

# Bounded Aggregation for Continuous Time Markov Decision Processes

Peter Buchholz, Iryna Dohndorf, Alexander Frank,  
and Dimitri Scheftelowitsch<sup>(✉)</sup>

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

**Abstract.** Markov decision processes suffer from two problems, namely the so-called *state space explosion* which may lead to long computation times and the *memoryless property of states* which limits the modeling power with respect to real systems. In this paper we combine existing state aggregation and optimization methods for a new aggregation based optimization method. More specifically, we compute reward bounds on an aggregated model by exchanging state space size with uncertainty. We propose an approach for continuous time Markov decision models with discounted or average reward measures.

The approach starts with a portioned state space which consists of blocks that represent an abstract, high-level view on the state space. The sojourn time in each block can then be represented by a phase-type distribution (PHD). Using known properties of PHDs, we can then bound sojourn times in the blocks and also the accumulated reward in each sojourn by constraining the set of possible initial vectors in order to derive tighter bounds for the sojourn times, and, ultimately, for the average or discounted reward measures. Furthermore, given a fixed policy for the CTMDP, we can then further constrain the initial vector which improves reward bounds. The aggregation approach is illustrated on randomly generated models.

**Keywords:** Markov Decision Process · Aggregation · Discounted reward · Average reward · Bounds

## 1 Introduction

Continuous time Markov decision processes (CTMDPs) are a well-established class of stochastic processes which are widely applied in performance and dependability analysis. A significant problem for Markov models are uncomfortably high-dimensional state spaces which lead to high runtime complexities. One way to handle this issue is bounded state aggregation, where each state results from aggregation of several detailed states, as discussed in [3, 7]. Another problem when decisions are added to Markov models are the inherent memoryless property of states. In Markov models this problem is alleviated by using phase type

distributions (PHDs) [6]. However, this step can not be used in Markov Decision Processes (MDPs) because decisions are made in states and if PHDs are used for generally distributed sojourn times in states, each state of the PHD becomes a decision state which does not correspond to the state of the system where decisions can be made. By bounded aggregation also this problem can be handled.

The processes resulting from bounded aggregation is a so called bounded parameter MDP (BMDP), where for some parameters only intervals and not exact values are known. When lower and upper bounds are known, a set of CTMDPs, rather than a single CTMDP, is described. The goal of an optimization is then the minimization or maximization of the worst result value. The available papers on bounded aggregation in Markov processes or Markov decision processes consider only discrete time models. In this paper, the approach is extended to continuous time models coincidentally computing improved bounding parameters based on recent work and available results for PHDs.

An extensive discussion of MDP theory and its applications is given in [13], where different optimality criteria for the unaltered MDP formalism are discussed. Given some uncertainty in transitions and rewards, a more general Markov model is necessary, like BMDPs described in [11]. In [4] some variants of policy and value iteration approaches to compute an optimal policy and value vector for the case of average and discounted reward optimality measures are evaluated with respect to their runtime.

There exist several papers concerned with aggregation of MDPs. In [10], a factored MDP is reduced to a MDP with an exponentially smaller state space MDP by stochastic bisimulation, such that an optimal policy for the reduced one is also optimal for the original MDP. The authors in [12] introduce different abstraction schemes for the states of a MDP. Some other techniques for state aggregation are given by  $(\varepsilon_p, \varepsilon_f)$ -lumpable partitions [14] and  $\varepsilon$ -homogeneous partitions [8]. In [15] numerical methods for bounding the stationary distribution for large state spaces are given which can be extended to obtain better bounds for BMDP models.

For the discrete time Markov models with state aggregation some approximations are studied: [1] treats approximate policy iteration for the described problem with discounted rewards and [18] attends to a value iteration algorithm.

In [8], it is shown how the reduced MDP with states corresponding to blocks of a partition of the state space can be generated. Furthermore, upper and lower bounds on the transition probabilities and rewards in the resulting BMDP model correspond to bounds on the transition probabilities for states that are grouped in the same partition. However, the mentioned approach operates only on discrete time models, and it computes simple bounds using minimal and maximal exit probabilities out of aggregated blocks.

In this paper, the approach of [8] is extended. Bounded aggregation for continuous time Markov decision models and the differences compared to available methods for discrete time Markov formalisms are considered. For discounted CTMDPs, continuous time introduces a different bounded aggregation

method. Another goal is to improve upper and lower bounds for continuous time bounded (semi) Markov decision problems and compare the results to the previous work [8].

The paper is organized as follows. In the following section, we give an overview of the mathematical foundations for (semi) Markov decision processes and their extensions to uncertain transition probabilities as well as uniformization and optimization techniques for computation of optimal value vectors for Markov processes. Then, in Sect. 3, we briefly summarize known aggregation results for MDPS and develop an extended bounded aggregation approach for CTMDPs to derive a reduced state space model and make computing improved bounds and optimal policies tractable even for large state spaces. Finally, we continue with some examples and discuss the results in Sect. 5.

## 2 Background and Definitions

Here, we introduce basic definitions and notation. Vector and matrix identifiers are written in bold script, and individual elements of a vector  $\mathbf{v}$  or a matrix  $\mathbf{M}$  are accessed by  $\mathbf{v}(i)$  and  $\mathbf{M}(i, j)$ . A column vector of ones is designated by  $\mathbf{1}$ .

### 2.1 Markov Decision Processes

A continuous-time Markov decision process (CTMDP) is defined as a tuple  $(\mathcal{S}, \mathcal{A}, (\mathbf{Q}^a)_{a \in \mathcal{A}}, (\mathbf{r}^a)_{a \in \mathcal{A}}, \mathbf{p})$  where  $\mathcal{S}$  is a finite set of *states* of a given order  $n$ ,  $\mathcal{A}$  is a finite set of *actions* of order  $m$ ,  $\mathbf{Q}^a \in \mathbb{R}^{n \times n}$  is a transition rate matrix with  $\mathbf{Q}^a(i, j)$  giving the transition rate of moving from the state  $i$  to some state  $j$  when action  $a$  has been chosen. For the transition rate matrix  $\mathbf{Q}^a$  it has to hold that  $\mathbf{Q}^a \mathbf{1} = \mathbf{0}$  and  $\mathbf{Q}^a(i, j) \geq 0$  if  $i \neq j$  for all actions  $a \in \mathcal{A}$ . Furthermore, the initial probability distribution vector  $\mathbf{p} \in \mathbb{R}^{1 \times n}$  and the reward rate vector  $\mathbf{r}^a \in \mathbb{R}^{n \times 1}$  define a MDP. In the following, the states are numbered as  $\mathcal{S} = \{1, \dots, n\}$ , and the actions are numbered as  $\mathcal{A} = \{1, \dots, m\}$ .

To optimize some performance criteria of a CTMDP decision rules and policies are specified. A decision rule is a mapping  $\mathbf{u}_t : \mathcal{S} \rightarrow \mathcal{A}$  which is an assignment of actions to states at some point in time  $t$ . A policy can then be defined as a sequence of decision rules  $\pi = (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_T)$  for some  $T \leq \infty$ . A deterministic policy which is independent of time  $t$  is called *pure*. We consider here only *pure* policies and denote them for simplicity as *policies*. A pure policy can be described by a vector  $\pi \in \mathcal{A}^{\mathcal{S}}$ . We use the notation  $\pi$  to denote the policy whereas the vector notation  $\pi$  is applied if specific elements of the policy are accessed, i.e.  $\pi(s)$  is the action chosen in state  $s$  under policy  $\pi$ . We designate by  $\mathbf{Q}^{\pi(t)}$  and  $\mathbf{r}^{\pi(t)}$  the matrices and vectors that are constructed from  $\mathbf{Q}^{\pi(t,s)}$  and  $\mathbf{r}^{\pi(t,s)}$  in row  $s$ . If the MDP is in state  $s$  and action  $a$  is selected, then it accumulates reward with rate  $\mathbf{r}^{\pi(t,s)}(s)$ , its sojourn time in this state is exponentially distributed with rate  $-\mathbf{Q}^{\pi(t,s)}(s, s)$ , and the transition probability to a different state  $s'$  is  $\frac{\mathbf{Q}^{\pi(t,s)}(s, s')}{-\mathbf{Q}^{\pi(t,s)}(s, s)}$ . For the definition of optimal policies and their values in CTMDPs as the methods for computing them we refer to the literature [5, 13, 17].

**Uniformization:** For long-term, “stationary” reward measures such as expected average reward and expected discounted reward, it is possible to transform continuous-time MDPs into discrete-time models with the same optimality behavior, which is sufficient for finding optimal policies [2, 13, 16]. Intuitively, a CTMDP is transformed into a discrete-time MDP in two steps: First, the sojourn time distributions are made identical for all states by introducing virtual self-transitions. Second, with uniform sojourn time distribution, the CTMDP is transformed to an equivalent discrete-time MDP.

Given a CTMDP  $(\mathcal{S}, \mathcal{A}, (\mathbf{Q}^a)_{a \in \mathcal{A}}, (\mathbf{r}^a)_{a \in \mathcal{A}}, \mathbf{p})$ , we can transform it into a discrete-time MDP with the following operations. First, we choose  $\lambda \geq -\mathbf{Q}^a(s, s)$  for all  $s \in \mathcal{S}, a \in \mathcal{A}$ ;  $\lambda$  is the so-called *uniformization rate*. Define matrices  $(\mathbf{P}^a)_{a \in \mathcal{A}}$  and vectors  $(\mathbf{z}^a)_{a \in \mathcal{A}}$  with  $\mathbf{P}^a = \mathbf{I} + \frac{1}{\lambda} \mathbf{Q}^a$ . The reward vectors are modified depending on the reward measure. For expected discounted rewards with discount rate  $\beta$ , we define reward vectors  $\mathbf{z}_\beta^a$  with  $\mathbf{z}_\beta^a(s) = \frac{\mathbf{r}^a(s)}{\lambda + \beta}$  and assume a discount factor  $\gamma = \frac{\lambda}{\lambda + \beta}$ . For the expected average reward measure, the reward vectors are  $\bar{\mathbf{z}}^a$  with  $\bar{\mathbf{z}}^a(s) = \frac{\mathbf{r}^a(s)}{\lambda}$ .

This construction yields a discrete-time MDP  $(\mathcal{S}, \mathcal{A}, (\mathbf{P}^a)_{a \in \mathcal{A}}, (\mathbf{z}^a)_{a \in \mathcal{A}})$ , where, depending on the reward measure selected, the reward vectors  $\mathbf{z}^a$  are either  $\bar{\mathbf{z}}^a$  or  $\mathbf{z}_\beta^a$ . The uniformization method is summarized in Algorithm 1.

---

**Algorithm 1.** Uniformization method for CTMDPs

---

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

- 1:  $\lambda = \max_{\forall i \in \mathcal{S}, \forall a \in \mathcal{A}} |\mathbf{Q}^a(i, i)|$ ;
- 2: **for**  $a \in \mathcal{A}$  **do**
- 3:    $\mathbf{P}^a = \mathbf{I} + \frac{1}{\lambda} \mathbf{Q}^a$ ;
- 4:   **if** *discounted* **then**
- 5:      $\mathbf{z}^a(i) = \frac{\mathbf{r}^a(i)}{\lambda + \beta}, \quad \forall i \in \mathcal{S}$ ;
- 6:   **else**
- 7:      $\mathbf{z}^a(i) = \frac{\mathbf{r}^a(i)}{\lambda}, \quad \forall i \in \mathcal{S}$ ;
- 8:   **if** *discounted* **then**
- 9:      $\gamma = \frac{\lambda}{\lambda + \beta}$ ;
- 10: **return** Discrete-time MDP  $(\mathcal{S}, \mathcal{A}, (\mathbf{P}^a)_{a \in \mathcal{A}}, (\mathbf{z}^a)_{a \in \mathcal{A}})$ , discount factor  $\gamma$  if *discounted* is true;

---

## 2.2 Bounded-Parameter Markov Decision Processes

In most cases, the parameters of a stochastic model are not known exactly. Consequently, they can be given by intervals rather than point estimates. The formalism of *bounded-parameter MDPs* [8, 11] captures this concept. Bounded-parameter MDPs have been often defined in the literature. We review their definition and some optimality results here.

A bounded-parameter MDP is a tuple  $\{\mathcal{S}, \mathcal{A}, \left(\mathbf{P}_{\downarrow}^a\right)_{a \in \mathcal{A}}, \left(\mathbf{r}_{\downarrow}^a\right)_{a \in \mathcal{A}}\}$  containing a set of discrete-time MDPs defined by a state and action space  $\mathcal{S}, \mathcal{A}$ . The discounting is performed with a discount factor  $\gamma \in [0, 1)$ . For each action  $a \in \mathcal{A}$  lower and upper bounds on the transition probability parameters  $\mathbf{P}_{\downarrow}^a = (\mathbf{P}_{\downarrow}^a, \mathbf{P}_{\uparrow}^a)$  are defined with matrices  $\mathbf{P}_{\downarrow}^a, \mathbf{P}_{\uparrow}^a$  satisfying the conditions

$$\mathbf{P}_{\downarrow}^a \leq \mathbf{P}_{\uparrow}^a,$$

where  $\mathbf{P}_{\downarrow}^a, \mathbf{P}_{\uparrow}^a \in \mathbb{R}_{\geq 0}^{n \times n}$  and  $\mathbf{P}_{\downarrow}^a \mathbf{1} \leq \mathbf{1} \leq \mathbf{P}_{\uparrow}^a \mathbf{1}$ . Similarly, lower and upper bounds for the rewards are defined as  $\mathbf{r}_{\downarrow}^a = (\mathbf{r}_{\downarrow}^a, \mathbf{r}_{\uparrow}^a)$  with  $\mathbf{r}_{\downarrow}^a, \mathbf{r}_{\uparrow}^a \in \mathbb{R}_{\geq 0}^{n \times 1}$  where the condition

$$\mathbf{r}_{\downarrow}^a \leq \mathbf{r}_{\uparrow}^a$$

is satisfied for all actions  $a \in \mathcal{A}$ .

The BMDP model defines a set of discrete-time MDPs with parameters varying according to this bounds. One is often interested in the set of policies that optimize the lower and upper bounds for the reward measures from the set  $\mathbf{r}_{\downarrow}^a$ . These policies are permissible for the whole set of MDPs contained in  $\{\mathcal{S}, \mathcal{A}, \left(\mathbf{P}_{\downarrow}^a\right)_{a \in \mathcal{A}}, \left(\mathbf{r}_{\downarrow}^a\right)_{a \in \mathcal{A}}\}$ , the optimistic policy optimizing the upper bound of  $\mathbf{r}_{\downarrow}^a$ , and the pessimistic policy optimizing the lower bound for the rewards. In the following we consider only the lower bound computation, since the upper bound case is analogous. As in the area of robust optimization, the objective is to obtain the optimal solution for the whole uncertainty set. For BMDPs with discounted reward criterion one is interested in policy that maximizes the lower bound. In the pessimistic case the policy should fulfill

$$\pi_{\downarrow} = \arg \max_{\pi \in \Pi} \min_{\mathbf{P}^{\pi} \in \mathbf{P}_{\uparrow}^{\pi}} g_{\gamma \downarrow}^{\pi} \quad (1)$$

for a value function  $g_{\gamma \downarrow}^{\pi}$  which maps the policy  $\pi$  and the  $\gamma$ -discounted MDP  $\mathbf{P}^{\pi}$  to the value of  $\pi$  in the MDP  $\mathbf{P}^{\pi}$ . To obtain the optimal pessimistic gain vector the Bellman-like equation has to be solved for each state  $i \in \mathcal{S}$

$$g_{\gamma \downarrow}(i) = \max_{\pi \in \Pi} \min_{\mathbf{P}^{\pi} \in \mathbf{P}_{\uparrow}^{\pi}} \left( \mathbf{r}_{\downarrow}^{\pi}(i) + \gamma \sum_{j \in \mathcal{S}} \mathbf{P}^{\pi}(i, j) g_{\gamma \downarrow}(j) \right). \quad (2)$$

The analysis of a BMDP can be performed efficiently regarding the nested max min operator [4]. For the BMDPs with average reward criterion, define the expected reward in the  $k$ -th step in the future  $R(\mathbf{r}, \mathbf{P}, k) = \mathbf{P}^k \mathbf{r}$ . Then an optimal policy and the associated gain vector is the solution of the following equations.

$$\begin{aligned} \bar{g}_{\downarrow} &= \max_{\pi \in \Pi} \min_{\mathbf{P} \in \mathbf{P}_{\uparrow}^{\pi}} \lim_{K \rightarrow \infty} \left( \frac{1}{K} \sum_{k=1}^K R(\mathbf{r}_{\downarrow}^{\pi}, \mathbf{P}, k) \right), \\ \pi_{\downarrow} &= \arg \max_{\pi \in \Pi} \min_{\mathbf{P} \in \mathbf{P}_{\uparrow}^{\pi}} \lim_{K \rightarrow \infty} \left( \frac{1}{K} \sum_{k=1}^K R(\mathbf{r}_{\downarrow}^{\pi}, \mathbf{P}, k) \right). \end{aligned} \quad (3)$$

For further reference on analysis algorithms for BMDPs, we refer to [4, 11].

### 3 Bounded Aggregation Approach

Now that the backgrounds and syntax are clear, we are able to deal with aggregation. In the first part of this section we describe a common method for state space aggregation. Then in the second part we present our main contribution by introducing a new aggregation method for CTMDPs with specific constraints. Our refinement is in the calculation of the rates  $\lambda_i^{a-}$  and  $\lambda_i^{a+}$ , which are the bounds for the diagonal elements of the aggregated transition rate matrices. The last part consists of an application example.

#### 3.1 Aggregation of MDPs

In this subsection we describe existing concepts of state space aggregation and of bounds and apply known results to continuous Markov processes. The state space  $\mathcal{S}$  can be clustered into blocks with states from the set  $\mathcal{S}_{ij}$  which exhibit nearly the same stochastic behavior with respect to other blocks [7–9]. There are different motivations and approaches to define or compute the state space partition. In general, the computation of an *optimal* partition with respect to minimal rates between blocks is NP-hard [8]. This implies that only heuristic approaches for computing a partition are useful. In our setting an additional motivation exists, namely the combination of states where decisions should be identical (e.g. due to physical restrictions like the unobservability of the detailed state). In the aggregated process all states in a block are represented by a single state such that a single decision is naturally chosen in this state.

Typically, parameter bounds for the aggregated process are obtained due to the bounds on the transition probabilities of separated blocks. Consider a continuous time Markov decision model. Assume that the generator matrix  $\mathbf{Q}^a$  can be structured into  $k$  submatrices  $\mathbf{Q}_{ij}^a$  of dimension  $n_i \times n_j$  belonging to some block of states  $\mathbf{B}_{ij}$ .

$$\mathbf{Q}^a = \begin{bmatrix} \mathbf{Q}_{11}^a & \cdots & \mathbf{Q}_{1k}^a \\ \vdots & \ddots & \vdots \\ \mathbf{Q}_{k1}^a & \cdots & \mathbf{Q}_{kk}^a \end{bmatrix}$$

Then the aggregated Markov process can be generated by substituting each block  $\mathbf{B}_{ij}$  by a single macro state  $s \in \{1, \dots, k\}$  thus shrinking the initial state space to overall  $k$  aggregates. Let  $\tilde{\mathcal{S}}$  denote the state space structured into macro states. Let now  $0 \leq q_{ij}^{a-} \leq q_{ij}^{a+} < \infty$  be the upper and lower bounds for transition rates between two macro states  $i, j \in \tilde{\mathcal{S}}$  as described in [3]. The bounds can then be computed with

$$\begin{aligned} q_{ij}^{a-} &= \min_{m=1, \dots, n_i} \left( \sum_{l=1}^{n_j} \mathbf{Q}_{ij}^a(m, l) \right) \\ q_{ij}^{a+} &= \max_{m=1, \dots, n_i} \left( \sum_{l=1}^{n_j} \mathbf{Q}_{ij}^a(m, l) \right) \end{aligned} \tag{4}$$

such that intervals bounding the uncertain transition rates between two macro states  $i$  and  $j$  can be easily obtained as  $q_{ij\downarrow}^a = \{q^a \mid q_{ij}^{a-} \leq q^a \leq q_{ij}^{a+}\}$ .

### 3.2 New Aggregation Method for CTMDPs

If we assume that the system starts in some macro state  $i \in \tilde{\mathcal{S}}$ , then the expected sojourn time in  $i$  under decision  $a$  can be obtained using phase-type distribution (PHD) with subgenerator matrix  $\mathbf{D}_i^a = \mathbf{Q}_{ii}^a$ . Then, the process stays in macro state  $i$  a time which is distributed according to the PHD with parameters  $(\phi, \mathbf{D}_i^a)$  and afterwards moves to the next macro state  $j$  with rate  $q_{ij}^a \in q_{ij\uparrow}^a$ .

Let us consider some macro state  $i \in \tilde{\mathcal{S}}$ . Rewriting the generator matrix  $\mathbf{Q}^a$  for some action  $a \in \mathcal{A}$  as

$$\mathbf{Q}^a = \begin{bmatrix} \mathbf{Q}_{ii}^a & \mathbf{E}_{i\rightarrow}^a \\ \mathbf{F}_{i\leftarrow}^a & \begin{pmatrix} \mathbf{Q}_{jj}^a & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \mathbf{Q}_{kk}^a \end{pmatrix} \end{bmatrix}, \quad (5)$$

where the transition rate matrix  $\mathbf{E}_{i\rightarrow}^a$  is of dimension  $n_i \times \sum_{l \in \tilde{\mathcal{S}} \setminus \{i\}} n_l$  and the matrix  $\mathbf{F}_{i\leftarrow}^a$  is of dimension  $\sum_{l \in \tilde{\mathcal{S}} \setminus \{i\}} n_l \times n_i$ , the initial vector of the PHD with subgenerator  $\mathbf{D}_i^a$  can be approximated using rows of the matrix  $\mathbf{F}_{i\leftarrow}^a$  as follows.

Let  $q = \sum_{l=1, l \neq i}^k n_l$  be the number of rows of the matrix  $\mathbf{F}_{i\leftarrow}^a$ . Note that PHD with subgenerator  $\mathbf{D}_i^a$  describes the sojourn time distribution of a macro state  $i$  corresponding to the submatrix  $\mathbf{Q}_{ii}^a$ . The initial vector  $\phi$  of the PHD with subgenerator  $\mathbf{D}_i^a$  can be guessed using rows of the matrix  $\mathbf{F}_{i\leftarrow}^a$  in order to bound the sojourn time as follows. We obtain an initial vector  $\phi_l^a$ , for each row  $l \in \{1, \dots, q\}$  of  $\mathbf{F}_{i\leftarrow}^a$ ,  $\forall a \in \mathcal{A}$  where  $\mathbf{F}_{i\leftarrow}^a(l \bullet)$  is the  $l$ th row of the matrix  $\mathbf{F}_{i\leftarrow}^a$  and  $\mathbf{F}_{i\leftarrow}^a(l \bullet) \neq \mathbf{0}$  as

$$\phi_l^a = \mathbf{F}_{i\leftarrow}^a(l \bullet) / \|\mathbf{F}_{i\leftarrow}^a(l \bullet)\|_1. \quad (6)$$

Note that Eq. 6 is in fact a normalization of the vector  $\mathbf{F}_{i\leftarrow}^a(l \bullet)$ . Evaluating the Eq. 6 for all non-zero rows  $q$  of  $\mathbf{F}_{i\leftarrow}^a$  and for all  $a \in \mathcal{A}$ , the initial vectors resulting in minimal and maximal expected sojourn times of the PHD with subgenerator  $\mathbf{D}_i^a$  can be computed. Then, the sojourn time bounds can be obtained as

$$\begin{aligned} \nu_i^{a-} &= \min_{\forall \phi_j \in \Phi} (\phi_j (-\mathbf{D}_i^a)^{-1} \mathbf{1}), \forall a \in \mathcal{A} \\ \nu_i^{a+} &= \max_{\forall \phi_j \in \Phi} (\phi_j (-\mathbf{D}_i^a)^{-1} \mathbf{1}), \forall a \in \mathcal{A} \end{aligned} \quad (7)$$

where  $\mathbf{1}$  is a vector of dimension  $n_i \times 1$ . In Eq. 7,  $\Phi$  is the set containing probability distribution vectors computed using (6) for all non-zero rows  $l \in \{1, \dots, q\}$  of  $\mathbf{F}_{i\leftarrow}^a$ , and all  $a \in \mathcal{A}$ . The rate bounds for the sojourn time distributions can then be estimated by  $\lambda_i^{a-} = \frac{1}{\nu_i^{a-}}$  and  $\lambda_i^{a+} = \frac{1}{\nu_i^{a+}}$  for all macro states  $i \in \tilde{\mathcal{S}}$ .

Now we turn our attention to the non-diagonal elements  $q_{ij}^{a\pm}$ . As the value of  $q_{ij}^{a\pm}$  is a bound on the exit rate from macro state  $i$  to macro state  $j$ , we bound it by computing the probability to enter macro state  $j$  from macro state  $i$ . This probability is  $\phi_k (-\mathbf{D}_i^a)^{-1} \mathbf{Q}_{ij}^a \mathbf{1}$ . By multiplying it with the rate bounds we get the bounds

$$\begin{aligned}
q_{ij}^{a-} &= \lambda_i^{a+} \min_{\forall \phi_k \in \Phi} (\phi_k (-D_i^a)^{-1} Q_{ij}^a \mathbf{1}), \forall a \in \mathcal{A} \\
q_{ij}^{a+} &= \lambda_i^{a-} \max_{\forall \phi_k \in \Phi} (\phi_k (-D_i^a)^{-1} Q_{ij}^a \mathbf{1}), \forall a \in \mathcal{A}.
\end{aligned} \tag{8}$$

as  $\nu_i^{a+} \geq \nu_i^{a-} \Leftrightarrow \lambda_i^{a+} \leq \lambda_i^{a-}$ .

Together, we obtain a continuous time BMDP model where  $\lambda_i^{a-}$  and  $\lambda_i^{a+}$  from (7) specify bounds of an exponential distribution for each macro state and (8) specifies bounds for transition rates between macro states. The resulting aggregated process is shown below.

$$Q^{a-} = \begin{bmatrix} -\lambda_1^{a-} & q_{12}^{a-} & \cdots & q_{1k}^{a-} \\ q_{21}^{a-} & -\lambda_2^{a-} & \cdots & q_{2k}^{a-} \\ \vdots & \ddots & \ddots & \vdots \\ q_{k1}^{a-} & q_{k2}^{a-} & \cdots & -\lambda_k^{a-} \end{bmatrix}, \quad Q^{a+} = \begin{bmatrix} -\lambda_1^{a+} & q_{12}^{a+} & \cdots & q_{1k}^{a+} \\ q_{21}^{a+} & -\lambda_2^{a+} & \cdots & q_{2k}^{a+} \\ \vdots & \ddots & \ddots & \vdots \\ q_{k1}^{a+} & q_{k2}^{a+} & \cdots & -\lambda_k^{a+} \end{bmatrix}. \tag{9}$$

As for bounding matrices  $Q^{a-}(i, j) \leq Q^{a+}(i, j)$  has to hold, the diagonal elements are  $-\lambda_i^{a-}$  resp.  $-\lambda_i^{a+}$  since  $-\lambda_i^{a-} \leq -\lambda_i^{a+}$ . Afterwards, the obtained bounds can be further improved as follows. First, we apply the uniformization technique described in Sect. 2.1 and solve Eq. 3 for the aggregated discrete-time BMDP model resulting from the uniformization. Then we use the pessimistic optimal policy  $\pi_\downarrow$  to update the bounds, but, in principle, also the optimistic optimal policy can be used to obtain tighter bounds.

Assume that the optimal policy  $\pi_\downarrow$  obtained for an aggregated process holds for all states partitioned in a block corresponding to the macro state for which the optimal action has been determined. We compute Eq. 7 where possible initial vectors in the set  $\Phi$  are obtained using optimal policy  $\pi_\downarrow$  as follows

$$\phi_l^{\pi_\downarrow(l)} = F_{i \leftarrow}^{\pi_\downarrow(l)}(l \bullet) / \|F_{i \leftarrow}^{\pi_\downarrow(l)}(l \bullet)\|_1, \tag{10}$$

for all non-zero rows  $l$  of the policy matrix  $Q^{\pi_\downarrow}$  which is assembled by picking  $Q^{\pi_\downarrow}(l \bullet) = Q^{\pi_\downarrow(l)}(l \bullet)$ . At first we optimize the sojourn time bounds  $\nu_i^{a-}$  and  $\nu_i^{a+}$  over the whole set of initial vectors  $\Phi$ . The update step supplies us a reduced subset of  $\Phi$  that leads to an improved optimization. In general we get tighter bounds for the sojourn times by recalculating. We can now summarize our approach in Algorithm 2.

---

**function** COMPUTE\_INITIAL\_VECTORS(Set  $\mathcal{F}$  containing matrices  $F$ )

$\Phi = \emptyset$ ;

**for**  $F \in \tilde{\mathcal{F}}$  **do**

$q = \text{rows}(F)$ ;

**for**  $i = 1 \rightarrow q$  **do**

        Compute  $\phi_i$  as given in Eq. 6;

$\Phi = \Phi \cup \phi_i$ ;

**return** Set  $\Phi$  containing guessed initial vectors;

---



---

**function** COMPUTE\_SOJOURN\_TIME\_BOUNDS( $\Phi$ , Set  $\mathcal{D}_i$  containing matrices  $D_i^a$ ,  $a \in \{1, \dots, m\}$ )  
 $\Lambda = \emptyset$ ;  
 Evaluate Eq. 7 using sets  $\Phi$  and  $\mathcal{D}_i$ ; Save results in the set  $\Lambda$ ;  
**return** Set  $\Lambda$  containing  $\lambda_i^{a-}$ ,  $\lambda_i^{a+}$  for all  $a \in \mathcal{A}$ ;

---



---

**Algorithm 2.** Aggregation algorithm for CTMDPs

---

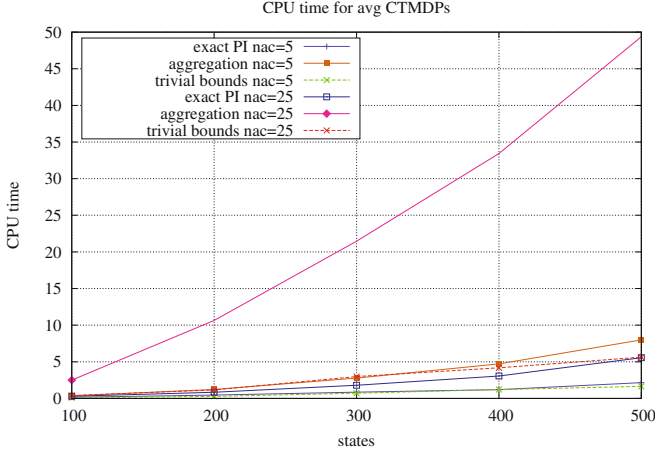
**Require:** Block structured CTMDP process with decision matrices  $Q^a$ ,  $\forall a \in \mathcal{A}$ . For each generator  $Q^a$ ,  $b$  blocks of dimension  $n_i \times n_j$  and corresponding submatrices  $Q_{ij}^a$ . Independent of  $a$  we define  $k$  as the number of blocks in a row of  $Q^1$ .

- 1:  $\mathcal{F} = \emptyset$ ;  $\mathcal{D}_1 = \emptyset, \dots, \mathcal{D}_k = \emptyset$ ;  $\Lambda_i = \emptyset, \dots, \Lambda_k = \emptyset$ ;
- 2: **for**  $i = 1 \rightarrow k$  **do**
- 3:   **for**  $\forall a \in \mathcal{A}$  **do**
- 4:     Compute a PHD with subgenerator  $D_i^a = Q_{ii}^a$ ;  $\mathcal{D}_i = \mathcal{D}_i \cup D_i^a$ ;
- 5:     Compute matrix  $F_{i\leftarrow}^a$  as given in Eq. 5;  $\mathcal{F} = \mathcal{F} \cup F_{i\leftarrow}^a$ ;
- 6:      $\Phi = \text{compute\_initial\_vectors}(\mathcal{F})$  ;
- 7:      $\Lambda_i = \text{compute\_sojourn\_time\_bounds}(\Phi, \mathcal{D}_i)$ ;
- 8:      $\tilde{r}_i^{a+}, \tilde{r}_i^{a-} = \text{maximum/minimum of all rewards in this block and action}$ ;
- 9:   **for**  $j = 1 \rightarrow k$  **do**
- 10:     **if**  $i \neq j$  **then**
- 11:       Compute transition rate bounds  $q_{ij}^{a-}$  and  $q_{ij}^{a+}$  by the given set  $\Phi$  using Eq. 8  $\forall a \in \mathcal{A}$ ;
- 12:    $\mathcal{F} = \emptyset$ ;
- 13:   Compute bounded discrete-time MDP model  $\left( \tilde{\mathcal{S}}, \mathcal{A}, \left( P_{\uparrow}^a \right)_{a \in \mathcal{A}}, \left( r_{\uparrow}^a \right)_{a \in \mathcal{A}}, p_{\uparrow} \right)$  using Algorithm 1;
- 14:   Compute optimal policy  $\pi_{\downarrow}$  and gain vector  $g_{\downarrow}$  using Eq. 1 and Eq. 2 or Eq. 3 ;
- 15:   Determine policy matrix  $Q^{\pi_{\downarrow}}$  with  $Q^{\pi_{\downarrow}}(l \bullet) = Q^{\pi_{\downarrow}(l)}(l \bullet)$  for each row  $l$  of  $Q^{\pi_{\downarrow}}$ ;
- 16:   **for**  $i = 1 \rightarrow k$  **do**                                     $\triangleright$  Update bounds according to the optimal policy
- 17:     Compute matrix  $F_{i\leftarrow}^{\pi_{\downarrow}}$  as given in Eq. 5;  $\mathcal{F} = \mathcal{F} \cup F_{i\leftarrow}^{\pi_{\downarrow}}$ ;
- 18:      $\Phi = \text{compute\_initial\_vectors}(\mathcal{F})$  ;
- 19:      $\Lambda_i = \text{compute\_sojourn\_time\_bounds}(\Phi, \mathcal{D}_i)$ ;
- 20:   **for**  $j = 1 \rightarrow k$  **do**
- 21:     **if**  $i \neq j$  **then**
- 22:       Compute transition rate bounds  $q_{ij}^{a-}$  and  $q_{ij}^{a+}$  by the given set  $\Phi$  using Eq. 8  $\forall a \in \mathcal{A}$ ;
- 23:    $\mathcal{F} = \emptyset$ ;
- 24:   Compute set of aggregated transition rate matrices  $\left( \tilde{Q}_{\uparrow}^a \right)_{a \in \mathcal{A}}$  like in 9;
- 25: **return**  $\left( \tilde{Q}_{\uparrow}^a \right)_{a \in \mathcal{A}}, \left( \tilde{r}_{\uparrow}^a \right)_{a \in \mathcal{A}}$

---

## 4 Experiments

We perform different experiments with randomly generated CTMDP instances with state space sizes ranging from 100 to 500 to compare the different aggregation approaches. All computations were performed on a machine with a 3.0 GHz



**Fig. 1.** CPU times for average reward criterion aggregation algorithms. The plotted results are obtained for random CTMDP models with 5 and 25 actions, an average reward of 5, an average sojourn time of  $1/5 \cdot |S|$  and a block size of mean  $10/3$ .

20-Core processor and 126 GB main memory running Debian Linux. We used *Matlab* implementation of our algorithms.

First, we analyzed randomly generated CTMDP models with dense matrices and reward vectors with a number of states varied from 100 to 500 and a number of actions  $\mathcal{A} = \{5, 25\}$ . The average sojourn times (the entrees on the diagonal of a transition rate matrix) depends on the size of states and is  $1/5 \cdot |S|$  and all nondiagonal elements are randomly and normalized to the sojourn time. Also the number of blocks and their size depend on  $|S|$ , cause there are  $3/10 \cdot |S|$  blocks given with a mean size of  $10/3$ . In both cases, the discounted and the average, the rewards are expected 5. For the discounted problem we test different values for  $\beta$ , but the results are similar enough to show you only one case for  $\beta = 2$ . For every combination of state space and actions we build ten examples, compare the separate results of the aggregation methods, and then we compute the mean of them. To obtain the average or discounted reward value and optimal strategy policy iteration method has been applied.

The plots in Fig. 1 show computation times for exact and aggregation algorithms for CTMDPs as a function of the number of states. We compared results obtained using the exact solution method, trivial aggregation and the improved aggregation methods. In the trivial aggregation method bounds for block sojourn times are obtained using minimal and maximal exit rates out of block. As one can see, the improved aggregation algorithm requires much more computation time. The reason is the computational effort required to compute Eq. 7 in order to derive tighter bounds.

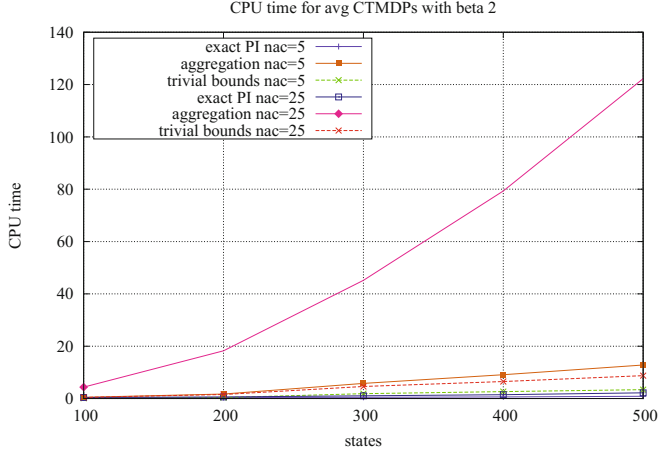
In Tables 1 and 2 we list the compared values computed by an intuitive aggregation algorithm and by our new aggregation method. The exact solution is only given to show you the quality of our results. The main consequence is the relative

**Table 1.** Results for average CTMDPs with 25 actions

Number of states (States)	Trivial lower bounds (TrivL)	Improved lower bounds (ImprL)	Exact solution (Exact)	Improved upper bounds (ImprU)	Trivial upper bounds (TrivU)	Relative ratio (Ratio)
100	0.14378	0.18362	0.24946	0.24040	0.24954	1.71247
200	0.06657	0.08501	0.12466	0.11924	0.12469	1.50527
300	0.04580	0.05705	0.08313	0.07958	0.08314	1.46999
400	0.03455	0.04293	0.06118	0.05954	0.06237	1.47638
500	0.02868	0.03531	0.04904	0.04768	0.04991	1.52802

**Table 2.** Results for discounted CTMDPs with 25 actions and discount factor 2

(States)	(TrivL)	(ImprL)	(Exact)	(ImprU)	(TrivU)	(Ratio)
100	2.87161	3.49862	5.08380	5.19395	5.22686	1.38927
200	2.66808	3.34325	4.97368	5.07058	5.10530	1.41288
300	2.73046	3.37402	4.94414	5.03400	5.06653	1.40827
400	2.69198	3.32580	4.92630	5.01026	5.04692	1.39925
500	2.64740	3.31049	4.91676	5.00030	5.03700	1.41653

**Fig. 2.** CPU times for average reward criterion aggregation algorithms. The plotted results are obtained for random CTMDP models with 5 and 25 actions, an average reward of 5, an average sojourn time of  $1/5 \cdot |S|$ , a block size of mean  $10/3$  and  $\beta = 2$ .

ratio, which is calculated as the quotient of the difference between the trivial bounds and improved bounds. With our aggregation method we gain a relative improvement of round about 50% in comparison to the intuitive algorithm. A relative ratio of 2 would mean, that the span of upper and lower bounds is halved.

We discuss the applicability of the aggregation approach using a small closed central server queueing network where jobs can be alternatively routed to one of two peripheral servers. The queueing network is illustrated in Fig. 3. The central queue is a FCFS station with exponentially distributed service times with rate  $\mu = 2$ . After leaving the central station, a job enters one of the two peripheral stations. The choice of the station is a decision which can be made upon leaving the central station. Service times at the peripheral stations are distributed according to an Erlang 2 distribution with mean 1 at station  $Q_2$  and according to a hyperexponential distribution with parameters  $\mu_{31} = 4$ ,  $\mu_{32} = 1/4$  and  $p_{21} = 0.2$  for queue  $Q_3$ . Thus, both peripheral queues have the same mean service time.

The decision to choose  $Q_2$  or  $Q_3$  can be made according to the current population at the queues but cannot be based on the state of the service time distribution which is introduced in the Markov model to describe non-exponential times. If the whole model is interpreted as a CTMDP, then the optimal policy will consider the internal state of the service time and it might be better to choose a longer queue. E.g., if in  $Q_3$  a single customer is in phase 2, then the mean service time of this customer is 4, whereas the mean service time of a customer in phase 2 of  $Q_2$  is 0.5. Thus, in this situation  $Q_2$  is the better choice as long as it contains at most 3 customers more than  $Q_3$ .

If decision have to be made based on the population only, states with the same population in the queues are collected in one block. This implies that in our case blocks contain up to 4 states, if  $Q_2$  and  $Q_3$  are non-empty. Using aggregation a BMDP is computed. The robust and therefore pessimistic policy for this BMDP avoids routing into queue  $Q_2$  as long as the population difference between  $Q_3$  does not become too large because in the worst case, the service time distribution is in the slower phase. On the other hand, an optimistic policy tries to route customers to queue  $Q_3$  because the service time might be much smaller than in  $Q_2$  whenever the customer in service is in the fast phase (Fig. 3).

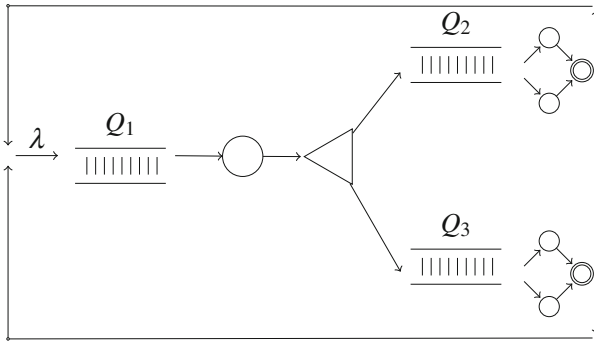


Fig. 3. Queueing network example.

**Table 3.** Results for the described example in the average and discounted case

(States)	(TrivL)	(ImprL)	(Exact)	(ImprU)	(TrivU)	(Ratio)
25 avg	0.6290	0.6743	1.0053	1.1535	1.1866	1.1636
61 avg	0.5756	0.6562	0.9912	1.1047	1.1779	1.3429
113 avg	0.6115	0.6707	0.8467	1.1384	1.2247	1.3111
25 disc	2.6608	2.9835	4.2256	5.4417	5.6080	1.1989
61 disc	2.5388	2.8003	4.3079	5.2663	5.4300	1.1724
113 disc	1.9375	2.2571	4.1899	5.4533	5.6438	1.1596

We analyze the model for the populations 3, 5 and 7 resulting in CTMDPs with 25, 61 and 113 states. The state spaces of the corresponding BMDPs contain 10, 21 and 36 states. For the results look at Table 3.

## 5 Conclusions

In this paper, we propose a state aggregation method for CTMDPs. In general, state aggregation enables one to reduce the number of states in a given CTMDP by deriving a bounded-parameter Markov model. The paper presents an improved aggregation approach to compute upper and lower reward bounds for CTMDPs for groups of similar states which are treated equally by a decision maker. It is shown that continuous time models can be efficiently aggregated when sojourn times in blocks are approximated using phase-type distributions. In the proposed method, one can refine the obtained bounds after an optimal policy has been computed. Comparing our results to established aggregation methods, we show that the proposed algorithm computes better bounds by an acceptable extra computational effort. Though the required CPU time is increased by a factor close to two, the difference between upper and lower bounds is reduced by nearly one half compared to a trivial aggregation algorithm.

The approach can be extended to refine the reward aggregation on the basis of stationary quantities in CTMDPs as presented in [3]. We have evaluated our aggregation method on randomly generated CTMDPs and a small queueing model. A special case that is in our opinion most interesting occurs when the states that are lumped into one are similar in behavior (with respect to transition and reward rates); we conjecture that in this case, our approach would show an even further improvement over the standard bounded-parameter aggregation approach. Furthermore, it is possible to improve the bounds for a fixed policy further by using the iterative bounding approach from [15].

## References

1. Abate, A., Češka, M., Kwiatkowska, M.: Approximate policy iteration for Markov decision processes via quantitative adaptive aggregations. In: Artho, C., Legay, A., Peled, D. (eds.) ATVA 2016. LNCS, vol. 9938, pp. 13–31. Springer, Cham (2016). doi:[10.1007/978-3-319-46520-3\\_2](https://doi.org/10.1007/978-3-319-46520-3_2)
2. Beutler, F.J., Ross, K.W.: Uniformization for semi-Markov decision processes under stationary policies. *J. Appl. Probability* **24**, 644–656 (1987)
3. Buchholz, P.: Bounding reward measures of Markov models using the Markov decision processes. *Numerical Lin. Alg. with Applic.* **18**(6), 919–930 (2011)
4. Buchholz, P., Dohndorf, I., Scheftelowitsch, D.: Analysis of Markov decision processes under parameter uncertainty. In: Reinecke, P., Di Marco, A. (eds.) EPEW 2017. LNCS, vol. 10497, pp. 3–18. Springer, Cham (2017). doi:[10.1007/978-3-319-66583-2\\_1](https://doi.org/10.1007/978-3-319-66583-2_1)
5. Buchholz, P., Hahn, E.M., Hermanns, H., Zhang, L.: Model checking algorithms for CTMDPs. In: Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, 14–20 July 2011, Proceedings, pp. 225–242 (2011)
6. Buchholz, P., Kriege, J., Felko, I.: Input Modeling with Phase-Type Distributions and Markov Models. SM. Springer, Cham (2014)
7. Courtois, P., Semal, P.: Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. *J. ACM* **31**(4), 804–825 (1984)
8. Dean, T.L., Givan, R., Leach, S.M.: Model reduction techniques for computing approximately optimal solutions for Markov decision processes. In: Geiger, D., Shenoy, P.P. (eds.) UAI 1997: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Brown University, Providence, Rhode Island, USA, 1–3 August 1997, pp. 124–131. Morgan Kaufmann (1997)
9. Franceschinis, G., Muntz, R.R.: Bounds for quasi-lumpable Markov chains. *Perform. Eval.* **20**(1–3), 223–243 (1994)
10. Givan, R., Dean, T.L., Greig, M.: Equivalence notions and model minimization in Markov decision processes. *Artif. Intell.* **147**(1–2), 163–223 (2003)
11. Givan, R., Leach, S.M., Dean, T.L.: Bounded-parameter Markov decision processes. *Artif. Intell.* **122**(1–2), 71–109 (2000)
12. Li, L., Walsh, T.J., Littman, M.L.: Towards a unified theory of state abstraction for MDPs. In: International Symposium on Artificial Intelligence and Mathematics, ISAIM 2006, Fort Lauderdale, Florida, USA, 4–6 January 2006 (2006)
13. Puterman, M.L.: Markov Decision Processes. Wiley, New York (2005)
14. Ren, Z., Krogh, B.: State aggregation in Markov decision processes. In: Proceedings of the 41st IEEE Conference on Decision and Control, vol. 4, pp. 3819–3824. IEEE (2002)
15. Semal, P.: Refinable bounds for large Markov chains. *IEEE Trans. Computers* **44**(10), 1216–1222 (1995)
16. Serfozo, R.F.: An equivalence between continuous and discrete time Markov decision processes. *Oper. Res.* **27**(3), 616–620 (1979)
17. Tewari, A., Bartlett, P.L.: Bounded parameter Markov decision processes with average reward criterion. In: Bshouty, N.H., Gentile, C. (eds.) COLT 2007. LNCS, vol. 4539, pp. 263–277. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-72927-3\\_20](https://doi.org/10.1007/978-3-540-72927-3_20)
18. Van Roy, B.: Performance loss bounds for approximate value iteration with state aggregation. *Math. Oper. Res.* **31**(2), 234–244 (2006)

Computer Performance Engineering

14th European Workshop, EPEW 2017, Berlin, Germany,

September 7-8, 2017, Proceedings

Reinecke, P.; Di Marco, A. (Eds.)

2017, XVI, 299 p. 103 illus., Softcover

ISBN: 978-3-319-66582-5