SIAM J. Control Optim.
Vol. 52, No. 6, pp. 3854–3886

SINGULARLY PERTURBED MARKOV DECISION PROCESSES: A MULTIRESOLUTION ALGORITHM[*]

CHIN PANG HO[†] AND PANOS PARPAS[†]

Abstract. Singular perturbation techniques allow the derivation of an aggregate model whose solution is asymptotically optimal for Markov decision processes with strong and weak interactions. We develop an algorithm that takes advantage of the asymptotic optimality of the aggregate model in order to compute the solution of the original model. We derive conditions for which the proposed algorithm has better worst case complexity than conventional contraction algorithms. Based on our complexity analysis, we show that the major benefit of aggregation is that the reduced order model is no longer ill conditioned. The reduction in the number of states (due to aggregation) is a secondary benefit. This is a surprising result since intuition would suggest that the reduced order model can be solved more efficiently because it has fewer states. However, we show that this is not necessarily the case. Our theoretical analysis and numerical experiments show that the proposed algorithm can compute the optimal solution with a reduction in computational complexity and without any penalty in accuracy.

Key words. Markov decision processes, multiscale modeling, multigrid methods, weak and strong interactions

AMS subject classification. 49L20

DOI. 10.1137/130944254

1. Introduction.

Recently there has been considerable interest in modeling and control of stochastic dynamics across different timescales. Typical applications appear in molecular dynamics [7], networked systems [11], manufacturing [15], and optimal control of energy systems [12], to name just a few. Controlling dynamics across different scales is computationally difficult, and a considerable amount of literature has been devoted to the challenge of finding approximate models that capture the effective dynamics of the system. The main techniques used for optimal control are based around aggregation, averaging, and homogenization. Starting from the work of Simon and Ando [16], hierarchical decomposition and aggregation has been at the core of approximation techniques for modeling and controlling dynamics across different scales. The literature around this topic is substantial, and we refer the interested reader to [10] for early work on singular perturbation techniques in optimal control. The averaging principle and applications in manufacturing are described in [15]. The homogenization for deterministic optimal control problems has been studied in [1]. The recent research monograph by Yin and Zhang [18] describes the main mathematical results in the context of stochastic optimal control using the theory of singularly perturbed Markov processes. The mathematical framework described in [18] is the one we adopt in this paper. The main result of the aggregation techniques and averaging principles reviewed in [15] and [18] is the derivation of an approximate model that captures the slow dynamics of the system. The approximate model is based on an asymptotic analysis of a singularly perturbed control problem. (See [18] for details and section 2 of this paper for precise definitions.)

[*] Received by the editors November 5, 2013; accepted for publication (in revised form) September 10, 2014; published electronically December 10, 2014. This work was partially supported by an FP7 Marie Curie Career Integration Grant (PCIG11-GA-2012-321698 SOC-MP-ES) and EPSRC grant EP/K040723/1.

http://www.siam.org/journals/sicon/52-6/94425.html

[†] Department of Computing, Imperial College London, 180 Queen's Gate, London, SW7 2AZ, UK (c.ho12@imperial.ac.uk, p.parpas@imperial.ac.uk).

The mathematical properties and especially the use of asymptotic techniques coupled with the perturbation approach for controlling Markov processes have been extensively studied. However, numerical methods that take into consideration the specific structure of multiscale Markov processes have not received much attention. Given all the work that has gone into the development of aggregate models, it is surprising that the obvious question of whether the reduced order models can be solved more efficiently than the original model has not been addressed. We take the first steps toward answering this question for a particular class of multiscale Markov processes. Based on our complexity analysis, we show that the major benefit of aggregation is that the reduced order model is no longer ill conditioned, and the reduction in the number of states (due to aggregation) is a secondary benefit. This is a surprising result since intuition would suggest that the reduced order model can be solved more efficiently because it has fewer states. However, it will be shown later that this is not necessarily the case. There is no standard definition for an ill conditioned Markov decision process (MDP). In the context of this paper, an MDP is ill conditioned if the contraction modulus of value iteration is approximately equal to one. This means that progress at each iteration will be extremely slow. We propose a class of multiresolution contraction algorithms that are not sensitive to the ill conditioning of weakly connected MDPs. Because we are considering a particular class of MDPs, we are able to improve the worst case complexity of algorithms based on value iteration. We illustrate our approach on value iteration, but any contraction algorithm can potentially be improved using the proposed scheme.

It is important to stress that the proposed algorithm aims to solve the original model and not just obtain an approximation using the aggregate model. The aggregate model is only asymptotically optimal, and our algorithm exploits its approximate optimality to reduce the number of iterations with the high dimensional (and often ill conditioned) model. Our algorithm is ideal when there is some scale separation, but it is not known whether there is sufficient scale separation to just solve the approximate model. This setting is the most frequently scenario encountered in practice. For simplicity, we study a multiscale MDP (MMDP) with two timescales, but generalizing the results to problems with more than two timescales is straightforward.

The rest of the paper is structured as follows: In section 2 we define the notation we use and provide a review of existing results. In section 3 we review complexity results for the value iteration algorithm. We extend some known results from discrete time to continuous time and give particular emphasis to MMDPs. In section 4 we review the full approximation scheme (FAS). The FAS can be used to accomplish some of the objectives we set to achieve in this paper, i.e., take advantage of the structure of MMDPs to improve the computational efficiency of algorithms for this class of MDPs. The FAS is a nonlinear extension to the traditional multigrid scheme, and in section 4, we show that it may not be an appropriate choice for MDPs. Based on our observations of the complexity of MMDPs in section 3 and the FAS scheme in section 4, we propose an alternative scheme in section 5. We named our proposed scheme the alternating multiresolution scheme (AMS) since it uses features from the FAS and known results regarding the quality of the approximate (aggregate) model. In section 6, we propose a refinement of our scheme that allows the scheme to be applied to problems that have a large number of actions. Finally, in section 7, we illustrate the proposed scheme on two applications, one from manufacturing and one from chemistry.

2. Multiscale Markov decision processes.

The notation and framework for MMDPs that we adopt in this paper is standard, and more information and results can be obtained in [18].

2.1. Markov decision processes.

Let $x_h(t)$ denote the state of a discrete state continuous time MDP at time t. We use the subscript h to denote processes that capture effects at the fast timescale h. We assume that the chain can take one of a finite number states $\mathcal{X}^h \triangleq \{l_1, l_2, \dots, l_N\}$. For each of the states i, $1 \leq i \leq N$, the available actions of state i are denoted by the set $\mathcal{A}_i^h$. A policy $u_h : \mathcal{X}^h \to \mathcal{A}^h$ maps states into actions and is described by $u_h = (a_1, a_2, \dots, a_N)$, where $a_i \in \mathcal{A}_i^h$ for $\forall i \in \mathcal{X}^h$. The space of all policies u_h's is denoted by $\mathcal{U}^h \triangleq \{(a_1, a_2, \dots, a_N) : a_i \in \mathcal{A}_i^h, i = 1, 2, \dots, N\}$, and we use $\mathcal{A}^h$ to denote the space of all possible actions, i.e., $\mathcal{A}^h = \cup_{i=1}^N \mathcal{A}_i^h$. We assume that we are given a cost function $G^h : \mathcal{X}^h \times \mathcal{A}^h \to \mathbb{R}$ that measures the cost associated with a particular state-action pair. We will focus on the infinite horizon case and denote the discount factor by ρ. All the results reported in the paper can be extended to the finite horizon case. We use $\mathcal{MDP}(N, L)$ to denote the class of problems for which $|\mathcal{X}^h| = N$, $|\mathcal{A}_i^h| = L$ for $i = 1, 2, \dots, N$. It is easy to generalize our results to the case where each of the action spaces $\mathcal{A}_i^h$ have different cardinality $|\mathcal{A}_i^h| = L_i$, but for ease of exposition we assume that $|\mathcal{A}_i^h| = L$ for $i = 1, 2, \dots, N$. With the notation introduced above we are now in a position to state the class of problems we study in this paper,

$$\min_{u_h \in \mathcal{U}^h} \quad J^h(i, u_h) = \mathbb{E}\left[\int_0^\infty e^{-\rho t} G^h(x_h(t), u_h(x_h(t)))\ \mathrm{d}t \middle| x_h(0) = i\right]. \tag{2.1}$$

The expectation above is taken with respect to a probability matrix P, and we use $P_{i,j}(t, s)$ to denote the probability of the process transitioning to state j at time t given that it starts from state i at time s. According to the theory of Markov processes, the transition matrix satisfies the following equation:

$$\frac{dP(t,s)}{dt} = P(t,s) Q_h^\epsilon(u_h), \quad P(s,s) = I_N, \tag{2.2}$$

where Q_h^ϵ denotes the infinitesimal generator of x_h, and I_N denotes the $N \times N$ identity matrix. We are focusing on a Markov process with a multiscale structure, and so we assume the generator of the process is defined as follows:

$$Q_h^\epsilon(u_h) = \frac{1}{\epsilon} \hat{Q}(u_h) + W(u_h), \tag{2.3}$$

where $\hat{Q}(u_h) = \mathrm{diag}(\hat{Q}_1(u_h), \hat{Q}_2(u_h), \dots \hat{Q}_m(u_h))$ is a block diagonal matrix with m blocks, with $\hat{Q}_k(u_h) \in \mathbb{R}^{n_k \times n_k}$ and $\sum_{j=1}^m n_j = N$, for $k = 1, 2, \dots, m$. We further assume that each $\hat{Q}_k(u_h)$ is a weakly irreducible Markov generator. For ease of exposition, we assume that all blocks have the same size (n). All our results can easily be generalized to the case where each block has size n_i. We use $\mathcal{X}_k = \{l_{k1}, \dots, l_{kn_k}\}$, $k = 1, \dots, m$, to denote the states corresponding to $\hat{Q}_k$. This decomposition is done so that $\mathcal{X} = \cup_{k=1}^m \mathcal{X}_k$. The small parameter ϵ is used to capture the multiscale structure of the process. When $\epsilon \ll 1$, the Markov process jumps frequently between the states within a block $\mathcal{X}_k$ and less frequently between states that belong to different blocks. The matrix $W(u_h)$ is also assumed to be a Markov generator, and it is used to model the transition between the blocks. The smaller the ϵ, the faster the transitions inside the blocks. As ϵ approaches zero, the transitions inside the blocks happen at such a fast rate that the process can be approximated by the equilibrium distribution inside each of the blocks. This idea can be made rigorous, and we refer the reader to Chapter 6 of [18] for the details. Our aim is to study the complexity and propose an efficient algorithm for the solution of the stochastic control problem in (2.1).

The class of weakly connected process will be denoted by $\mathcal{MMDP}(\epsilon, n, m, L)$, which is a subclass of $\mathcal{MDP}(nm, L)$. Using the dynamic programming principle, it can be shown that the value function associated with the problem

$$v_h(i) = \min_{u_h \in \mathcal{U}^h} J^h(i, u_h) \tag{2.4}$$

satisfies the so-called the Hamilton–Jacobi–Bellman (HJB) equation,

$$\rho v_h(i) = \min_{a \in \mathcal{A}_i^h} \left[G^h(i,a) + \sum_{j \in \mathcal{X}^h, j \neq i} q_{ij}^h(a)[v_h(j) - v_h(i)] \right]. \tag{2.5}$$

Notice that we use min instead of inf in our problem definition because the action space is discrete and we will assume the G^h is bounded. It was shown in [18] that (2.5) is equivalent to

$$v_h(i) = \min_{a \in \mathcal{A}_i^h} \left[\frac{G^h(i,a)}{|q_{ii}^h(a)| + \rho} + \sum_{j \neq i} \frac{q_{ij}^h(a)}{|q_{ii}^h(a)| + \rho} v_h(j) \right]. \tag{2.6}$$

Our analysis will be based on the properties of the contraction operator derived from value iteration. With that in mind we can rewrite the problem of computing the value function as the solution of the nonlinear equation

$$A_h v_h = 0,$$

where A_h is a nonlinear operator defined as follows:

$$(A_h v_h)(i) \triangleq \min_{a \in \mathcal{A}_i^h} \left[\frac{G^h(i,a)}{|q_{ii}^h(a)| + \rho} + \sum_{j \neq i} \frac{q_{ij}^h(a)}{|q_{ii}^h(a)| + \rho} v_h(j) \right] - v_h(i). \tag{2.7}$$

The contraction operator will be denoted by T_h and is defined below,

$$(T_h v_h)(i) \triangleq \min_{a \in \mathcal{A}_i^h} \left[\frac{G^h(i,a)}{|q_{ii}^h(a)| + \rho} + \sum_{j \neq i} \frac{q_{ij}^h(a)}{|q_{ii}^h(a)| + \rho} v_h(j) \right]. \tag{2.8}$$

So far we have not used our assumption that $\epsilon \ll 1$. This setting has been extensively studied in the last thirty years, and in the next section we summarize the results we will need in our analysis.

2.2. The coarse model. The computational cost of solving (2.1) exactly is extremely high when the Markov process has a large number of states. Many researchers noticed that if the problem has the multiscale structure described in the previous section, then the computational costs can be reduced by considering an approximate model. In the approximate model, each set of states associated with each of the "fast" blocks is aggregated to a single state. For this reason the resulting approximate model is called the aggregate model. In this paper, however, we will adopt the terminologies from the multigrid community by using *fine model* (defined in (2.1)) and *coarse model* (defined in (2.9) below) instead of exact and aggregate model. It can be shown that if ϵ is small enough, the coarse model becomes arbitrarily accurate. There are many results of this type for the model described in the previous section, as well as generalizations to different models. These results are described in [18], and we refer the reader there for a comprehensive literature review. In our work we will need to make use of the coarse model and we describe the notation we use below.

The state space of the coarse model is denoted by $\mathcal{X}^H \triangleq \{l'_1, l'_2, \dots, l'_m\}$, where each state i in the coarse model represents block i in the fine model. The available actions of state i in the coarse model a_i^H are combinations of the available actions in block i, and they form the action space $\mathcal{A}_i^H \triangleq \{(a_1^i, a_2^i, \dots, a_n^i) : a_j^i \in \mathcal{A}^h_{\mathcal{X}_i(j)}\}$. Therefore, u_H is the policy of the coarse model, and it takes values from the policy space $\mathcal{U}^H \triangleq \{(a_1^H, a_2^H, \dots, a_m^H) : a_i^H \in \mathcal{A}_i^H, i = 1, 2, \dots, m\}$. The coarse model is an $\mathcal{MDP}(m, L^n)$ model.

In order to define the coarse model, we also need to define both the coarse Markov generator and the coarse objective function. Let $\varphi_1(u_H), \dots, \varphi_m(u_H)$ denote the stationary distributions of the blocks $1, 2, \dots, m$ in the form of column vectors under policy u_H. We obtain the corresponding Markov generator,

$$Q_H(u_H) = \varphi(u_H) W(u_H) \tilde{\mathbb{1}},$$

where

$$\begin{aligned} \varphi(u_H) &= \operatorname{diag}(\varphi_1^T(u_H), \varphi_2^T(u_H), \dots, \varphi_m^T(u_H)), \\ \tilde{\mathbb{1}} &= \operatorname{diag}(\underbrace{\mathbb{1}_{n\times 1}, \mathbb{1}_{n\times 1}, \dots, \mathbb{1}_{n\times 1}}_{m \text{ copies}}), \end{aligned}$$

where $\mathbb{1}_{n\times 1} \triangleq (1, 1, \dots, 1)^T \in \mathbb{R}^{n\times 1}$, and $\operatorname{diag}(\cdot)$ is a function which maps its argument to a diagonal matrix. The coarse cost function is given by

$$G^H(i, u_H) = \sum_{k=1}^{n} (\varphi_i(u_H))_k G^h(\mathcal{X}_i(k), u_H) \quad \forall i \in \mathcal{X}^H.$$

Given the notation above, the coarse model is

$$\begin{aligned} \min_{u_H} \quad & J^H(i, u_H) = \mathbb{E}\left[\int_0^\infty e^{-\rho t} G^H(x_H(t), u_H(x_H(t)))\ \mathrm{d}t\right], \\ \text{s.t.} \quad & x_H \sim Q_H(u_H(x_H(t)))\ ,\ t \geq 0, \\ & x_H(0) = i \quad , \quad u_H \in \mathcal{U}^H, \\ & v_H(i) = \min_{u_H \in \mathcal{U}^H} J^H(i, u_H). \end{aligned} \tag{2.9}$$

The corresponding HJB equation becomes

$$v_H(i) = \min_{a_H \in \mathcal{A}_i^H} \left[\frac{G^H(i, a_H)}{|q_{ii}^H(a_H)| + \rho} + \sum_{j\neq i} \frac{q_{ij}^H(a_H)}{|q_{ii}^H(a_H)| + \rho} v_H(j)\right]. \tag{2.10}$$

For the problem, we denote by A_H and T_H the nonlinear operator and its corresponding contraction operator, respectively. Using singular perturbation techniques (see [18]), it can be shown that under the assumptions made in this paper the following result holds:

$$v_H^\star(k) \to v_h^\star(i) \quad \text{for} \quad i \in \mathcal{X}_k \quad \text{as} \quad \epsilon \to 0,$$

where $v_H^\star$ denotes the solution of (2.9) and $v_h^\star$ denotes the solution of (2.1). Also,

$$|v_H^\star(k) - v_h^\star(i)| = O(\epsilon) \quad \text{for} \quad i \in \mathcal{X}_k. \tag{2.11}$$

3. Computational complexity of multiscale Markov decision processes. In this section we review the complexity of value iteration for the MMDP model introduced in the previous section. The purpose of this section is twofold. First, the complexity of MDPs in continuous time has not received as much attention as that of their discrete-time counterparts. Even though the complexity results here are new, they are straightforward generalizations of results from discrete time. The second and main purpose of this section is to point out that the convergence rate of value iteration becomes arbitrarily bad when ϵ becomes small. We believe that this insight is an important consideration when designing algorithms for multiscale processes. Previously, it was claimed that the coarse model might have lower complexity because it has fewer states than the fine model. Here we show that an additional and (as discussed later on) more important advantage is that the coarse model is better conditioned since it does not depend on ϵ. We also show that the complexity results below are tight.

3.1. Value-iteration. Value iteration is one of the first methods to be proposed to solve dynamic programming problems. Value iteration is used to compute the value function. After the value function, $v_h^\star(x)$, is obtained, the optimal policy $u_h^\star$ can be obtained by

$$u_h^\star(i) \in \arg\min_{a \in \mathcal{A}_i^h} \left[\frac{G^h(i,a)}{|q_{ii}^h(a)| + \rho} + \sum_{j \neq i} \frac{q_{ij}^h(a)}{|q_{ii}^h(a)| + \rho} v_h^\star(j) \right].$$

Therefore, solving the HJB equation is equivalent to solving for the optimal policy $u_h^\star$ [2]. State-of-the-art deterministic methods for solving HJB equations fall into three broad categories: linear programming, policy-iteration, and value-iteration [13]. In this paper, value-iteration is applied to solve the HJB equation even though the central idea of this paper can be applied to all three of the methods. The extension of the proposed framework to the stochastic case (e.g., to approximate dynamic programming techniques) is beyond the scope of the current paper. Value iteration is simply defined as

$$v_h^{k+1} = T_h v_h^k. \tag{3.1}$$

The nonlinear operator T_h was defined in (2.8), and it is well known that it is a contraction mapping,

$$\|T_h v_1 - T_h v_2\|_\infty \leq \alpha_h \|v_1 - v_2\|_\infty, \tag{3.2}$$

where $v_1, v_2 \in \mathbb{R}^N$, α_h is the Lipschitz constant, $0 < \alpha_h < 1$. For the MDP model we study in this paper, the Lipschitz constant is given by

$$\alpha_h = \max_{i \in \mathcal{X}^h, a \in \mathcal{A}_i^h} \frac{|q_{ii}^h(a)|}{|q_{ii}^h(a)| + \rho}. \tag{3.3}$$

The Lipschitz constant above is derived in [18], where the HJB equation (2.6) is reformulated as an HJB equation for discrete time problems with a discount factor α_h in (3.3). For discrete time problems, the discount rate itself is the Lipschitz constant of the problem [13]. Using the Banach fixed point theorem [17], it can be shown that for an initial guess v_h^0, one can compute the solution of the HJB equation by iteratively applying T_h on v_h^0,

$$v_h^{\tau_h} \triangleq T_h^{\tau_h} v_h^0 = \underbrace{(T_h \circ T_h \circ \cdots \circ T_h)}_{\tau_h \text{ copies}} v_h^0 \to v_h^\star \quad \text{as} \quad \tau_h \to \infty.$$

The Lipschitz constant α_h is an upper bound for the convergence rate of the value-iteration algorithm. In particular,

$$\|v_h^\star - v_h^{\tau_h}\|_\infty \leq \alpha_h \|v_h^\star - v_h^{\tau_h - 1}\|_\infty \leq \alpha_h^{\tau_h} \|v_h^\star - v_h^0\|_\infty. \tag{3.4}$$

A smaller α_h guarantees a faster convergence rate for algorithm. Other than equation (3.4), we will make use of the following well-known properties of contraction mappings,

$$\|v_h^\star - v_h^{\tau_h}\|_\infty \leq \frac{\alpha_h}{1-\alpha_h} \|v_h^{\tau_h} - v_h^{\tau_h - 1}\|_\infty, \tag{3.5}$$

$$\leq \frac{\alpha_h^{\tau_h}}{1-\alpha_h} \|v_h^1 - v_h^0\|_\infty. \tag{3.6}$$

3.2. Model assumptions. In this section, we state our assumptions, and these will hold throughout the paper. Some of our results will be asymptotic and will rely on the assumption that the problem has multiscale structure stronger than certain threshold ϵ_0. To be precise we assume that ϵ is such that the value function of the fine model (2.1) and its corresponding coarse model satisfy the inequality

$$|v_H^\star(k) - v_h^\star(i)| \leq \tilde{K}\epsilon \;\forall \epsilon < \epsilon_0, \; \forall i \in \mathcal{X}_k \tag{3.7}$$

for some constants $\tilde{K}$ and $\epsilon_0 < 1$. That this inequality holds for an ϵ small enough follows from [18, Theorem 7.10, p. 273].

Our second main assumption is that the objective function is bounded. We will assume that there exists a constant ζ such that

$$0 \leq G^h(x_h, a_h) \leq \zeta \;\forall x_h \in \mathcal{X}^h, a_h \in \mathcal{A}^h.$$

The bounded assumption is needed to avoid trivialities. Since $G^h(\cdot,\cdot)$ is assumed to be bounded, the value function should also be bounded; in other words, there exists a constant $\hat{K}$ such that

$$0 \leq \|v_h^\star\|_\infty \leq \hat{K}, \; 0 \leq \|v_H^\star\|_\infty \leq \hat{K}.$$

Without loss of generality we will assume that the initial guess v_h^0 and v_H^0 are both zero vectors of the appropriate dimensions. Finally, to simplify our notation, instead of using $\hat{K}$ and $\tilde{K}$ we will directly use $K \triangleq \max\{\hat{K}, \tilde{K}\}$, where $\tilde{K}$ is defined in equation (3.7). With these two assumptions, we obtain

$$0 = \|v_h^0\|_\infty \leq \|v_h^1\|_\infty \leq \|v_h^2\|_\infty \leq \cdots \leq \|v_h^\star\|_\infty \leq K, \tag{3.8}$$

$$0 = \|v_H^0\|_\infty \leq \|v_H^1\|_\infty \leq \|v_H^2\|_\infty \leq \cdots \leq \|v_H^\star\|_\infty \leq K. \tag{3.9}$$

The above inequalities follow from the fact that both operators T_h and T_H are monotone contraction operators [2]. Also, from (3.8)–(3.9), we have

$$\|v_h^\star - v_h\|_\infty \le K,\ \|T_h v_h - v_h\|_\infty \le K \quad \forall v_h,$$
$$\|v_H^\star - v_H\|_\infty \le K,\ \|T_H v_H - v_H\|_\infty \le K \quad \forall v_H.$$

Notice that (3.7) and $\tilde{K} \le K$ gives

$$|v_H^\star(k) - v_h^\star(i)| \le K\epsilon \ \forall i \in \mathcal{X}_k. \tag{3.10}$$

3.3. Complexity. In this section we discuss the complexity of continuous time MDPs. The complexity result in this section is a variant of the existing discrete time result [6]. We use $\delta > 0$ to denote the convergence tolerance for the value iteration algorithm, i.e., the algorithm terminates when

$$\|v_h^\star - v_h^{\tau_h}\| < \delta. \tag{3.11}$$

The parameter $\delta > 0$ is user specified and since T_h is a contraction mapping, for large enough τ_h, the above inequality can be satisfied. A more interesting question is how large τ_h should be to guarantee that (3.11) holds. We answer this question in the lemma below by providing an upper bound, and then we give an example to show that this bound is tight.

Lemma 3.1. *The number of iterations in the value-iteration algorithm is bounded by*

$$\tau_h \le \max\left\{ \frac{\log\left(\dfrac{K}{(1-\alpha_h)\delta}\right)}{|\log \alpha_h|}, 0 \right\}, \tag{3.12}$$

where $K = \max\{\hat{K}, \tilde{K}\}$.

Proof. We use (3.6),

$$\|v_h^\star - v_h^{\tau_h}\|_\infty \le \frac{\alpha_h^{\tau_h}}{1-\alpha_h}\|v_h^1 - v_h^0\|_\infty \le \frac{\alpha_h^{\tau_h}}{1-\alpha_h}K, \tag{3.13}$$

where we used the fact that $\|T_h v_h^0 - v_h^0\| < K$. We then select τ_h' such that

$$\frac{\alpha_h^{\tau_h'}}{1-\alpha_h}K = \delta.$$

Rearranging the preceding equation, we obtain the following expression:

$$\tau_h' = \frac{\log\left(\dfrac{K}{(1-\alpha_h)\delta}\right)}{|\log \alpha_h|}.$$

Since $v_h^{\tau_h'}$ guarantee the desired accuracy, we have $\tau_h \le \max\{\tau_h', 0\}$. □

Equation (3.12) gives an upper bound for the number of iterations we need when using value-iteration. The complexity of the value-iteration algorithm can be easily derived from Lemma 3.1. The complexity model we consider in this paper is consistent with [6], where each arithmetic operation or comparison is considered to cost one unit of computation.

THEOREM 3.2. *The worst–case complexity for the value iteration algorithm in* (3.1) *for $\mathcal{MDP}(N, L)$ is*

$$O\left(\max\left\{\frac{\log\left(\frac{1}{(1-\alpha_h)\delta}\right)}{|\log\alpha_h|}, 0\right\}N^2L\right). \tag{3.14}$$

Proof. For the contraction operator in (2.8) and for an MDP problem $\mathcal{MDP}(N, L)$, the worst-case complexity of computing $T_h v_h$ is $O(N^2L)$, since the total complexity of the algorithm is the number of iterations multiplied by the cost per iteration. We obtain the complexity result it (3.14) by applying Lemma 3.1. □

A natural question to ask is whether the complexity result in Theorem (3.2) is tight. We end this section by showing that indeed the bound is tight.

Remark 3.3. Consider an instance of $\mathcal{MDP}(N, L)$ that satisfies the following:

- $|\mathcal{A}_i^h| = 1\ \forall i \in \mathcal{X}^h$. This assumption means that the corresponding HJB equation is a linear equation.
- The cost function $G^h(i, a_i) = g > 0\ \forall i \in \mathcal{X}^h$ is a constant. Therefore, the value function $v_h^\star(i) = \mathbf{v}^\star$ is also a positive constant.
- $q_{ii}^h = q\ \forall i \in \mathcal{X}^h$, i.e., each state has the same jump rate.
- The initial guess $v_h^0 = 0$ is a zero vector, so $v_h^0 < T_h v_h^0 < T_h^2 v_h^0 < \cdots < v_h^\star$.

Given the assumptions above, it follows that $v_h^\tau \triangleq T_h^\tau v_h^0$ are all constants, i.e., $v_h^\tau(i) = \mathbf{v}^\tau\ \forall i \in \mathcal{X}^h, \tau \in \mathbb{Z}^+$. Consider the error reduction rate between iteration τ and $\tau + 1$,

$$\begin{aligned}
\|v_h^\star - v_h^{\tau+1}\|_\infty &= \|v_h^\star - T_h^{\tau+1} v_h^0\|_\infty \\
&= \|T_h v_h^\star - T_h v_h^\tau\|_\infty \\
&= \max_{i\in\mathcal{X}^h}\left|\frac{g}{|q|+\rho} + \sum_{j\neq i}\frac{q_{ij}^h}{|q|+\rho}\mathbf{v}^\star - \frac{g}{|q|+\rho} - \sum_{j\neq i}\frac{q_{ij}^h}{|q|+\rho}\mathbf{v}^\tau\right| \\
&= \max_{i\in\mathcal{X}^h}\left|\sum_{j\neq i}\frac{q_{ij}^h}{|q|+\rho}(\mathbf{v}^\star - \mathbf{v}^\tau)\right| \\
&= \max_{i\in\mathcal{X}^h}\left|\frac{|q|}{|q|+\rho}\right|\|v_h^\star - v_h^\tau\|_\infty \\
&= \alpha_h\|v_h^\star - v_h^\tau\|_\infty.
\end{aligned}$$

Therefore, in this particular instance of an $\mathcal{MDP}(N, L)$, the number of iterations is exactly the one given by Lemma 3.1.

3.4. Convergence rate and complexity for multiscale Markov decision processes. The main motivation for stating Theorem 3.2 is that it will enable us to make precise statements concerning the computational advantages of the coarse model derived in section 2.2. Using the results derived above, we show that *the principal benefit of the coarse model is not that the number of states is less, but that the rate of convergence is much higher* (provided that scale separation is present). In fact, the complexity of the coarse model when no scale separation is present, i.e., $\epsilon \approx 1$, is greater than that of the original model. The lemma below shows that $\mathcal{MMDP}(\epsilon, n, m, L)$ becomes ill conditioned as ϵ approaches zero. Note that there is no standard definition for an ill conditioned MDP. However, in the context of this paper, an MDP is ill conditioned if the contraction modulus of value iteration is approximately equal to one. The lemma below shows that this indeed is the case if the MDP is singularly perturbed.

LEMMA 3.4. *For $\mathcal{MMDP}(\epsilon, n, m, L)$ with Lipschitz constant α_h,*

$$\alpha_h \to 1 \quad as \quad \epsilon \to 0.$$

Proof. In $\mathcal{MMDP}(\epsilon, n, m, L)$, the Lipschitz constant has the form

$$\alpha_h = \max_{i \in \mathcal{X}^h, a \in \mathcal{A}_i^h} \frac{\left|\frac{1}{\epsilon}\hat{q}_{ii}(a) + w_{ii}(a)\right|}{\left|\frac{1}{\epsilon}\hat{q}_{ii}(a) + w_{ii}(a)\right| + \rho} \to 1 \quad \text{as} \quad \epsilon \to 0. \qquad \square$$

When ϵ is small, the guaranteed improvement in each iteration is almost zero for the fine model. On the other hand, for the coarse model the corresponding Lipschitz constant is given by

$$\alpha_H = \max_{i \in \mathcal{X}^H, a_H \in \mathcal{A}_i^H} \frac{\left|q_{ii}^H(a_H)\right|}{\left|q_{ii}^H(a_H)\right| + \rho}.$$

Crucially, α_H is independent of the multiscale structure of the original model. Therefore, there exists a $\epsilon^\star$ such that

$$\alpha_H \le \alpha_h \quad \text{for} \quad \epsilon \le \epsilon^\star.$$

In other words, the guaranteed convergence behavior of coarse model is superior to that of the fine model when ϵ is small enough. We end this section by comparing the computational complexity associated with the two models,

(3.15)

$$\text{Fine Model, } \mathcal{MMDP}(\epsilon, n, m, L) : O\left(\max\left\{\frac{\log\left(\frac{1}{(1-\alpha_h)\delta}\right)}{|\log \alpha_h|}, 0\right\}(mn)^2 L\right),$$

$$\text{Coarse Model, } \mathcal{MDP}(m, L^n) : O\left(\max\left\{\frac{\log\left(\frac{1}{(1-\alpha_H)\delta}\right)}{|\log \alpha_H|}, 0\right\} m^2 L^n\right).$$

If $\epsilon \approx 1$, then there are no benefits to aggregating the model using the singular perturbation approach. Indeed, as the preceding equation shows, the coarse model has an exponential dependence on n that is not present in the original model. We will discuss ways to alleviate this issue in section 6. Finally, in the setting of this paper, as $\epsilon \to 0$, the complexity of the fine model goes to infinity.

4. Analysis of the full approximation scheme. The conventional way to exploit multiresolution structure of a model is the FAS (see, e.g., [9]). The FAS is an extension of the multigrid scheme to nonlinear problems. Algorithms based on multigrid are in spirit close to the scheme we propose in this paper. In other words, multigrid algorithms try to solve the fine model by considering a hierarchy of approximate models. We also develop a scheme that fits within this general principle, but we propose a different way to couple the models together than the one used in FAS. We stress that the theory around the FAS is still valid and that the convergence proof developed in [9] can be used to show that the FAS will converge to the solution of $\mathcal{MMDP}(\epsilon, n, m, L)$. However, we will use a simple numerical example to illustrate the point that even though convergence is guaranteed, the rate of convergence is likely to be worse than just solving the fine model with the single level value iteration algorithm. In section 5, we show how to overcome this problem of the FAS by proposing a different way of incorporating information from the coarse model to the iterations of the fine model. We refer the interested reader to the tutorial in [3] for an introduction to multigrid and the FAS. The FAS is rigorously developed in [9]. In this section, we just mention some of the key ideas behind multigrid and FAS in order to understand how the existing framework is likely to fail for multiscale MDPs.

4.1. Prolongation and restriction. The first step in the development of the FAS is the definition of the prolongation and restriction operators. The prolongation operator is used to transfer solutions from the coarse model to the fine model. We use I_H^h and I_h^H to denote the prolongation and restriction operators, respectively. Typically, they are linear operators, and in this paper we take I_H^h and I_h^H to be constant matrices. The exact definition of these operators is problem dependant. For the class of models we consider in this paper, it is natural to define I_H^h and I_h^H based on the asymptotic properties of the fine and coarse models. The prolongation operator is given by

$$I_H^h = \text{diag}(\underbrace{\mathbb{1}_{n\times 1}, \mathbb{1}_{n\times 1}, \dots, \mathbb{1}_{n\times 1}}_{\text{m copies}}) \in \mathbb{R}^{nm\times m}.$$

The choice of I_H^h is based on (3.10), which shows that the value functions are asymptotically the same for the states that are in the same block. The definition of the restriction operator is not as straightforward as that of I_H^h. There is no obvious property to approximate v_H by v_h. However, a natural choice that can be rigorously justified (see section 5) is to restrict v_h into the same size as v_H using the stationary distribution of each block. Let φ_i denote the column vector for the stationary distributions associated with block i. We define the restriction operator as follows:

$$I_h^H = \text{diag}(\varphi_1^T, \varphi_2^T, \dots, \varphi_m^T) \in \mathbb{R}^{m\times nm}. \tag{4.1}$$

In other words, we "compress" the values of the value function in block i by forming a convex combination with the elements in each of the blocks. Notice that $Q_h^\epsilon(u_h)$ depends on the policy u_h and so there exist many different stationary distributions for each block. To address this problem, we select $Q_h^\epsilon(\tilde{u}_h)$ with $\tilde{u}_h$ as the best policy for the current incumbent solution at iteration τ, v_h^τ. That is, we select $\tilde{u}$ such that

$$\tilde{u}_h(i) \in \arg\min_{a\in\mathcal{A}_i^h} \left[\frac{G^h(i,a)}{|q_{ii}^h(a)| + \rho} + \sum_{j\neq i} \frac{q_{ij}^h(a)}{|q_{ii}^h(a)| + \rho} v_h^\tau(j) \right]$$

for the current solution v_h^τ, and apply (4.1) with $\varphi_1, \varphi_2, \dots, \varphi_m$ as the stationary distributions of blocks $1, 2, \dots, m$ in $Q_h^\epsilon(\tilde{u}_h)$. It follows from (3.10) that there exists a constant K such that

$$\|v_h^\star - I_H^h v_H^\star\|_\infty \le K\epsilon. \tag{4.2}$$

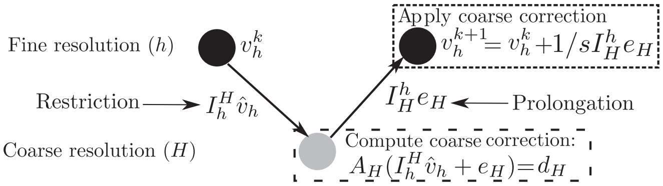


Fig. 1. *The FAS.*

4.2. The FAS algorithm. With the definitions of I_h^H and I_H^h provided above, we are now in a position to fully specify the FAS. The main idea of the FAS is simple, and we describe it as a solution algorithm for the following general nonlinear equation:

$$A_h(v_h) = f_h. \tag{4.3}$$

In our case, A_h is given in (2.7) and f_h can be taken to be zero (at the finest level). Given an incumbent solution $\hat{v}_h$, we can proceed to compute the exact correction for $\hat{v}_h$ so that it solves (4.3). That is, we compute an $e_h^\star$ such that

$$A_h(\hat{v}_h + e_h^\star) = f_h.$$

Of course the preceding nonlinear equation is just as hard as the original problem. The idea behind the FAS is instead of computing $e_h^\star$ using the fine model, an approximation of $e_h^\star$ is computed using the coarse model by solving the correction problem

$$A_H(I_h^H \hat{v}_h + e_H) = d_H,$$

where

$$d_H \triangleq A_H(I_h^H \hat{v}_h) - sI_h^H(A_h(\hat{v}_h) - f_h)$$

for some stepsize s. The existences of s and d_H establishes a useful relation between the fine and coarse models [9]. Finally, we complete the correction $\hat{v}_h + 1/sI_H^h e_H$. Figure 1 illustrates the main steps of FAS. We first compute $\hat{v}_h$ by τ applications of T_h. We then restrict the solution to the coarse scale and perform some iterations in the coarse scale to obtain an approximate correction e_H. We then prolongate the error correction term to the fine model and continue performing iterations at the fine scale. The addition of the prolongated error $1/sI_H^h e_H$ to the current solution $\hat{v}_h$ can lead to faster convergence rates than just using the fine model. The stepsize s is needed because this is a nonlinear problem. Obviously, it is possible to have more than one level. The full details of the algorithm are given in [9].

4.3. Numerical example from a multiscale manufacturing system. The full approximation scheme appears to be a good method to solve multiscale MDPs. It solved both the problems we set out to address in this paper, i.e., it uses the coarse model that is better conditioned, but still computes an exact solution for the original model. However, we will show using a simple example that for MDPs with multiscale structure the FAS can have an extremely slow convergence rate. The algorithm still converges, but it is much slower than simple value iteration. We propose a solution to this issue in the next section.

The example we use is not a contrived model but a simple and widely used model motivated by a manufacturing application. The model is described in [18] and concerns the control of a manufacturing process with two machines. Each machine has two states, up and down. We use 1 to denote that machine is working and 0 for the state when the machine is broken down. The total number of states in the system is $\{(1,1),(0,1),(1,0),(0,0)\}$, where (i,j) represents the state where machine 1 in state i and machine 2 in state j. In this manufacturing process, the state of each machine depends on the action a, which is the rate of preventive maintenance. The overall goal of the problem is to pick the policy u such that the machines do not break down often while the cost of maintenance is not too high. The model further assumes that the two machines have failure rates that occur in different timescales. To reflect this assumption, the following generator is used:

$$Q_h^{\epsilon}(a) = \frac{1}{\epsilon}\begin{bmatrix} -\lambda_1(a) & \lambda_1(a) & 0 & 0 \\ \mu_1(a) & -\mu_1(a) & 0 & 0 \\ 0 & 0 & -\lambda_1(a) & \lambda_1(a) \\ 0 & 0 & \mu_1(a) & -\mu_1(a) \end{bmatrix} + \begin{bmatrix} -\lambda_2(a) & 0 & \lambda_2(a) & 0 \\ 0 & -\lambda_2(a) & 0 & \lambda_2(a) \\ \mu_2(a) & 0 & -\mu_2(a) & 0 \\ 0 & \mu_2(a) & 0 & -\mu_2(a) \end{bmatrix}, \tag{4.4}$$

where $\lambda_1(a)/\epsilon$ and $\mu_1(a)/\epsilon$ are the breakdown and repair rates for machine 1 and $\lambda_2(a)$ and $\mu_2(a)$ for machine 2, respectively. As we can see, (4.5) is in the same form of (2.3). Intuitively, the more preventive maintenance is performed on a machine, then the machine is more likely to stay in state 1. For this simple example, we assume $\mathcal{X}^h = \{1,2,3,4\}$, $a \in \{1,2,\ldots,5\}$, and

$$\lambda_1(a) = 1/a,\ \mu_1(a) = a^2,$$
$$\lambda_2(a) = 3/a,\ \mu_2(a) = 3a.$$

A higher value of a would ensure the system is online more often. Of course, the more maintenance is performed, the higher the costs. To reflect this trade-off we use the following objective function:

$$G^h(x,a) = x^2 + a^2 \qquad \forall x \in \mathcal{X}^h \quad , \quad a \in \{1,2,3,4,5\}.$$

We used the FAS scheme to solve the infinite horizon version of the model described above. We plot the iteration history of the FAS against the exact solution of the fine model in Figure 2. The exact solution was obtained using linear programming. It may initially appear the FAS has a similar performance as the value iteration algorithm when applied to the fine model. On closer inspection, this is not the case. To illustrate this point we zoom in to the part of the computation where the FAS jumps to the coarse model (iteration 5000 in this example). From Figure 2 we see that actually no useful computation is performed during the coarse iterations. We point out that we tried different strategies for updating the stepsize as well as experimenting with the different parameters (such as when to jump to the coarse model and how many iterations to perform there). The numerical performance of the FAS is disappointing. It appears that the correction does not help the incumbent solution to get closer to the exact solution. In the next section, we discuss some possible reasons why the FAS may not be suitable for solving MDPs.

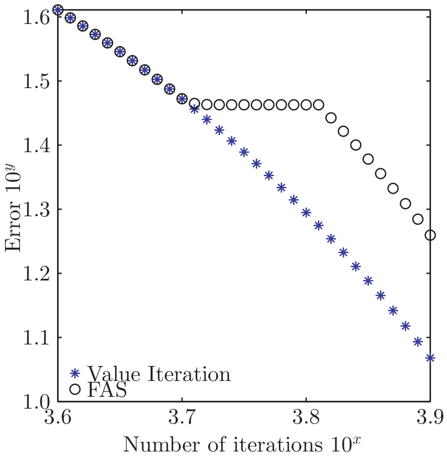


FIG. 2. *Performance of the FAS, $\epsilon = 10^{-2}$, initial number of iterations in the fine model:* 5000*, stepsize $s = 1$. The figure shows that no useful computation is performed by the FAS during the coarse iterations.*

4.4. Lack of progress in the coarse iterations of the FAS. In this section, we provide some possible explanations as to why the coarse iterations of the FAS do not provide useful corrections to the current fine solution. To simplify the analysis, suppose that there is only one policy u_h. We drop the dependence on u_h from Q_h^ϵ and $G^h(\cdot)$. It is easy to generalize our conclusions to the case when the policy space is richer. With these simplifications, our model reduces to the linear equation

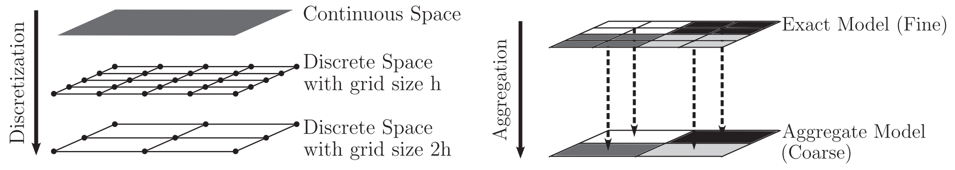
$$(A_h v_h)(i) \triangleq \frac{G^h(i)}{|q_{ii}^h| + \rho} + \sum_{j \neq i} \frac{q_{ij}^h}{|q_{ii}^h| + \rho} v_h(j) - v_h(i) = 0,$$

which can be written more compactly as $\mathcal{L}_h v_h = b_h$, where

$$\mathcal{L}_h = \begin{bmatrix} -1 & \dfrac{q_{12}^h}{|q_{11}^h| + \rho} & \cdots & \dfrac{q_{1nm}^h}{|q_{11}^h| + \rho} \\ \dfrac{q_{21}^h}{|q_{22}^h| + \rho} & -1 & \cdots & \dfrac{q_{2nm}^h}{|q_{22}^h| + \rho} \\ \vdots & \ddots & \ddots & \vdots \\ \dfrac{q_{nm1}^h}{|q_{nmnm}^h| + \rho} & \cdots & \cdots & -1 \end{bmatrix}, \ b_h = \begin{bmatrix} -\dfrac{G^h(1)}{|q_{11}^h| + \rho} \\ -\dfrac{G^h(2)}{|q_{22}^h| + \rho} \\ \vdots \\ -\dfrac{G^h(nm)}{|q_{nmnm}^h| + \rho} \end{bmatrix}.$$

The corresponding coarse model also reduces to the linear system $\mathcal{L}_H v_H = b_H$, where

$$\mathcal{L}_H = \begin{bmatrix} -1 & \dfrac{q_{12}^H}{|q_{11}^H| + \rho} & \cdots & \dfrac{q_{1m}^H}{|q_{11}^H| + \rho} \\ \dfrac{q_{21}^H}{|q_{22}^H| + \rho} & -1 & \cdots & \dfrac{q_{2m}^H}{|q_{22}^H| + \rho} \\ \vdots & \ddots & \ddots & \vdots \\ \dfrac{q_{m1}^H}{|q_{mm}^H| + \rho} & \cdots & \cdots & -1 \end{bmatrix}, \ b_H = \begin{bmatrix} -\dfrac{G^h(1)}{|q_{11}^h| + \rho} \\ -\dfrac{G^h(2)}{|q_{22}^h| + \rho} \\ \vdots \\ -\dfrac{G^h(m)}{|q_{mm}^h| + \rho} \end{bmatrix}.$$



(a) Traditional Discretization in Differential Equations

(b) Aggregation for MMDP

FIG. 3. *Different ideas between discretization and aggregation.*

Given an incumbent solution v_h and the exact correction $e_h^\star$, we have

$$\mathcal{L}_h(v_h + e_h^\star) = b_h,$$

from which we obtain the following:

$$e_h^\star = \mathcal{L}_h^{-1} b_h - v_h = \mathcal{L}_h^{-1} b_h - \mathcal{L}_h^{-1}\mathcal{L}_h v_h = \mathcal{L}_h^{-1}(b_h - \mathcal{L}_h v_h).$$

Letting $d_h = \mathcal{L}_h v_h - b_h$, we obtain

$$e_h^\star = \mathcal{L}_h^{-1}(b_h - \mathcal{L}_h v_h) = -\mathcal{L}_h^{-1} d_h.$$

The FAS approximates $e_h^\star$ by computing a correction in the coarse model. For the correction problem, we let $v_H \triangleq I_h^H v_h$, $d_H \triangleq \mathcal{L}_H v_H - s I_h^H d_h$ and compute

$$\tilde{v}_H \triangleq \mathcal{L}_H^{-1} d_H = v_H - s\mathcal{L}_H^{-1} I_h^H d_h.$$

Then, the correction is

$$\tilde{e}_h \triangleq \frac{1}{s} I_H^h(\tilde{v}_H - v_H) = \frac{1}{s} I_H^h(-s\mathcal{L}_H^{-1} I_h^H d_h) = -I_H^h \mathcal{L}_H^{-1} I_h^H d_h.$$

In the case when $I_H^h \mathcal{L}_H^{-1} I_h^H \approx \mathcal{L}_h^{-1}$, the correction problem provides a good approximation of $e_h^\star$. Traditionally, multigrid methods are aimed toward the solution of differential equations and discretize a continuous space into different grid sizes. The assumption that $I_H^h \mathcal{L}_H^{-1} I_h^H \approx \mathcal{L}_h^{-1}$ usually holds because $\mathcal{L}_H$ and $\mathcal{L}_h$ are discrete operators derived from the same continuous operator. However, in the case of MMDP, our coarse model is obtained by averaging each block with its stationary distribution, which makes $\mathcal{L}_H$ different than $\mathcal{L}_h$. Also, as $\epsilon \to 0$, $\mathcal{L}_H$ remains unchanged, but this is not the case for $\mathcal{L}_h$. Figure 3 illustrates the differences between the two kinds of problems.

In order to give some deeper insights into the numerical challenges caused by $\mathcal{MMDP}$ models, we consider the example from the previous subsection when we have a single action, $a = 1$. In this simple setting we can compute $\mathcal{L}_h$ and $\mathcal{L}_H$ exactly and see the differences between the two operators. We performed this analysis with the parameters described in the previous section and found that the difference between $I_H^h \mathcal{L}_H^{-1} I_h^H$ and $\mathcal{L}_h^{-1}$ is very large, especially for smaller ϵ. The difference between the two operators was measured using the spectral norm. We also computed the eigenvalues of $\mathcal{L}_h^{-1}$ in closed form. The resulting expression are long but can be easily computed using a symbolic mathematics package. From our calculations we observed that as ϵ approaches zero, the matrix $\mathcal{L}_h$ becomes nearly singular and therefore its inverse does not exist. In contrast, $\mathcal{L}_H$ is independent of ϵ and its inverse always exists.

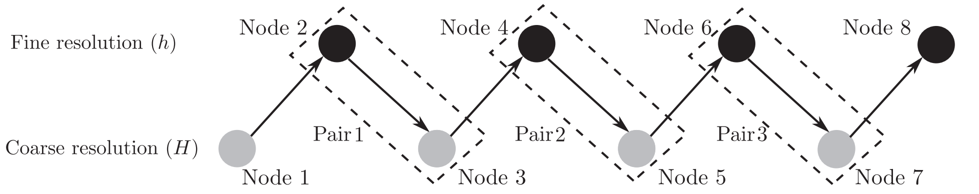


Fig. 4. *The AMS.*

This explains why the difference between $I_H^h \mathcal{L}_H^{-1} I_h^H$ and $\mathcal{L}_h^{-1}$ is very large for small ϵ. In addition, we found that this difference, when measured using the spectral norm, is a log-linear function of ϵ. This indicates that FAS is not suitable for our problem because the basic motivation of FAS does not fit with the setup of $\mathcal{MMDP}$ models. In the next section, we will introduce a new scheme, the alternating multiresolution scheme that attempts to address some of these issues.

5. The alternating multiresolution scheme. We have already seen in section 4 that the traditional full approximation scheme is not suitable for $\mathcal{MMDP}$. We introduce a new algorithm, the AMS, to address the low convergence rate of the FAS. One can think of the AMS as a modified version FAS. In particular, we eliminate the correction problem in the coarse model and replace it with the original coarse problem. The main idea of the AMS is to split all the iterations in the coarse model into many pieces.

Algorithm 1. The P-AMS.

```
• Start with initial guess $v_H^0$.
• $v_H^1 \leftarrow T_H^{\tau_{H,P}} v_H^0$.
• $v_h^0 \leftarrow I_H^h v_H^1$.
for $p = 1, 2, \dots, P$ do
    • $v_h^1 \leftarrow T_h^{\tau_{h,P}} v_h^0$.
    • $v_H^1 \leftarrow I_h^H v_h^1$.
    • $v_H^2 \leftarrow T_H^{\tau_{H,P}} v_H^1$.
    • $v_h^0 \leftarrow v_h^1 + s I_H^h \left[ (v_H^2 - v_H^1) \right]$ (where $s$ is the stepsize).
end for
while $A_h(v_h^0) \geq \delta$ do
    • $v_h^0 \leftarrow T_h v_h^0$.
end while
```

In the AMS, neither the coarse model nor the fine model are solved completely once. Instead, we apply the coarse contraction map $\tau_{H,P}$ times to the initial guess of the coarse model then project the solution to the fine model as an initial guess. We then apply the fine contraction map T_h for $\tau_{h,P}$ times, project it back to the coarse model for $\tau_{H,P}$ iterations to find the approximate error, and so on. The scheme is shown in Figure 4. For convenience we number the nodes and pair up one fine iteration node with one coarse iteration node together. We start with node 1, which is a coarse iteration node, and we add 1 to the iteration counter whenever we switch between coarse and fine iterations. With this indexing convention, all the odd nodes are iterations with the coarse model, and all the even nodes are iterations with the fine model. For an alternating multiresolution scheme with M nodes, we pair up node $2j$ and node $2j + 1$ together, $j = 1, 2, \dots, (M - 2)/2$. A P-AMS denotes the alternating multiresolution scheme with P pairs. Figure 4 illustrates the P-AMS with P = 3. We state the full algorithm below.

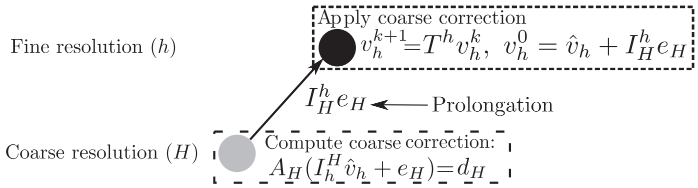

Apply coarse correction
Fine resolution (h)
$v_h^{k+1}=T^h v_h^k,\ v_h^0=\hat{v}_h+I_H^h e_H$
$I_H^h e_H$
Prolongation
Coarse resolution (H)
Compute coarse correction:
$A_H(I_h^H\hat{v}_h+e_H)=d_H$

FIG. 5. *The one-way multiresolution scheme (OWMS).*

In order to have a fully specified algorithm, we need to decide $\tau_{h,P}$, $\tau_{H,P}$, and a strategy to pick the stepsize parameter s. We discuss how the number of iterations is determined below. We first discuss this issue on a simplified version of AMS before addressing the general case.

5.1. One-way multiresolution scheme. We begin our analysis of the AMS for the specific case where we only have two nodes. We call this specific scheme the one-way multiresolution scheme (OWMS). In this scheme, we solve the coarse model first and prolongate the solution as an initial guess for the fine model. Figure 5 illustrates the simplified scheme. The lemma below gives an upper bound on the number of iterations that need to be performed in the coarse model. The significance of the lemma below is that it provides an estimate of the number of iterations required and relies on known input data.

LEMMA 5.1. *The number of iterations required to achieve the following accuracy in the coarse model,*

$$\left\|v_H^{\tau_{H,0}}-v_H^{\tau_{H,0}-1}\right\|_\infty \le K\frac{\epsilon(1-\alpha_H)}{\alpha_H}, \tag{5.1}$$

is bounded by

$$\tau_{H,0}\le\frac{\log\left(\dfrac{1}{\epsilon(1-\alpha_H)}\right)}{|\log\alpha_H|}. \tag{5.2}$$

In addition, when (5.1) is satisfied, $v_H^{\tau_{H,0}}$ satisfies

$$\|v_H^\star-v_H^{\tau_{H,0}}\|_\infty\le K\epsilon, \tag{5.3}$$

where the constant K was defined in (3.10).

Proof. Using the contraction property (3.5),

$$\|v_H^\star-v_H^{\tau_{H,0}}\|_\infty\le\frac{\alpha_H}{1-\alpha_H}\|v_H^{\tau_{H,0}}-v_H^{\tau_{H,0}-1}\|_\infty\le\frac{\alpha_H}{1-\alpha_H}K\frac{\epsilon(1-\alpha_H)}{\alpha_H}\le K\epsilon.$$

In order to find the bound of $\tau_{H,0}$, we select $\tau'_{H,0}$ such that

$$\|v_H^{\tau'_{H,0}}-v_H^{\tau'_{H,0}-1}\|_\infty\le\alpha_H^{\tau'_{H,0}-1}\|v_H^1-v_H^0\|_\infty\le\alpha_H^{\tau'_{H,0}-1}K=K\frac{\epsilon(1-\alpha_H)}{\alpha_H},$$

and so

$$\alpha_H^{\tau'_{H,0}-1}K=K\frac{\epsilon(1-\alpha_H)}{\alpha_H}, \tag{5.4}$$

$$\tau'_{H,0}=\frac{\log\left(\dfrac{1}{\epsilon(1-\alpha_H)}\right)}{|\log\alpha_H|}. \tag{5.5}$$

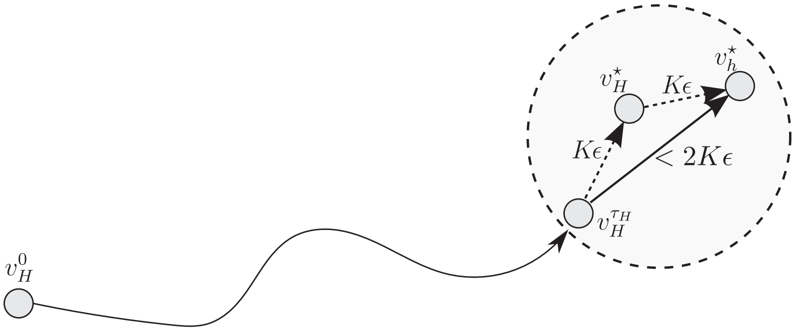


FIG. 6. *Stopping criteria in the coarse model.*

Notice that $\tau'_{H,0} > 0$ because both $\epsilon < 1$ and $\alpha_H < 1$. Since $v_H^{\tau'_{H,0}}$ guarantees the desired accuracy, we have $\tau_{H,0} \le \tau'_{H,0}$, for $\tau'_{H,0}$ in (5.5). □

Figure 6 illustrates the concept behind the stopping criterion developed in the preceding lemma. The reason we do not compute the exact $v_H^\star$ is that using $v_H^{\tau_{H,0}}$ as the initial point for the fine iterations (bold line) is always faster than the alternative (dotted line) of computing the fine solution $v_H^\star$ and then performing fine iterations to compute $v_h^\star$. The preceding lemma just gives a rigorous backing to the intuitive idea that the exact solution of the approximate model does not add enough information to justify the cost of computing it.

The next step in the definition of the OWMS is the definition of $\tau_{h,0}$, i.e., the number of iterations that need to be performed in the fine scale. Of course, this number must depend on the a user specified error tolerance δ defined as follows:

$$\|v_h^\star - v_h^{\tau_{h,0}}\|_\infty < \delta.$$

Any solution that satisfies the solution above is called δ-optimal. (Note that under our assumptions $v_h^\star$ is unique.) The upper bound derived in the lemma below depends on δ and the amount of scale separation present in the problem ϵ.

LEMMA 5.2. *Suppose that the initial point for the fine iterations is*

$$v_h^0 \triangleq I_H^h v_H^{\tau'_{H,0}},$$

where $\tau'_{H,0}$ satisfies (5.1). *Then the number of iterations in the fine model required to compute a δ-optimal solution is bounded by*

$$\tau_{h,0} \le \max\left\{\frac{\log\left(\dfrac{K\epsilon(2-\alpha_H)}{\delta}\right)}{|\log\alpha_h|}, 0\right\}. \tag{5.6}$$

Proof. Since the initial guess of the fine model is the solution in the coarse model, using Lemma 5.1, we have

$$\begin{aligned}\|v_h^0 - v_h^\star\|_\infty &\le \|v_h^0 - I_H^h v_H^\star\|_\infty + \|I_H^h v_H^\star - v_h^\star\|_\infty,\\ &\le \alpha_H^{\tau'_{H,0}}\|v_H^0 - v_H^\star\|_\infty + K\epsilon,\\ &\le \alpha_H^{\tau'_{H,0}}K + K\epsilon,\\ &= \epsilon(1-\alpha_H)K + K\epsilon,\\ &= K\epsilon(2-\alpha_H).\end{aligned}$$

Hence, we select a parameter $\tau'_{h,0}$ such that

$$\|v_h^{\tau'_{h,0}} - v_h^\star\|_\infty \le \alpha_h^{\tau'_{h,0}} \|v_h^0 - v_h^\star\|_\infty \le \alpha_h^{\tau'_{h,0}} K\epsilon(2-\alpha_H) = \delta,$$

and so

$$\tau'_{h,0} = \frac{\log\left(\dfrac{K\epsilon(2-\alpha_H)}{\delta}\right)}{|\log \alpha_h|},$$

as required. □

It follows from the lemma above that the number of iterations in the fine model decrease as ϵ decreases. This is because when ϵ decreases, the coarse model is a better approximation of the fine model and so we require fewer iterations in the fine model. This is in stark contrast to the classical single scale value iteration algorithm that requires more iterations as ϵ decreases. As $\epsilon \to 0$, the contraction modulus of value iteration α_h will tend to 1. In practice this means that if there is sufficient scale separation in the model, value iteration will be extremely slow. According to Lemma 3.1, the number of iterations required will tend to infinity as $\alpha_h \to 1$. However, when $\epsilon \to 0$, the prolongated value function $I_H^h v_H$ will equal tend to v_h, and so only using the coarse model will be enough to get an accurate solution.

For OWMS and when $\epsilon \to 0$, it follows from Lemma 5.2 that the upper bound of the number of iterations needed in the fine model tends to zero, i.e., no iterations are needed in the fine model. This is very good news from the point of computation because, as alluded to above, when ϵ is small the algorithm may require arbitrarily many iterations using the full model to converge. At the regime of $\epsilon \to 0$, the coarse model can replace the fine model completely, and OWMS can detect that ϵ is small enough and not perform any expensive iterations using the fine model.

Note that ϵ is not an input parameter for our algorithm but represents the scale difference in the problem. On the other hand, δ is a user specified parameter and it represents the accuracy of the final solution. Therefore, the two parameters are independent of each other. For larger δ, the final solution could be less accurate, and so it requires less iterations in the fine model. This again can be seen from Lemma 5.2. While ϵ and δ are not directly related, they could have similar effects, for example, a small δ (high accuracy), and for a fixed ϵ will mean that more iterations using the fine model will be performed by both value iteration and OWMS. Similarly, for a fixed δ, a small ϵ would imply more iterations for the classical value iteration algorithm, but the situation for OWMS is more complicated. For example, if $\epsilon \ll \delta$, then OWMS will make no iterations with the fine model. All these relationships can be derived from the expressions derived in Lemma 5.2 and depend on parameters that are known (up to a multiplicative constant) by the user.

Of course, comparing just the number of iterations in the fine model is not sufficient. In order to perform a more rigorous and fair comparison between the newly proposed scheme OWMS and the classical single scale value iteration, we derive the complexity of OWMS. We then find the conditions the MDP has to satisfy in order for the OWMS to have a more favorable complexity than value iteration.

THEOREM 5.3. *For $\mathcal{MMDP}(\epsilon, n, m, L)$, the complexity of the OWMS is*

$$O\left(\frac{\log\left(\dfrac{1}{\epsilon(1-\alpha_H)}\right)}{|\log \alpha_H|} m^2 L^n + \max\left\{\frac{\log\left(\dfrac{\epsilon(2-\alpha_H)}{\delta}\right)}{|\log \alpha_h|}, 0\right\}(nm)^2 L\right). \tag{5.7}$$

Proof. The complexity of the algorithm is divided into two parts. The first part is the computational complexity associated with the coarse model. The second part is the complexity associated with fine iterations. Combining the information obtained by the Lemmas 5.1 and 5.2, we obtain the required result. □

We are now in a position to derive conditions that the MDP needs to satisfy so that we can guarantee that the proposed scheme will outperform value iteration.

THEOREM 5.4. *Suppose that the tolerance $\delta < \min\{\epsilon(2-\alpha_H), 1\}$. For $\mathcal{MMDP}$ (ϵ, n, m, L), the complexity of the OWMS is less than the complexity of the value-iteration if*

$$n^2 \geq \frac{\log(\epsilon(1-\alpha_H))}{\log(\epsilon(2-\alpha_H)(1-\alpha_h))} \frac{|\log \alpha_h|}{|\log \alpha_H|} L^{n-1}. \tag{5.8}$$

Proof. From (5.7) and (3.15), we know the complexity of both algorithms. Also, since $\delta < \min\{\epsilon(2-\alpha_H), 1\}$,

$$\log\left(\frac{\epsilon(2-\alpha_H)}{\delta}\right) > 0 \quad \text{and} \quad \log\left(\frac{1}{(1-\alpha_h)\delta}\right) > 0.$$

We proceed by computing the difference between the complexities:

$$\frac{\log\left(\dfrac{1}{(1-\alpha_h)\delta}\right)}{|\log \alpha_h|}(nm)^2 L - \frac{\log\left(\dfrac{1}{\epsilon(1-\alpha_H)}\right)}{|\log \alpha_H|} m^2 L^n - \frac{\log\left(\dfrac{\epsilon(2-\alpha_H)}{\delta}\right)}{|\log \alpha_h|}(nm)^2 L$$

$$= \frac{m^2}{L}\left[\log\left(\frac{1}{\epsilon(1-\alpha_h)(2-\alpha_H)}\right)\frac{n^2}{|\log \alpha_h|} - \log\left(\frac{1}{\epsilon(1-\alpha_H)}\right)\frac{L^{n-1}}{|\log \alpha_H|}\right].$$

Using inequality (5.8) implies that the difference is greater than

$$\begin{aligned} &\geq \frac{m^2}{L}\left[\log\left(\frac{1}{\epsilon(1-\alpha_H)}\right)\frac{L^{n-1}}{|\log \alpha_H|} - \log\left(\frac{1}{\epsilon(1-\alpha_H)}\right)\frac{L^{n-1}}{|\log \alpha_H|}\right]. \\ &\geq 0. \end{aligned}$$

□

As we can see, the complexity of the OWMS is not always less than the single resolution algorithm. This is due to the fact that the number of actions for each coarse state is an exponential compared to the number of actions for each single state in the fine model. We will return to this issue in section 6. If the problem has sufficient scale separation (which is the setting of this paper), then we see that because

$$\frac{\log(\epsilon(1-\alpha_H))}{\log(\epsilon(1-\alpha_h)(2-\alpha_H))} \frac{|\log \alpha_h|}{|\log \alpha_H|} \to 0 \quad \text{as} \quad \epsilon \to 0,$$

inequality (5.8) is asymptotically satisfied in ϵ. Therefore, for a small enough ϵ, the complexity of the OWMS is less than that of value iteration.

5.2. Convergence analysis of AMS. In this section we turn our attention to the full AMS which includes the OWMS as a special case. We first prove some technical lemmas that will be used later on.

LEMMA 5.5. *$I_h^H I_H^h = I_m$, where I_m is the identity matrix in $\mathbb{R}^{m\times m}$.*

Proof. By definition, $I_H^h = \text{diag}(\underbrace{\mathbb{1}_{n\times 1}, \mathbb{1}_{n\times 1}, \dots, \mathbb{1}_{n\times 1}}_{\text{m copies}})$, so for any $v = (v_i) \in \mathbb{R}^m$,

$$I_H^h v = \text{diag}(\underbrace{\mathbb{1}_{n\times 1}, \mathbb{1}_{n\times 1}, \dots, \mathbb{1}_{n\times 1}}_{\text{m copies}})v = \text{diag}(\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_m}),$$

where $\mathbf{v_i} = v_i \mathbb{1}_{n\times 1}$, for $i = 1, 2, \dots, m$. The equality above follows from the fact that the vector v is premultiplied with a block diagonal matrix. Using the definition of I_h^H, we obtain

$$I_h^H I_H^h v = \text{diag}(\varphi_1^T, \varphi_2^T, \dots, \varphi_m^T)\text{diag}(\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_m}).$$

Using the fact that the φ_i are the stationary distribution vectors,

$$\sum_{j=1}^{n} (\varphi_i)_j = 1 \quad \forall i \in \{1, 2, \dots, m\},$$

we obtain $\varphi_i^T \mathbf{v_i} = v_i$ and $I_h^H I_H^h v = v$, as claimed. ◻

LEMMA 5.6. *$\|I_H^h\|_\infty = \|I_h^H\|_\infty = 1$.*

Proof. I_H^h is a matrix with only one 1 in each of its rows, and so $\|I_H^h\|_\infty = 1$. On the other hand, I_h^H is a stochastic matrix, and so $\|I_h^H\|_\infty = 1$. ◻

LEMMA 5.7. *$\|I - I_H^h I_h^H\|_\infty \le 2$, where I is the identity matrix in $\mathbb{R}^{mn\times mn}$.*

Proof. Notice that I_h^H is a stochastic matrix, and

$$I_H^h I_h^H = \text{diag}(\underbrace{\mathbb{1}_{n\times 1}, \mathbb{1}_{n\times 1}, \dots, \mathbb{1}_{n\times 1}}_{\text{m copies}})\text{diag}(\varphi_1^T, \varphi_2^T, \dots, \varphi_m^T) = \text{diag}(\hat{\varphi}_1^T, \hat{\varphi}_2^T, \dots, \hat{\varphi}_m^T),$$

where $\hat{\varphi}_i^T \triangleq \mathbb{1}_{n\times 1}\varphi_i^T$ for $i = 1, 2, \dots, m$. In addition, $I_H^h I_h^H$ is also a stochastic matrix because each of $\hat{\varphi}_i^T$ is a stochastic matrix. Therefore, the sum of the absolute values for every row of $I - I_H^h I_h^H$ must be less than or equal to 2. ◻

Suppose we have a current solution v_h; we then restrict it to the coarse model for correction, and we then prolong the correction to the fine model and add it to v_h. We call this process a correction, and the associated *correction operator* is defined as

$$\mathcal{T}_\tau v_h \triangleq v_h + sI_H^h(T_H^\tau I_h^H v_h - I_h^H v_h). \tag{5.9}$$

If the correction $v_h' = \mathcal{T}_\tau v_h$ is useful for the problem, then the new solution v_h' should be closer to the optimal solution than the original v_h. In the next lemma, we provide a link between the corrected value function v_h' and the current incumbent v_h.

LEMMA 5.8. *For the current value v_h in the fine level, let $v_h' = \mathcal{T}_\tau v_h$, and then*

$$\|v_h^\star - v_h'\|_\infty \le [|1-s| + s\alpha_H^\tau]\, \|v_h^\star - v_h\|_\infty + (s\alpha_H^\tau + s)K\epsilon + 2Ks, \tag{5.10}$$

where $s \ge 0$ is the fixed stepsize.

Proof.

$$\begin{aligned}
\|v_h^\star - v_h'\|_\infty &= \|v_h^\star - v_h - sI_H^h(v_H^\tau - I_h^H v_h)\|_\infty \\
&= \|v_h^\star - v_h - sI_H^h v_H^\tau + sI_H^h I_h^H v_h\|_\infty \\
&\le |1-s|\|v_h^\star - v_h\|_\infty + s\|v_h^\star - I_H^h v_H^\tau\|_\infty \\
&\quad + s\|v_h - I_H^h I_h^H v_h\|_\infty \\
&\le |1-s|\|v_h^\star - v_h\|_\infty + s\|v_h^\star - I_H^h v_H^\star\|_\infty \\
&\quad + s\|I_H^h v_H^\star - I_H^h v_H^\tau\|_\infty + s\|I - I_H^h I_h^H\|_\infty\|v_h\|_\infty \\
&\le |1-s|\|v_h^\star - v_h\|_\infty + sK\epsilon + s\|I_H^h\|_\infty\|v_H^\star - v_H^\tau\|_\infty + 2Ks \\
&\le |1-s|\|v_h^\star - v_h\|_\infty + sK\epsilon + s\alpha_H^\tau\|v_H^\star - I_h^H v_h\|_\infty + 2Ks \\
&\le |1-s|\|v_h^\star - v_h\|_\infty + sK\epsilon + s\alpha_H^\tau\|v_H^\star - I_h^H v_h^\star\|_\infty \\
&\quad + s\alpha_H^\tau\|I_h^H v_h^\star - I_h^H v_h\|_\infty + 2Ks \\
&\le |1-s|\|v_h^\star - v_h\|_\infty + sK\epsilon + s\alpha_H^\tau\|I_h^H\|_\infty\|I_H^h v_H^\star - v_h^\star\|_\infty \\
&\quad + s\alpha_H^\tau\|v_h^\star - v_h\|_\infty + 2Ks \\
&\le (|1-s| + s\alpha_H^\tau)\|v_h^\star - v_h\|_\infty + (s + s\alpha_H^\tau)K\epsilon + 2Ks. \qquad \square
\end{aligned}$$

Lemma 5.8 provides a bound for the difference between the new solution v_h' and the optimal solution $v_h^\star$. Using the preceding result, we then derive the conditions required for the new error to be smaller than the previous error. Therefore, when these conditions are satisfied, Algorithm 1 is a contraction, and the convergence is guaranteed by the fixed point theorem.

THEOREM 5.9. *Algorithm* 1 *is guaranteed to be a contraction if*

$$\begin{aligned}
(1-\alpha_H^\tau)\|v_h^\star - v_h\|_\infty &\ge (\alpha_H^\tau + 1)K\epsilon + 2K \quad \textit{for} \quad 0 \le s \le 1, \\
\left[\frac{2}{s} - (1+\alpha_H^\tau)\right]\|v_h^\star - v_h\|_\infty &\ge (\alpha_H^\tau + 1)K\epsilon + 2K \quad \textit{for} \quad s > 1.
\end{aligned}$$

Proof. In the case $0 \le s \le 1$, using Lemma 5.8 we obtain

$$\begin{aligned}
\|v_h^\star - v_h'\|_\infty &\le [|1-s| + s\alpha_H^\tau]\,\|v_h^\star - v_h\|_\infty + (s\alpha_H^\tau + s)K\epsilon + 2Ks \\
&= \|v_h^\star - v_h\|_\infty + s\left([\alpha_H^\tau - 1]\,\|v_h^\star - v_h\|_\infty\right) + s\left((\alpha_H^\tau + 1)K\epsilon + 2K\right) \\
&\le \|v_h^\star - v_h\|_\infty - s\left((\alpha_H^\tau + 1)K\epsilon + 2K\right) + s\left((\alpha_H^\tau + 1)K\epsilon + 2K\right) \\
&\le \|v_h^\star - v_h\|_\infty.
\end{aligned}$$

For the case $1 < s$, we obtain

$$\begin{aligned}
\|v_h^\star - v_h'\|_\infty &\le [|1-s| + s\alpha_H^\tau]\,\|v_h^\star - v_h\|_\infty + (s\alpha_H^\tau + s)K\epsilon + 2Ks \\
&= \|v_h^\star - v_h\|_\infty + s\left(-\frac{2}{s} + (1+\alpha_H^\tau)\right)\|v_h^\star - v_h\|_\infty + s\left((\alpha_H^\tau + 1)K\epsilon + 2K\right) \\
&\le \|v_h^\star - v_h\|_\infty - s\left((\alpha_H^\tau + 1)\tilde{K}\epsilon + 2\right) + s\left((\alpha_H^\tau + 1)K\epsilon + 2K\right) \\
&\le \|v_h^\star - v_h\|_\infty. \qquad \square
\end{aligned}$$

We note from the conditions of Theorem 5.9 that $(\alpha_H^\tau + 1)K\epsilon + 2K$ is a constant through out all the iterations. However, $\|v_h^\star - v_h\|_\infty$ depends on the current solution v_h. This indicates the correction is not guaranteed to be useful when the current solution is too close to the exact solution $v_h^\star$. This suggests that we should restrict the number of coarse iterations of the algorithm. We should not perform coarse iterations when the current solution is close to $v_h^\star$. The result does not provide insight into how to select s since it depends on the optimal solution. In the result below, we provide the range that s can take so that the conditions of Theorem 5.9 are satisfied.

COROLLARY 5.10. *Algorithm* 1 *is guaranteed to be a contraction if the stepsize is chosen so that*

$$0 \leq s \leq \frac{2}{1+\alpha_H^\tau}.$$

Proof. Using the result from Theorem 5.9, we obtain

$$(\alpha_H^\tau + 1)K\epsilon + 2K \geq 0.$$

Algorithm 1 is guaranteed to be a contraction only if

$$1 - \alpha_H^\tau \geq 0 \quad \text{for} \quad 0 \leq s \leq 1, \tag{5.11}$$

$$\frac{2}{s} - (1+\alpha_H^\tau) \geq 0 \quad \text{for} \quad s > 1. \tag{5.12}$$

In the case when $s \leq 1$, (5.11) is always satisfied, and in addition the following holds:

$$|1-s| + s\alpha_H^\tau = 1 - s + s\alpha_H^\tau = 1 - s(1-\alpha_H^\tau) \leq 1.$$

However, when $s > 1$, (5.12) is only satisfied when

$$s \leq \frac{2}{1+\alpha_H^\tau}.$$

For $1 < s \leq \dfrac{2}{1+\alpha_H^\tau}$,

$$|1-s| + s\alpha_H^\tau = -1 + s + s\alpha_H^\tau = -1 + s(1+\alpha_H^\tau) \leq 1. \qquad \square$$

The above corollary shows that when $s > 2/(1+\alpha_H^\tau)$, it is not guaranteed that the correction would be useful. With the results obtained above concerning the correction iterations, we are now in a position to derive the complexity of the proposed scheme.

Let $v_{a,h}^{2i}$ denote the solution after calculations in node $2i$, and let $v_{a,H}^{2i-1}$ be the solution after calculations in node $2i-1$ for $i = 1, 2, \ldots, P+1$. Since the incumbent solution $v_{a,H}^{2i-1}$ is in the coarse level, we prolongate the solution of the coarse model to the fine model as follows:

$$v_a^j = \begin{cases} v_{a,h}^j & \text{if } j \text{ is even,} \\ I_H^h v_{a,H}^j & \text{if } j \text{ is odd} \end{cases} \quad \text{for } j = 1, 2, \ldots, 2(P+1).$$

With this notation, the flow of computations for the AMS is

$$v_a^0 \Rightarrow v_a^1 \Rightarrow v_a^2 \Rightarrow \cdots \Rightarrow v_a^{2P+1} \Rightarrow v_a^{2(P+1)},$$

where $v_a^{2(P+1)}$ is the final solution computed by the algorithm and therefore satisfies

$$\|v_h^\star - v_a^{2(P+1)}\|_\infty \leq \delta,$$

where δ is a user specified error tolerance. We assume that for P-AMS, we always perform $\tau_{H,P}$ iterations in each coarse iteration node and $\tau_{h,P}$ iterations in first P fine iteration node for constants P, $\tau_{h,P}$, and $\tau_{H,P}$. In the complexity analysis below, we fix these constants and calculate the number of iterations the algorithms needs to refine the final solution from the AMS.

Lemma 5.11. *Suppose that in the P-AMS, we perform $\tau_{H,P}$ iterations in node $1, 3, 5, \ldots, 2P+1$ and $\tau_{h,P}$ iterations in node $2, 4, 6, \ldots, 2P$, and with a constant stepsize s satisfying the conditions in Corollary* 5.10. *Then the initial point (v_a^{2P+1}) from which the computations in the final node $2P+2$ start satisfies*

$$\|v_h^\star - v_a^{2P+1}\|_\infty \le \eta^P K\epsilon + \eta^P \alpha_H^{\tau_{H,P}} K + \frac{1-\eta^{P+1}}{1-\eta} sC, \tag{5.13}$$

where $\eta \triangleq (|1-s| + s\alpha_H^{\tau_{H,P}})\alpha_h^{\tau_{h,P}}$ and $C \triangleq (1+\alpha_H^{\tau_{H,P}})K\epsilon + 2K$.

Proof. We will prove this using induction. Using the contraction mapping property,

$$\begin{aligned}\|v_h^\star - v_a^1\|_\infty &\le \|v_h^\star - I_H^h v_H^\star\|_\infty + \|I_H^h v_H^\star - v_a^1\|_\infty\\ &\le K\epsilon + \alpha_H^{\tau_{H,P}}\|v_H^\star - v_H^0\|_\infty\\ &= K\epsilon + \alpha_H^{\tau_{H,P}} K\end{aligned}$$

and

$$\|v_h^\star - v_a^2\|_\infty \le \alpha_h^{\tau_{h,P}}\|v_h^\star - v_a^1\|_\infty.$$

Using Lemma 5.8, we have

$$\|v_h^\star - v_a^3\|_\infty \le \left[|1-s| + s\alpha_H^{\tau_{H,P}}\right]\|v_h^\star - v_a^2\|_\infty + (s\alpha_H^{\tau_{H,P}} + s)K\epsilon + 2Ks,$$

which gives the following estimate:

$$\begin{aligned}\|v_h^\star - v_a^3\|_\infty &\le \left[|1-s| + s\alpha_H^{\tau_{H,P}}\right]\alpha_h^{\tau_{h,P}}\|v_h^\star - v_a^1\|_\infty + (s\alpha_H^{\tau_{H,P}} + s)K\epsilon + 2Ks\\ &= \eta\|v_h^\star - v_a^1\|_\infty + sC.\end{aligned}$$

By induction, after P pairs, we have

$$\begin{aligned}\|v_h^\star - v_a^{2P+1}\|_\infty &\le \eta^P\|v_h^\star - v_a^1\|_\infty + sC\sum_{k=0}^{P}\eta^k\\ &= \eta^P\|v_h^\star - v_a^1\|_\infty + \left(\frac{1-\eta^{P+1}}{1-\eta}\right)sC\\ &\le \eta^P K\epsilon + \eta^P\alpha_H^{\tau_{H,P}}K + \left(\frac{1-\eta^{P+1}}{1-\eta}\right)sC,\end{aligned}$$

as required. ∎

Lemma 5.11 establishes the error of the current solution before entering the last node. We now complete the analysis by deriving an upper bound for the number of iterations required in the fine model to get the final solution with tolerance δ.

Lemma 5.12. *Suppose that in the P pairs alternating multiresolution scheme, we perform $\tau_{H,P}$ iterations in node $1, 3, 5, \ldots, 2P+1$ and $\tau_{h,P}$ iterations in node $2, 4, 6, \ldots, 2P$, and with a constant stepsize s satisfying the conditions in Corollary* 5.10.

The number of iterations in the last node τ_a is bounded by

$$\tau_a \leq \max\left\{ \frac{\log\left(\dfrac{KZ}{\delta}\right)}{|\log \alpha_h|}, 0 \right\}, \tag{5.14}$$

where

$$Z \triangleq \eta^P \epsilon + \eta^P \alpha_H^{\tau_{H,P}} + \frac{1-\eta^{P+1}}{1-\eta} s\left((1+\alpha_H^{\tau_{H,P}})\epsilon + 2\right), \tag{5.15}$$

$$\eta \triangleq (|1-s| + s\alpha_H^{\tau_{H,P}})\alpha_h^{\tau_{h,P}}. \tag{5.16}$$

Proof. Using Lemma 5.11 implies

$$\begin{aligned} \|v_h^\star - v_a^{2P+1}\|_\infty &\leq \eta^P K\epsilon + \eta^P \alpha_H^{\tau_{H,P}} K + \left(\frac{1-\eta^{P+1}}{1-\eta}\right) sC \\ &\leq K \underbrace{\left[\eta^P \epsilon + \eta^P \alpha_H^{\tau_{H,P}} + \frac{1-\eta^{P+1}}{1-\eta} s\left((1+\alpha_H^{\tau_{H,P}})\epsilon + 2\right)\right]}_{=:Z}. \end{aligned}$$

Now let $v_h^0 \triangleq v_a^{2P+1}$ and select τ_a' such that

$$\|v_h^{\tau_a'} - v_h^\star\|_\infty \leq \alpha_h^{\tau_a'} \|v_h^0 - v_h^\star\|_\infty \leq \alpha_h^{\tau_a'} KZ = \delta$$

for a fixed tolerance δ. Using the same analysis as in Lemma 5.2, we obtain

$$\tau_a' = \frac{\log\left(\dfrac{KZ}{\delta}\right)}{|\log \alpha_h|}, \tag{5.17}$$

and therefore $\tau_a \leq \max\{\tau_a', 0\}$ as required. □

Theorem 5.13. *The $\mathcal{MMDP}(\epsilon, n, m, L)$, a P-AMS with $\tau_{H,P}$ iterations in node $1, 3, 5, \ldots, 2P+1$ and $\tau_{h,P}$ iterations in node $2, 4, 6, \ldots, 2P$, and a constant stepsize s satisfying the conditions in Corollary* 5.10 *have the following worst-case computational complexity,*

$$O\left(\left[\max\left\{ \frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|}, 0 \right\} + P\tau_{h,P} \right] (mn)^2 L + (P+1)\tau_{H,P} m^2 L^n \right), \tag{5.18}$$

where Z is defined in (5.15).

Proof. Using Lemma 5.12, we obtain the total number of iterations in both the fine and coarse models. Therefore, the complexity is the sum of the number of iterations multiplied by the cost of each iteration for both fine and coarse model. □

The theorem below shows that OWMS is one specific case of AMS if the parameters of AMS are judiciously chosen.

Theorem 5.14. *Suppose the tolerance $\delta < \min\{Z, \epsilon(2-\alpha_H)\}$, where Z is defined in* (5.15). *For $\mathcal{MMDP}(\epsilon, n, m, L)$, AMS has the same complexity as OWMS in the case when $P = 0$, $s = 0$, and $\tau_{H,P} = \tau_{H,0}'$, where $\tau_{H,0}'$ is defined in* (5.5).

Proof. Using Theorem 5.13 with $P = 0$, $s = 0$, and $\tau_{H,P} = \tau'_{H,0}$, where $\tau'_{H,0}$ is defined in (5.5), we obtain

$$\begin{aligned}
&\left[\frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|} + P\tau_{h,P}\right](mn)^2 L + (P+1)\tau_{H,P} m^2 L^n \\
&= \frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|}(mn)^2 L + \tau_{H,P} m^2 L^n \\
&= \frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|}(mn)^2 L + \frac{\log\left(\dfrac{1}{\epsilon(1-\alpha_H)}\right)}{|\log \alpha_H|} m^2 L^n,
\end{aligned}$$

where

$$\begin{aligned}
Z &= \eta^P \epsilon + \eta^P \alpha_H^{\tau_{H,P}} + \frac{1-\eta^{P+1}}{1-\eta} s\left((1+\alpha_H^{\tau_{H,P}})\epsilon + 2\right) \\
&= \epsilon + \alpha^{\tau'_{H,0}} \\
&= \epsilon + \epsilon(1-\alpha_H) \qquad \text{(Equation (5.4) implies } \alpha_H^{\tau'_{H,0}} = \epsilon(1-\alpha)) \\
&= \epsilon(2-\alpha_H).
\end{aligned}$$

Therefore, the complexity of AMS becomes

$$O\left(\frac{\log\left(\dfrac{\epsilon(2-\alpha_H)}{\delta}\right)}{|\log \alpha_h|}(mn)^2 L + \frac{\log\left(\dfrac{1}{\epsilon(1-\alpha_H)}\right)}{|\log \alpha_H|} m^2 L^n\right),$$

which is same as the OWMS. □

Using the preceding theorem, we can conclude that if we define the parameters,

$$\begin{aligned}
(s^\star, \tau^\star_{h,P}, \tau^\star_{H,P}, P^\star) = \arg \min_{s,\tau_{h,P},\tau_{H,P},P} &\left[\max\left\{\frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|}, 0\right\} + P\tau_{h,P}\right](mn)^2 L \\
&+ (P+1)\tau_{H,P} m^2 L^n,
\end{aligned}$$

then the complexity of AMS with parameters $(s^\star, \tau^\star_{h,P}, \tau^\star_{H,P}, P^\star)$ must be less than or equal to the complexity of OWMS. Notice that the parameters $(s^\star, \tau^\star_{h,P}, \tau^\star_{H,P}, P^\star)$ are optimizing the worse case complexity. While the result above is useful, it is difficult to obtain a closed form solution for these parameters. Instead, in our numerical results we will use a suboptimal solution that is motivated by the optimal parameter selection problem above.

6. Action space sampling for the coarse model. It follows from the complexity analysis of the preceding sections that the advantages of the coarse model are (a) the dimensionality reduction in the state space and (b) the improved convergence rate. However, the action space in the coarse model is exponentially larger than the fine model. If the fine model $\mathcal{MMDP}(\epsilon, n, m, L)$ is aggregated using perturbation theory, then the coarse model is an $\mathcal{MDP}(m, L^n)$ problem. We will take advantage of the well established links between linear programming and MDPs together with constraints sampling techniques from [5, 4] to address the computational cost associated with the coarse model.

6.1. Linear programming and MDPs. It is well known that the HJB equation (2.10) can be solved by the following LP:

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{X}^H} v_H(i) \\ \text{s.t.} \quad & v_H(i) \leq \frac{G^H(i, a_H)}{|q_{ii}^H(a_H)| + \rho} + \sum_{j \neq i} \frac{q_{ij}^H(a_H)}{|q_{ii}^H(a_H)| + \rho} v_H(j) \quad \forall a_H \in \mathcal{A}_i^H, i \in \mathcal{X}^H. \end{aligned}$$

The LP formulation of $\mathcal{MDP}(m, L^n)$ has m variables and mL^n constraints. For large-scale problems, it is very likely that $m \ll mL^n$, and so we will have a lot more constraints than variables. In this commonly encountered scenario, most of the constraints are not active at the optimum, and eliminating them could reduce the computational cost of the problem [8]. We will use the constraint sampling technique to reduce the action space and therefore the complexity. We will make the following assumption regarding the relationship between samples samples and states.

Assumption 1. Let $\mathcal{SSET} \triangleq \{(a_1, x_1), (a_2, x_2), (a_3, x_3), \ldots, (a_R, x_R)\}$ be the set of R samples from the probability mass function $\psi(a_H, x_H)$, and let $\mathcal{S}_i^A \triangleq \{(a_H, i) : a_H \in \mathcal{A}_i^H\}$. Then, $\mathcal{SSET} \cap \mathcal{S}_i^A \neq \emptyset \ \forall i \in \mathcal{X}^H$.

Assumption 1 implies that the set of samples will contain at least one state-action pair for each state. We use this assumption to ensure that our samples are enough to formulate another MDP which has action sets that are the (nonempty) subset of the action sets in the original problem.

LEMMA 6.1. *Let $\tilde{\mathcal{A}}_i^H \triangleq \{a_H : (a_H, i) \in \mathcal{SSET} \cap \mathcal{S}_i^A\}$; then the optimal solution of the LP*

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{X}^H} \tilde{v}_H(i), \\ s.t. \quad & \tilde{v}_H(i) \leq \frac{G^H(i, a_H)}{|q_{ii}^H(a_H)| + \rho} + \sum_{j \neq i} \frac{q_{ij}^H(a_H)}{|q_{ii}^H(a_H)| + \rho} \tilde{v}_H(j) \quad \forall (a_H, i) \in \mathcal{SSET} \end{aligned} \tag{6.1}$$

is the value function of the MDP

$$\begin{aligned} \min \quad & \tilde{J}^H(i, u_H) = \mathbb{E}\left[\int_0^\infty e^{-\rho t} G^H(x_H(t), u_H(x(t)))\ \mathrm{d}t\right], \\ s.t. \quad & x_H \sim Q_H(u_H(x_H(t)))\ ,\ t \geq 0, \\ & x_H(0) = i \quad , \quad u_H \in \tilde{\mathcal{U}}^H, \\ & \tilde{v}_H(i) = \min_{u_H \in \tilde{\mathcal{U}}^H} \tilde{J}^H(i, u_H), \end{aligned} \tag{6.2}$$

where $\tilde{\mathcal{U}}^H$ is the policy space for $\tilde{\mathcal{A}}_i^H \ \forall i \in \mathcal{X}^H$.

Proof. The LP in (6.1) is just the reformulation of (6.2) as a linear program. □

We refer to the MDP in (6.2) as the *reduced coarse model.* Note that in the reduced coarse model, we still maintain the fast convergence rate due to the elimination of the multiscale structure and at the same time we are able to control the complexity per iteration by decreasing the number of actions. Of course when reduced order policy space contains the optimal policy, then indeed the solution of the reduced coarse model coincides with the the coarse solution. This simple observation is established below.

LEMMA 6.2. *The solution of the reduced coarse model is the same as the solution of the coarse model when*

$$u^\star_H(i) \in \tilde{\mathcal{A}}^H_i \qquad \forall i \in \mathcal{X}^H. \tag{6.3}$$

In other words, the policy space of the reduced coarse model contains the optimal policy.

Proof. Since the policy space in the reduced coarse model is a subset of the policy space in the coarse model, if $u^\star_H$ minimizes the expected discounted cost in the coarse model, it also minimizes the expected discounted cost in the reduced coarse model with the same value function. □

Of course it is unreasonable to make such a strong assumption as the one above. Instead, we will analyze the performance of OWMS and AMS by sampling the actions in the coarse model uniformly. In practice, it is often the case that some action-state pairs are more important than others. As a result, the uniform distribution assumption may not be the best from a computational perspective. However, if no additional assumption is made about the MDP, then this is a valid assumption to examine. We use basic combinatorics to obtain a quantitative estimate of the probability of obtaining the optimal policy from the reduced coarse model.

THEOREM 6.3. *For $\mathcal{MMDP}(\epsilon, n, m, L)$, suppose that a P-AMS with $\tau_{H,P}$ iterations in node $1, 3, 5, \ldots, 2P+1$ and $\tau_{h,P}$ iterations in node $2, 4, 6, \ldots, 2P$ and stepsize s is used. If $R = (1-\sigma)^{1/m} L^n$ samples are drawn from A^H_i $\forall i \in \mathcal{X}^H$, then the optimal solution will be obtained with probability $1-\sigma$ and the complexity of AMS is*

$$O\left(\left[\max\left\{\frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|}, 0\right\} + P\tau_{h,P}\right](mn)^2 L + (P+1)\tau_{H,P} m^2 R\right), \tag{6.4}$$

where Z is defined in Lemma 5.12.

Proof. For state i in the coarse model, only one optimal action is needed to construct the optimal policy. For R actions that are drawn from $\mathcal{A}^H_i$, the total number of possible combinations is $\binom{L^n}{R}$. If the optimal action is obtained in the samples, the total number of possible combinations is $\binom{L^n-1}{R-1}$. So, the probability of obtaining the optimal action is

$$\frac{\dbinom{L^n-1}{R-1}}{\dbinom{L^n}{R}} = \frac{R}{L^n}.$$

The optimal policy is obtained only if each action space in the reduced coarse model contains its optimal action. The probability of obtaining the optimal policy is then $1-\sigma \triangleq (\frac{R}{L^n})^m$, and so $R = (1-\sigma)^{1/m} L^n$. If $\tilde{v}^\star_H = v^\star_H$, the convergence of the reduced coarse model is same as coarse model with less actions in each state. Therefore, with

$R = (1-\sigma)^{1/m} L^n$, the AMS has probability $1-\sigma$ to obtain $v_h^\star$ with complexity

$$O\left(\left[\max\left\{\frac{\log\left(\dfrac{Z}{\delta}\right)}{|\log \alpha_h|}, 0\right\} + P\tau_{h,P}\right](mn)^2 L + (P+1)\tau_{H,P} m^2 R\right),$$

as stated in Theorem 5.13. □

OWMS is a specific case of AMS, and so Theorem 6.3 covers the case of OWMS too. While the result above guarantees $\tilde{v}_H^\star = v_H^\star$ with certain probability, it may still require a large number of samples. To address this last point, we make use of the result from Calafiore [4] to estimate the number of action-state pairs that would guarantee that one extra action-state pair would not change the value function of the reduced coarse model with certain probability. This is useful because it provides a guide of the number of sufficient samples required to obtain the most "useful" actions so that the value function $\tilde{v}_H^\star$ is likely to be close to $v_H^\star$.

THEOREM 6.4. *Consider the reduced coarse model with state-action pairs set $\mathcal{SSET}$ with size mR (R actions in each state in average) and let Assumption* 1 *hold. Let $(\tilde{a}_H, \tilde{x}_H)$ be a state-action pair drawn from $\psi(a_H, x_H)$. Let $\tilde{v}_H^\star$ be the value function of reduced coarse model with $\mathcal{SSET}$, and $\hat{v}_H^\star$ be the value function of reduced coarse model with $\mathcal{SSET} \cup (\tilde{a}_H, \tilde{x}_H)$. Let*

$$\mathcal{V}(\mathcal{SSET}) \triangleq \mathbb{P}((\tilde{a}_H, \tilde{x}_H) \in \psi(a_H, x_H) : \tilde{v}_H^\star = \hat{v}_H^\star), \tag{6.5}$$

which is the probability that adding an extra action-state pair drawn from $\psi(a_H, x_H)$ does not change the optimal policy in the reduced coarse model. Then,

$$\mathbb{P}(\mathcal{SSET} \in \psi(a_H, x_H)^{mR} : \mathcal{V}(\mathcal{SSET}) \geq 1-\ell) \geq 1-\beta \tag{6.6}$$

if

$$mR \geq \frac{2}{\ell}\log\beta^{-1} + \frac{4}{\ell}(m-1) \tag{6.7}$$

for $\ell, \beta \in (0,1)$.

Proof. The proof is provided in [4]. □

Note that the lower bound of the number of state-action pairs does not depend on the number of state-action pairs in the original coarse model. In practice, it is possible that one would have an idea of what actions are more likely to be optimal in each state. Such knowledge can be used to construct a better constraint sampling density $\psi(a_H, x_H)$.

7. Numerical experiments. In this section, we illustrate the performance of the AMS and the OWMS using two numerical examples. The first example is the widely used example from the field of manufacturing we introduced in section 4.3. The second example is motivated from stochastic molecular dynamics. Applications in molecular dynamics (MD) have a strong multiscale structure, and applications related to MD and stochastic optimal control are beginning to emerge [14]. In the results below we compare the performance of the proposed algorithms and the conventional value iteration algorithm. The numerical performance of FAS has already been discussed. For reasons explained in section 4.4, FAS is not competitive with any of the other methods.

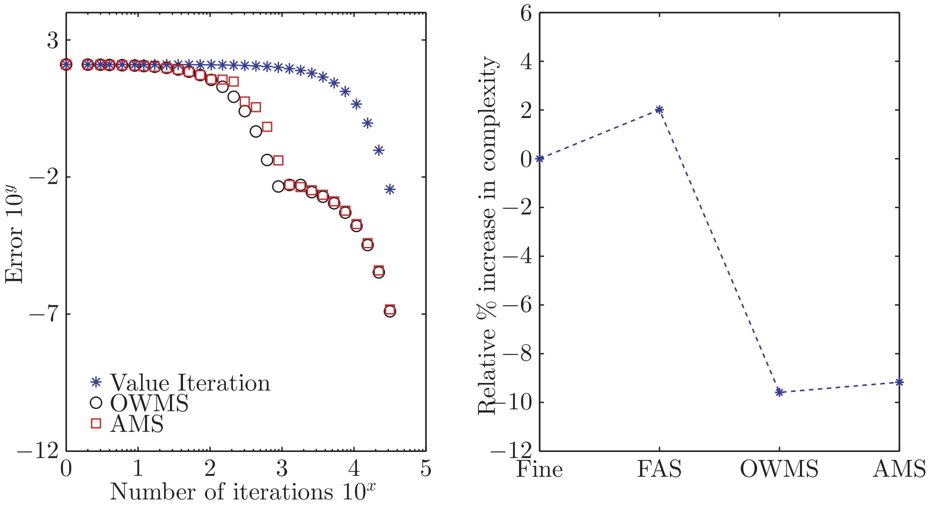


FIG. 7. *Numerical performance of the different algorithms. Parameters:* $v_h^0 = 0$, $v_H^0 = 0$, $s = 1.15$, *and* $\rho = 0.05$. *(Left) Iteration history. (Right) Relative increase in realized complexity of the different algorithms. Value iteration was taken to be the baseline. Compared to value iteration, conventional FAS has an increased complexity, whereas the proposed schemes achieve a* 10% *reduction.*

The AMS requires the specification of four parameters. In the numerical examples below we fix $\tau_{h,P} = \tau_{H,P} = 100$. The stepsize s is selected according to the estimates in Corollary 5.10. The parameter P is determined adaptively. To monitor the progress of the algorithm, we define the measure

$$\Phi(j) \triangleq ||A_h v_a^{j-2} - A_h v_a^j||_\infty \quad \forall j \in \{2 + 2i : i \in \mathbb{Z}^+\},$$

which can be interpreted as the improvement of the solution in node $j-1$ and j. It follows from Theorem 5.9 that the coarse correction is only guaranteed to be a contraction if the current solution is far away from the true solution. With that in mind, we define

$$\Psi(j) \triangleq \frac{\Phi(j-2) - \Phi(j)}{\Phi(j-2)} \quad \forall j \in \{2 + 2i : i \in \mathbb{Z}^+\}$$

to be the percentage change compared to the last two nodes. This measure of change was used to detect whether the coarse correction is still useful during calculations, and we stop using the coarse correction after node j when $\Psi(j)$ is less than some constant ϱ. In the experiments below we set $\varrho = 0.1$, and we used $\epsilon = 10^{-2}$. With this parameter choice for ϵ, the two problems contain some multiscale structure but it is not strong enough to just use the coarse model.

7.1. Manufacturing example. Recall that in section 4.3 we introduced an example motivated by a multiscale manufacturing process. We showed that the FAS fails in this example, and this was one of our motivations for developing the proposed scheme. Figure 7 shows the performance of the different schemes for this example. The parameter settings are exactly the same as in section 4.3. In Figure 7 (left) we show the iteration history, in terms of the distance to the exact solution, for each of the algorithms. The exact solution was obtained using linear programming. We do not plot the performance of the FAS for this model because it makes it difficult to see the differences between value iteration and our algorithms. The iteration history of FAS was plotted against value iteration in Figure 2. In Figure 7 (right) we plot the (relative) comparisons of realized complexity of the different algorithms as the number of iterations multiplied by the cost per iteration. We show the realized complexity of each algorithm in relation to the realized complexity of value iteration. Since the size of the problem is not large, we do not apply sampling in this example. Both of our proposed multiresolution algorithms are better than solely using the fine model. In this example, the total complexity can be reduced by 10% without any penalty on the accuracy. Notice that in this example, we have $\alpha_H = 0.9967$. Therefore, the choice of our stepsize $s = 1.15$ is reasonable because $s < 2/(1+\alpha_H^{\tau_{H,P}})$ with $\tau_{H,P} = 100$ and $s = 1.15$.

7.2. Example from molecular dynamics. In this section we use the proposed scheme to solve a larger problem motivated by molecular dynamics. The problem of controlling molecular dynamics is an active research area with many applications in material science and chemical engineering [14]. The potential energy of molecules is usually modeled as a stochastic differential equation, or as a Markov chain in the discrete case. Even though the transitions from one energy level to another energy level are considered to be stochastic, the underlying randomness is structured. Molecules are stable when they are at a local minimum of the potential energy and are very likely to make fast changes around the neighborhood of the local minimum. It is rare that molecules would move from one stable configuration to another. The event that a molecule jumps from one well to another is characterized as a rare event.

In this example, we consider a Markov chain with 50 states, where $\hat{Q}$ is a block diagonal matrix with 10 blocks, and each block is a 5×5 matrix. The state space is $\mathcal{X}^h = \{1, 2, \ldots, 50\}$. In this particular example, each block represents the transitions between different configurations within a stable configuration. In practice, this Markov chain is obtained by discretizing a stochastic differential equation. For this reason, we assume that $\hat{q}_{ij} = 0$ if $|i-j| > 1$. The matrix W represents the connection from one stable configuration to another. We assume $w_{ij} = 0$ if (i, j) is not from the set $\{(a, b) : (a, b) \in \{5, 6, 10, 11, 15, 16, 20, 21, 25, 26\}^2, |b-a| \leq 1\}$. Since the system is large, we simply sample the entries for $\hat{Q}$ and W uniformly from the set $\{1, 2, 3, \ldots, 9\}$. In order to introduce a control element into the model, we assume that there exists a catalysis that can be used to speed up or slow down the rate by which the system moves between states. To be precise, we assume the following form of the generator:

$$Q_h^\epsilon(a) = 3^a \left[\frac{1}{\epsilon}\hat{Q} + W\right],$$

where $a \in \{-1, -2/3, -1/3, 0, 1/3, 2/3, 1\}$. The objective function is $G(x, a) = x + 50|a|$, and as before we solve the infinite horizon model. Optimizing the system with this particular choice of cost function aims to control the dynamics so that the system remains in or close to state 1 without using too much catalysis. Figure 8 shows the numerical results of this example. For this example, FAS performed particularly badly. In order to have a clearer comparison between the proposed algorithms and value iteration, we do not plot the iteration history of the FAS. Notice that the OWMS scheme takes more time to converge than value iteration. After a few hundred iterations in the coarse model, the coarse model becomes ineffective because the current solution is "too close" to the exact solution. Therefore, in this example, OWMS spends a lot of

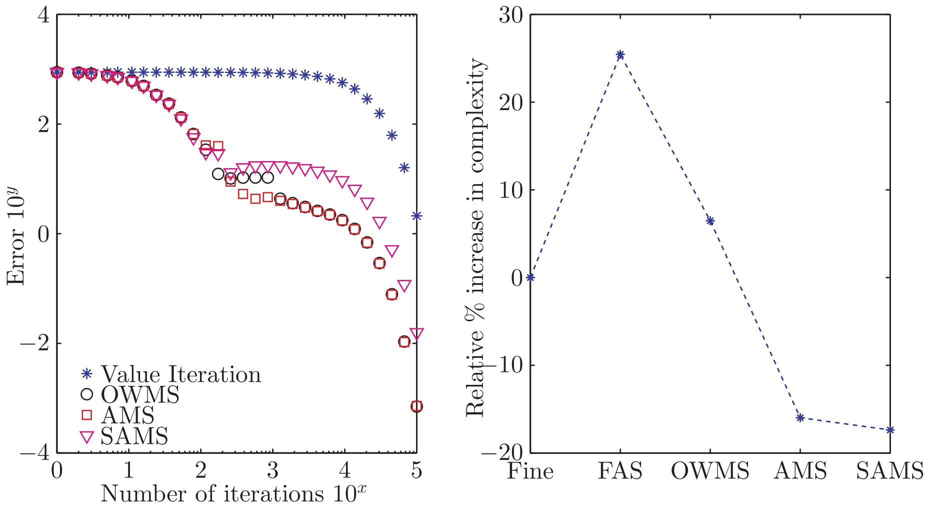


FIG. 8. *Numerical results of different algorithms. Parameters:* $v_h^0 = 0$*,* $v_H^0 = 0$*,* $s = 1.1$*,* $R = 10000$*, and* $\rho = 0.05$*. (Left) Convergence in each iteration. (Right) Relative increase in realized complexity of the different algorithms. Value iteration was taken to be the baseline.*

expensive iterations that do not achieve a significant error reduction. However, AMS still outperforms value iteration. In the fine model, each state has seven available actions, and so in the coarse model, each state has $7^5 = 16807$ actions. As the action space is so large, it is reasonable to apply the action sampling technique described in section 6. In Figure 8 we plot the results when using AMS with action sampling (SAMS), and $R = 10000$ samples. From Figure 8 (left) we see that applying the sampling technique leads to a slower convergence rate of our scheme compared to the original AMS and OWMS. However, since the size of the coarse model is reduced, the time spend per iteration is less. This point is made in Figure 8 (right), where the proposed scheme with action sampling computes the solution with less realized complexity compared to all the others. The performance is not much better than the original AMS due to the fact that uniform sampling is not very effective. In practice, we would expect that it is possible to find a good distribution $\psi(a_H, x_H)$ via empirical analysis. Still, even without optimizing the proposed schemes, we are able to achieve close to a 20% improvement over value iteration.

8. Discussion. We proposed the AMS for MDPs with a multiscale structure. Our scheme is an alternative framework and, under certain conditions, is theoretically superior to the conventional numerical methods used for this class of problems. The main idea of AMS is to use the coarse (aggregate) model as much as possible and avoid using the expensive full (fine) model for all the iterations. It was already known that the coarse model has fewer states than the fine model. But more importantly, we showed that the coarse model is also better conditioned. Using complexity analysis and numerical examples, we showed that the proposed scheme outperforms value iteration and the conventional multigrid based method (FAS). We also proposed a sampling method to address the problem of large action space in the coarse model.

Our proposed scheme exploits the multiscale structure of the problem but does not depend on it, i.e., the convergence does not depend on having scale separation. When there is no scale separation, the algorithm still computes the correct solution, but there are no real benefits in terms of reduction of computation times. When the multiscale structure is very sharp, most of the calculations will be done in the coarse model, and the final solution is computed by sightly correcting the approximate solution using the fine model.

We believe that the general scheme proposed in this paper can be extended to more general settings or even to different classes of problems. For example, one could replace value iteration by policy iteration. As long as the underlying algorithm is a contraction, the theoretical results of this paper can be used to evaluate its performance.

Acknowledgment. The authors wish to acknowledge the three anonymous referees for their helpful comments that led to substantial improvements of the paper.

REFERENCES

[1] M. Bard and C. March, *Multiscale singular perturbations and homogenization of optimal control problems*, Ser. Adv. Math. Appl. Sci., 76 (2008), pp. 1–27.

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control:* 2, Athena Scientific, Nashua, NH, 2007.

[3] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, 2000.

[4] G. C. Calafiore, *Random convex programs*, SIAM J. Optim., 20 (2010), pp. 3427–3464.

[5] G. Calafiore and M. C. Campi, *Uncertain convex programs: Randomized solutions and confidence levels*, Math. Program. Ser. A, 102 (2005), pp. 25–46.

[6] C. Chow and J. N. Tsitsiklis, *An optimal one-way multigrid algorithm for discrete-time stochastic control*, IEEE Trans. Automat. Control, 36 (1991), pp. 898–914.

[7] P. D. Christofides, *Control and Optimization of Multiscale Process Systems*, Birkhäuser, Basel, Switzerland, 2009.

[8] D. P. de Farias and B. V. Roy, *On constraint sampling in the linear programming approach to approximate dynamic programming*, Math. Oper. Res., 29 (2004), pp. 462–478.

[9] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, New York, 2003.

[10] P. Kokotovic, H. K. Khali, and J. O'reilly, *Singular Perturbation Methods in Control: Analysis and Design*, Vol. 25, SIAM, Philadelphia, 1987.

[11] S. P. Meyn, *Control Techniques for Complex Networks*, Cambridge University Press, Cambridge, 2008.

[12] P. Parpas and M. Webster, *A stochastic multiscale model for electricity generation capacity expansion*, European J. Oper. Res., 232 (2014), pp. 359–374.

[13] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley, New York, 2011.

[14] C. Schutte, S. Winkelmann, and C. Hartmann, *Optimial control of molecular dynamics using Markov state models*, Math. Program. Ser. B, 134 (2012), pp. 259–282.

[15] S. P. Sethi and Q. Zhang, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, Birkhäuser, Basel, Switzerland, 1994.

[16] H. A. Simon and A. Ando, *Aggregation of variables in dynamic systems*, Econometrica, 29 (1961), pp. pp. 111–138.

[17] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[18] G. Yin and Q. Zhang, *Continuous-Time Markov Chains and Applications*, 3rd ed., Springer, New York, 2013.