);
6: **until** $\max_{i\in\mathcal{S}}\left(\boldsymbol{v}^{(k+1)}(i) - \boldsymbol{v}^{(k)}(i)\right) - \min_{i\in\mathcal{S}}\left(\boldsymbol{v}^{(k+1)}(i) - \boldsymbol{v}^{(k)}(i)\right) < \epsilon$
7: Set $\bar{\boldsymbol{\pi}} = \boldsymbol{\pi}^{(k)}$, $G = \boldsymbol{w}^{(k)}(1)$ and $\boldsymbol{h} = \boldsymbol{w}$;
8: **return** An ϵ-optimal policy $\bar{\boldsymbol{\pi}}$, average gain G and bias vector $\boldsymbol{h}$;

Algorithm 17 Policy iteration 1 for discrete-time BMDPs with average reward criterion when optimal *lower* bound should be computed

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$;

1: Specify $\bar{\boldsymbol{\phi}}^{(1)} \in \Pi$ some pure initial policy, $\bar{\boldsymbol{h}}^{(0)} = \boldsymbol{r}^{\bar{\phi}(1)}_{\downarrow}$ and set $k = 1$;
2: $\boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(k-1)}) = \arg\min_{\boldsymbol{P}\in\boldsymbol{P}^{\bar{\phi}^{(k)}}_{\updownarrow}} (\boldsymbol{P}\bar{\boldsymbol{h}}^{(k-1)})$;
Solve

$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\downarrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(k-1)})\right)\bar{\boldsymbol{h}}^{(k)} + \bar{H}_{\downarrow}\mathbb{1};$$

by setting $\bar{\boldsymbol{h}}^{(k)}(i_0) = 0$ for some fixed state $i_0 \in \mathcal{S}$. Compute $\bar{\boldsymbol{G}}^{(k)}_{\downarrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(k-1)})\right)$. Then $\tilde{\boldsymbol{G}}^{(k)}_{\downarrow}$ is the matrix with the column corresponding to state i_0 replaced by a column of 1's. Solve the linear system

$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}} = \tilde{\boldsymbol{G}}^{(k)}_{\downarrow}\boldsymbol{w}$$

where $\bar{H}^{(k)}_{\downarrow}$ is the i_0th component of the solution vector $\boldsymbol{w}$ and $\bar{\boldsymbol{h}}^{(k)}(i) = \boldsymbol{w}(i)$ for $i \neq i_0$;
3: **for** $i \in \mathcal{S}$; **do**
4: $\quad$ **for** $a \in \mathcal{A}$; **do**
5: $\quad\quad f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \bar{\boldsymbol{h}}^{(k)}) = \min_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}}\left(\boldsymbol{P}(i\bullet)\bar{\boldsymbol{h}}^{(k)})\right)$;
6: $\quad$ Choose $\bar{\boldsymbol{\phi}}^{(k+1)}(i)$ to satisfy

$$\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}^a_{\downarrow}(i) + f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \bar{\boldsymbol{h}}^{(k)})\right)$$

$\quad$ keeping $\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \bar{\boldsymbol{\phi}}^{(k)}(i)$ when possible;
7: **if** $\bar{\boldsymbol{\phi}}^{(k+1)} = \bar{\boldsymbol{\phi}}^{(k)}$ **then**
8: $\quad$ Set $\bar{\boldsymbol{\phi}}^*_{\downarrow} = \bar{\boldsymbol{\phi}}^{(k+1)}$ and terminate. Otherwise set $k = k+1$ and go to Step 2;
9: **return** An optimal policy $\bar{\boldsymbol{\phi}}^*_{\downarrow}$;

Algorithm 18 Policy iteration 1 for discrete-time BMDPs with average reward criterion when optimal *upper* bound should be computed

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$;

1: Specify $\bar{\boldsymbol{\phi}}^{(1)} \in \Pi$ some pure initial policy, $\bar{\boldsymbol{h}}^{(0)} = \boldsymbol{r}^{\bar{\phi}(1)}_{\uparrow}$ and set $k = 1$;

2: $\boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(k-1)}) = \arg\max_{\boldsymbol{P}\in\boldsymbol{P}^{\bar{\phi}^{(k)}}_{\updownarrow}} (\boldsymbol{P}\bar{\boldsymbol{h}}^{(k-1)})$;

Solve

$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\uparrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(k-1)})\right)\bar{\boldsymbol{h}}^{(k)} + \bar{H}_{\uparrow}\mathbb{1};$$

by setting $\bar{\boldsymbol{h}}^{(k)}(i_0) = 0$ for some fixed state $i_0 \in \mathcal{S}$. Compute $\bar{\boldsymbol{G}}^{(k)}_{\uparrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\uparrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(k-1)})\right)$. Then $\tilde{\boldsymbol{G}}^{(k)}_{\uparrow}$ is the matrix with the column corresponding to state i_0 replaced by a column of 1's. Solve the linear system

$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}} = \tilde{\boldsymbol{G}}^{(k)}_{\uparrow}\boldsymbol{w}$$

where $\bar{H}^{(k)}_{\uparrow}$ is the i_0th component of the solution vector $\boldsymbol{w}$ and $\bar{\boldsymbol{h}}^{(k)}(i) = \boldsymbol{w}(i)$ for $i \neq i_0$;

3: **for** $i \in \mathcal{S}$; **do**

4: **for** $a \in \mathcal{A}$; **do**

5: $f^a_{\uparrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \bar{\boldsymbol{h}}^{(k)}) = \max_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}} \left(\boldsymbol{P}(i\bullet)\bar{\boldsymbol{h}}^{(k)})\right)$;

6: Choose $\bar{\boldsymbol{\phi}}^{(k+1)}(i)$ to satisfy

$$\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a_{\uparrow}(i) + f^a_{\uparrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \bar{\boldsymbol{h}}^{(k)})\right)$$

keeping $\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \bar{\boldsymbol{\phi}}^{(k)}(i)$ when possible;

7: **if** $\bar{\boldsymbol{\phi}}^{(k+1)} = \bar{\boldsymbol{\phi}}^{(k)}$ **then**

8: Set $\bar{\boldsymbol{\phi}}^*_{\uparrow} = \bar{\boldsymbol{\phi}}^{(k+1)}$ and terminate. Otherwise set $k = k + 1$ and go to Step 2;

9: **return** An optimal policy $\bar{\boldsymbol{\phi}}^*_{\uparrow}$;

Algorithm 19 Policy iteration 2 for discrete-time BMDPs with average reward criterion when optimal *lower* bound should be computed

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$;

1: Specify $\bar{\boldsymbol{\phi}}^{(1)} \in \Pi$ some pure initial policy, $\bar{\boldsymbol{h}}^{(0)} = \boldsymbol{r}^{\bar{\phi}(1)}_{\downarrow}$ and set $k = 1$, $l = 1$;
2: **repeat**
3: $\boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(l-1)}) = \arg\min_{\boldsymbol{P}\in\boldsymbol{P}^{\bar{\phi}(k)}_{\updownarrow}} (\boldsymbol{P}\bar{\boldsymbol{h}}^{(l-1)})$;
Solve
$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\downarrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(l-1)})\right)\bar{\boldsymbol{h}}^{(l)} + \bar{H}_{\downarrow}\mathbb{1};$$
by setting $\bar{\boldsymbol{h}}^{(l)}(i_0) = 0$ for some fixed state $i_0 \in \mathcal{S}$. Compute $\bar{\boldsymbol{G}}^{(k)}_{\downarrow} = \left(\boldsymbol{I} - \boldsymbol{M}_{\downarrow}(\boldsymbol{P}^{\bar{\boldsymbol{\phi}}^{(k)}}_{\updownarrow}, \bar{\boldsymbol{h}}^{(l-1)})\right)$. Then $\tilde{\boldsymbol{G}}^{(k)}_{\downarrow}$ is the matrix with the column corresponding to state i_0 replaced by a column of 1's. Solve the linear system
$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}} = \tilde{\boldsymbol{G}}^{(k)}_{\downarrow}\boldsymbol{w}$$
where $\bar{H}^{(k)}_{\downarrow}$ is the i_0th component of the solution vector $\boldsymbol{w}$ and $\bar{\boldsymbol{h}}^{(l)}(i) = \boldsymbol{w}(i)$ for $i \neq i_0$;
4: **until** $\left\|\bar{\boldsymbol{h}}^{(l)} - \bar{\boldsymbol{h}}^{(l-1)}\right\| < \epsilon$;
5: $\bar{\boldsymbol{h}} = \bar{\boldsymbol{h}}^{(l)}$;
6: **for** $i \in \mathcal{S}$; **do**
7: **for** $a \in \mathcal{A}$; **do**
8: $f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \bar{\boldsymbol{h}}) = \min_{\boldsymbol{P}\in\boldsymbol{P}^a_{\updownarrow}} \left(\boldsymbol{P}(i\bullet)\bar{\boldsymbol{h}}\right)$;
9: Choose $\bar{\boldsymbol{\phi}}^{(k+1)}(i)$ to satisfy
$$\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}} \left(\boldsymbol{r}^a_{\downarrow}(i) + f^a_{\downarrow}(\boldsymbol{P}^a_{\updownarrow}(i\bullet), \bar{\boldsymbol{h}})\right)$$
keeping $\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \bar{\boldsymbol{\phi}}^{(k)}(i)$ when possible;
10: **if** $\bar{\boldsymbol{\phi}}^{(k+1)} = \bar{\boldsymbol{\phi}}^{(k)}$ **then**
11: Set $\bar{\boldsymbol{\phi}}^{*}_{\downarrow} = \bar{\boldsymbol{\phi}}^{(k+1)}$ and terminate. Otherwise set $k = k+1$, $l = 1$, $\bar{\boldsymbol{h}}^{(0)} = \bar{\boldsymbol{h}}$ and go to Step 2;
12: **return** An optimal policy $\bar{\boldsymbol{\phi}}^{*}_{\downarrow}$;

Algorithm 20 Policy iteration 2 for discrete-time BMDPs with average reward criterion when optimal *upper* bound should be computed

Require: BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})_{a\in\mathcal{A}}\right)$;

1: Specify $\bar{\boldsymbol{\phi}}^{(1)} \in \Pi$ some pure initial policy, $\bar{\boldsymbol{h}}^{(0)} = \boldsymbol{r}_{\uparrow}^{\bar{\phi}(1)}$ and set $k = 1$, $l = 1$;

2: **repeat**

3: $\boldsymbol{M}_{\uparrow}(\boldsymbol{P}_{\updownarrow}^{\bar{\boldsymbol{\phi}}^{(k)}}, \bar{\boldsymbol{h}}^{(l-1)}) = \arg\max_{\boldsymbol{P}\in\boldsymbol{P}_{\updownarrow}^{\bar{\phi}(k)}} (\boldsymbol{P}\bar{\boldsymbol{h}}^{(l-1)})$;

Solve

$$\boldsymbol{r}_{\uparrow}^{\bar{\boldsymbol{\phi}}^{(k)}} = \left(\boldsymbol{I} - \boldsymbol{M}_{\uparrow}(\boldsymbol{P}_{\updownarrow}^{\bar{\boldsymbol{\phi}}^{(k)}}, \bar{\boldsymbol{h}}^{(l-1)})\right)\bar{\boldsymbol{h}}^{(l)} + \bar{H}_{\uparrow}\mathbb{1};$$

by setting $\bar{\boldsymbol{h}}^{l}(i_0) = 0$ for some fixed state $i_0 \in \mathcal{S}$. Compute $\bar{\boldsymbol{G}}_{\uparrow}^{(k)} = \left(\boldsymbol{I} - \boldsymbol{M}_{\uparrow}(\boldsymbol{P}_{\updownarrow}^{\bar{\boldsymbol{\phi}}^{(k)}}, \bar{\boldsymbol{h}}^{(l-1)})\right)$. Then $\tilde{\boldsymbol{G}}_{\uparrow}^{(k)}$ is the matrix with the column corresponding to state i_0 replaced by a column of 1's. Solve the linear system

$$\boldsymbol{r}^{\bar{\boldsymbol{\phi}}^{(k)}} = \tilde{\boldsymbol{G}}_{\uparrow}^{(k)}\boldsymbol{w}$$

where $\bar{H}_{\uparrow}^{(k)}$ is the i_0th component of the solution vector $\boldsymbol{w}$ and $\bar{\boldsymbol{h}}^{(l)}(i) = \boldsymbol{w}(i)$ for $i \neq i_0$;

4: **until** $\left\|\bar{\boldsymbol{h}}^{(l)} - \bar{\boldsymbol{h}}^{(l-1)}\right\| < \epsilon$;

5: $\bar{\boldsymbol{h}} = \bar{\boldsymbol{h}}^{(l)}$;

6: **for** $i \in \mathcal{S}$; **do**

7: **for** $a \in \mathcal{A}$; **do**

8: $f_{\uparrow}^{a}(\boldsymbol{P}_{\updownarrow}^{a}(i\bullet), \bar{\boldsymbol{h}}) = \max_{\boldsymbol{P}\in\boldsymbol{P}_{\updownarrow}^{a}}\left(\boldsymbol{P}(i\bullet)\bar{\boldsymbol{h}})\right)$;

9: Choose $\bar{\boldsymbol{\phi}}^{(k+1)}(i)$ to satisfy

$$\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \arg\max_{a\in\mathcal{A}}\left(\boldsymbol{r}_{\uparrow}^{a}(i) + f_{\uparrow}^{a}(\boldsymbol{P}_{\updownarrow}^{a}(i\bullet), \bar{\boldsymbol{h}})\right)$$

keeping $\bar{\boldsymbol{\phi}}^{(k+1)}(i) = \bar{\boldsymbol{\phi}}^{(k)}(i)$ when possible;

10: **if** $\bar{\boldsymbol{\phi}}^{(k+1)} = \bar{\boldsymbol{\phi}}^{(k)}$ **then**

11: Set $\bar{\boldsymbol{\phi}}_{\uparrow}^{*} = \bar{\boldsymbol{\phi}}^{(k+1)}$ and terminate. Otherwise set $k = k + 1$, $l = 1$, $\bar{\boldsymbol{h}}^{(0)} = \bar{\boldsymbol{h}}$ and go to Step 2;

12: **return** An optimal policy $\bar{\boldsymbol{\phi}}_{\uparrow}^{*}$;

1.4 Solution methods for BSMDPs

Algorithm 21 Transformation method for BSMDPs with average reward criterion

---

Require: BSMDP $\Big((\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{F}^a_{\updownarrow}(i,t))_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})\Big)$;

for $(\rightarrow, \leftarrow) \in \{(\uparrow,\downarrow),(\downarrow,\uparrow)\}$ **do**

  for $(i,a) \in \mathcal{S}\times\mathcal{A}$ **do**

    Let the $\{(\boldsymbol{p}^{(i)}_{\downarrow},\ \boldsymbol{D}^{(i)}_{0\downarrow})\}^a_{i\in\mathcal{S}}$ and $\{(\boldsymbol{p}^{(i)}_{\uparrow},\ \boldsymbol{D}^{(i)}_{0\uparrow})\}^a_{i\in\mathcal{S}}$ be the sets of Phase-type distributions corresponding to the action a;

    $\boldsymbol{y}^a_{\rightarrow}(i) = -\boldsymbol{p}^{(i)}_{\rightarrow}(\boldsymbol{D}^{(i)}_{0\rightarrow})^{-1}\mathbb{1}$;

    $\boldsymbol{s}^a_{\rightarrow}(i) = \frac{\boldsymbol{r}^a_{\rightarrow}(i)}{\boldsymbol{y}^a_{\rightarrow}(i)}$;

  $\eta = \min_{i\in\mathcal{S},a\in\mathcal{A}} \frac{\boldsymbol{y}^a(i)_{\downarrow}}{1-\boldsymbol{P}^a_{\rightarrow}(i,i)}$;

  for $a \in \mathcal{A}$ **do**

    $\boldsymbol{S} = \eta(\boldsymbol{P}^a_{\rightarrow} - \boldsymbol{I})$;

    $\boldsymbol{d} = \mathrm{diag}(\boldsymbol{S})$;

    $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{y}^a_{\rightarrow})^{-1}(\boldsymbol{S} - \mathrm{diag}(\boldsymbol{d}))$;

    $\boldsymbol{e} = \mathrm{diag}(\boldsymbol{y}^a_{\leftarrow})^{-1}\boldsymbol{d} + \mathbb{1}$;

    $\boldsymbol{Q}^a_{\rightarrow} = \mathrm{diag}(\boldsymbol{e}) + \boldsymbol{R}$;

return BMDP $\Big((\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{s}_{\updownarrow})_{a\in\mathcal{A}}\Big)$, η ;

Algorithm 22 Analysis of BSMDPs under the average reward criterion

---

Require: BSMDP $\Big((\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{F}^a_{\updownarrow}(i,t))_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})\Big)$;

Transform $\Big((\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{F}^a_{\updownarrow}(i,t))_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})\Big)$ into BMDP $\Big((\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{s}_{\updownarrow})_{a\in\mathcal{A}}\Big)$, $\eta \in \mathbb{R}$ using Algorithm 21.

Analyze BMDP $\Big(\mathcal{S}, \mathcal{A}, (\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{s}_{\updownarrow})_{a\in\mathcal{A}}\Big)$ with one of methods 16 , 17, 18, 19, 20 and obtain policy ϕ, gain H, bias vector $\boldsymbol{h}$.

return $(\phi, H, \eta\boldsymbol{h})$

Algorithm 23 Transformation method for BSMDPs with discounted reward criterion

Require: BSMDP $\Big((\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{F}^a_{\updownarrow}(i,t))_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})\Big)$, discount rate β;

$(\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}} = (\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}$;

for $a \in \mathcal{A}$ **do**

  Let the $\{(\boldsymbol{p}^{(i)}_{\downarrow}, \boldsymbol{D}^{(i)}_{0\downarrow})\}^a_{i\in\mathcal{S}}$ and $\{(\boldsymbol{p}^{(i)}_{\uparrow}, \boldsymbol{D}^{(i)}_{0\uparrow})\}^a_{i\in\mathcal{S}}$ be the sets of Phase-type distributions corresponding to the action a;

for $a \in \mathcal{A}$ **do**

  for $i \in \mathcal{S}$ **do**

    Using $\{(\boldsymbol{p}^{(i)}_{\downarrow}, \boldsymbol{D}^{(i)}_{0\downarrow})\}^a_{i\in\mathcal{S}}$ compute lower bound vectors and matrices
    $\boldsymbol{d}^{(i)}_{1\downarrow} = -\boldsymbol{D}^{(i)}_{0\downarrow}\mathbb{1}$;
    Set $\boldsymbol{P}^{(i)}_{\downarrow} = \boldsymbol{D}^{(i)}_{0\downarrow} - \beta\boldsymbol{I}$ and $\lambda_{\downarrow} = \max_{\forall i,j\in\mathcal{S}} |\boldsymbol{P}^{(i)}_{\downarrow}(i,j)|$;
    $\boldsymbol{P}^{(i)}_{\downarrow} = \frac{1}{\lambda_{\downarrow}}\boldsymbol{P}^{(i)}_{\downarrow} + \boldsymbol{I}$,
    $\boldsymbol{d}^{(i)}_{1\downarrow} = \frac{1}{\lambda_{\downarrow}}\boldsymbol{d}^{(i)}_{1\downarrow}$ and the time step $\Delta_{\downarrow} = 10/\lambda_{\downarrow}$;
    Compute $\boldsymbol{s}^a_{\downarrow}(i) = \boldsymbol{r}^a_{\downarrow}(i)\int\limits_0^{\infty}(1 - F^a_{\downarrow}(i,t))e^{-\beta t}dt$ and
    the discount factor $\boldsymbol{\gamma}^a_{\downarrow}(i) = \int\limits_0^{\infty} f^a_{\downarrow}(i,t)e^{-\beta t}dt$ with the uniformization based method [1] using $\boldsymbol{P}^{(i)}_{\downarrow}$, $\boldsymbol{d}^{(i)}_{1\downarrow}$, $\Delta_{\downarrow}$.

    Using $\{(\boldsymbol{p}^{(i)}_{\uparrow}, \boldsymbol{D}^{(i)}_{0\uparrow})\}^a_{i\in\mathcal{S}}$ compute upper bound vectors and matrices
    $\boldsymbol{d}^{(i)}_{1\uparrow} = -\boldsymbol{D}^{(i)}_{0\uparrow}\mathbb{1}$;
    $\boldsymbol{P}^{(i)}_{\uparrow} = \boldsymbol{D}^{(i)}_{0\uparrow} - \beta\boldsymbol{I}$;
    $\lambda_{\uparrow} = \max_{\forall i,j\in\mathcal{S}} |\boldsymbol{P}^{(i)}_{\uparrow}(i,j)|$;
    $\boldsymbol{P}^{(i)}_{\uparrow} = \frac{1}{\lambda_{\uparrow}}\boldsymbol{P}^{(i)}_{\uparrow} + \boldsymbol{I}$;
    $\boldsymbol{d}^{(i)}_{1\uparrow} = \frac{1}{\lambda_{\uparrow}}\boldsymbol{d}^{(i)}_{1\uparrow}$ and the time step $\Delta_{\uparrow} = 10/\lambda_{\uparrow}$;
    Compute $\boldsymbol{s}^a_{\uparrow}(i) = \boldsymbol{r}^a_{\uparrow}(i)\int\limits_0^{\infty}(1 - F^a_{\uparrow}(i,t))e^{-\beta t}dt$ and
    the discount factor $\boldsymbol{\gamma}^a_{\uparrow}(i) = \int\limits_0^{\infty} f^a_{\uparrow}(i,t)e^{-\beta t}dt$ with the uniformization based method [1] using $\boldsymbol{P}^{(i)}_{\uparrow}$, $\boldsymbol{d}^{(i)}_{1\uparrow}$, $\Delta_{\uparrow}$.

return $(\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}}$, $(\boldsymbol{s}_{\updownarrow})_{a\in\mathcal{A}}$, $(\boldsymbol{\gamma}_{\updownarrow})_{a\in\mathcal{A}}$;

Algorithm 24 Analysis of BSMDPs under the discounted reward criterion

---

Require: BSMDP $\left((\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{F}^a_{\updownarrow}(i,t))_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})\right)$, discount rate $\gamma \in [0,1)$;

1: Transform $\left((\boldsymbol{P}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{F}^a_{\updownarrow}(i,t))_{a\in\mathcal{A}}, (\boldsymbol{r}^a_{\updownarrow})\right)$ into BMDP $\left((\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{s}_{\updownarrow})_{a\in\mathcal{A}}\right)$, state-dependent discount factor vector $(\boldsymbol{\gamma}^a_{\updownarrow} \in \mathbb{R}^{|\mathcal{S}|})_{a\in\mathcal{A}}$ using Algorithm 23.

2: Analyze BMDP $\left(\mathcal{S}, \mathcal{A}, (\boldsymbol{Q}^a_{\updownarrow})_{a\in\mathcal{A}}, (\boldsymbol{s}_{\updownarrow})_{a\in\mathcal{A}}\right)$ with one of methods 13, 14, 15 under discount factor $(\boldsymbol{\gamma}^a_{\updownarrow})_{a\in\mathcal{A}}$ and obtain policy ϕ, gain vector $\boldsymbol{v}$.

3: **return** $(\phi, \boldsymbol{v})$

1.5 Solution methods for continuous-time processes

Algorithm 25 Uniformization method for CTMDPs

Require: CTMDP $(\mathcal{S}, \mathcal{A}, \{\boldsymbol{Q}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{r}^a\}_{a\in A})$, discount factor β, *discounted* is true for the discounted reward measure and false else.

$\lambda = \max_{\forall i\in\mathcal{S},\forall a\in\mathcal{A}} |\boldsymbol{Q}^a(i,i)|$;
for $a \in \mathcal{A}$ **do**
  $\boldsymbol{P}^a = \boldsymbol{I} + \frac{1}{\lambda}\boldsymbol{Q}^a$;
  if *discounted* **then**
    $\boldsymbol{z}^a(i) = \frac{\boldsymbol{r}^a(i)}{\lambda+\beta}, \quad \forall\ i \in \mathcal{S}$;
  else
    $\boldsymbol{z}^a(i) = \frac{\boldsymbol{r}^a(i)}{\lambda}, \quad \forall\ i \in \mathcal{S}$;
if *discounted* **then**
  $\gamma = \frac{\lambda}{\lambda+\beta}$;
return Discrete-time MDP $\{\mathcal{S}, \mathcal{A}, \{\boldsymbol{P}^a\}_{a\in\mathcal{A}}, \{\boldsymbol{z}^a\}_{a\in\mathcal{A}}\}$, discount factor γ if *discounted* is true;

References

1. D. Gross and D. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32, 1984.