Dissimilarity for Linear Dynamical Systems

Giorgio Bacci[1(✉)], Giovanni Bacci[1], Kim Guldstrand Larsen[1], Giuseppe Squillace[2], Mirco Tribastone[2], Max Tschaikowski[1], and Andrea Vandin[3,4]

[1] Computer Science Department, Aalborg University, Aalborg, Denmark
{grbacci,giovbacci,klg,tschaikowski}@cs.aau.dk
[2] SysMA Unit, IMT Lucca, Lucca, Italy
{giuseppe.squillace,mirco.tribastone}@imtlucca.it
[3] Sant'Anna School of Advanced Studies, Pisa, Italy
andrea.vandin@santannapisa.it
[4] DTU Denmark, Copenhagen, Denmark

Abstract. We introduce backward dissimilarity (BD) for discrete-time linear dynamical systems (LDS), which relaxes existing notions of bisimulations by allowing for approximate comparisons. BD is an invariant property stating that the difference along the evolution of the dynamics governing two state variables is bounded by a constant, which we call dissimilarity. We demonstrate the applicability of BD in a simple case study and showcase its use concerning: (i) robust model comparison; (ii) approximate model reduction; and (iii) approximate data recovery. Our main technical contribution is a policy-iteration algorithm to compute BDs. Using a prototype implementation, we apply it to benchmarks from network science and discrete-time Markov chains and compare it against a related notion of bisimulation for linear control systems.

Keywords: linear dynamical systems · bisimulation · robustness

1 Introduction

Discrete-time linear dynamical systems (LDS) are a simple yet powerful mathematical tool in many disciplines: the equations of motion of discrete-time Markov chains are given by an LDS where each equation represents the probability of the process being in a given state at any time; the numerical solution of linear system of equations and systems of ordinary differential equations can be seen as an LDS whose dynamics provides an approximating sequence; in engineering sciences, LDS are a fundamental model for estimating and controlling the behavior of physical systems.

As with all dynamical models, notions that can establish whether state variables of an LDS can be formally related to each other can be used to answer questions about *comparison* and *reduction*. In the former, one wishes to identify certain relations between variables of two models; when these two models are two copies of the same one, establishing such a relation, e.g., a formal equivalence,

J. Hillston et al. (Eds.): QEST+FORMATS 2024, LNCS 14996, pp. 125–142, 2024.
https://doi.org/10.1007/978-3-031-68416-6_8

can lead to a reduction in the number of variables. This approach is very well understood when reasoning about such relations using bisimulation and related notions, which have been developed for a variety of models, e.g., in discrete [38] time, in continuous time [53] and in the deterministic [15], stochastic [40] and spatial [58] setting.

One of the major criticisms when dealing with models of real-world systems is that *exact* relations are too fragile [12,31,34,37]. This is due to the fact that imprecisions and uncertainties in model parameters and initial conditions likely violate the criteria underlying those relations. It is therefore a natural question to ask whether such criteria can be relaxed to allow for more robust notions that can relate systems *approximately*. Ideally, one would like to establish such relations with some guarantees about the dynamics of related variables.

In this paper, we present a novel notion of *dissimilarity* for LDS. We borrow the term from statistics, where a dissimilarity matrix between n objects is an $n \times n$ matrix that prescribes pairwise distances (not necessarily metric) between them. Here, we define the dissimilarity D between two state variables of an LDS in terms of an invariant stating that the absolute difference between the solutions is always less than or equal to D. Such invariant can be regarded as a relaxation of a strict notion of equivalence that can be traced back to exact lumpability [10] (also termed *backward bisimulation*) for Markov chains [46], which has been later generalized for various classes of ordinary differential equations [14,15,17]. Our notion of dissimilarity, which we call *backward dissimilarity* (BD), aims at capturing variables that have nearby solutions at all time points.

The main technical contributions are:

1. We show that backward dissimilarity is a proper relaxation of backward equivalence [15], as variables exhibit dissimilarity zero precisely when they are backward equivalent. To establish this result we needed to reduce the problem of finding backward dissimilarities to that of solving a number of (possibly inhomogeneous) *transportation problems* [22,26], ultimately resulting in a fixpoint characterization of backward dissimilarity.
2. We provide an effective algorithm to compute (tight) backward dissimilarities. Our solution, which is inspired by the policy-iteration algorithm for Markov decision processes [50], iteratively computes better strategies on how to compare the dynamics of each variable by performing only local updates.

Applications. Similarly to bisimulation metrics [3,5,23], BD can be used for *robust model comparison* and *approximate transfer of behavioural properties* (e.g., liveness and safety) from one variable to another, with the error bounded by the dissimilarity of variables. Thus, it is a valuable tool for model verification in general. Moreover, BD can be further applied for implementing *approximate model reduction* (via clustering) and used for *approximate data recovery*. In Sect. 6, we showcase these applications on a simple thermostat model from [24]. More specifically, we show (i) how to use BD for approximate model comparison; and (ii) how to recover missing readings of the temperature of one room from the temperature readings of adjacent rooms, compensating for malfunctioning sensors; and (iii) how to integrate BD and clustering techniques to perform approximate model reduction with certified error bounds.

2 Preliminaries and Notation

Let $\mathbb{N}$ and $\mathbb{R}$ denote the set of natural and real numbers, respectively. For $n \in \mathbb{N}$ we write $[n]$ for the set $\{1, \ldots, n\} \subseteq \mathbb{N}$. Let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ be an n-dimensional real-valued vector. We denote by $\|x\|_1 = \sum_i |x_i|$ the 1-norm of x, and by $\|x\|_\infty = \max_i |x_i|$ the infinite norm. For an $m \times n$-matrix $A \in \mathbb{R}^{m\times n}$ and vectors $x, y \in \mathbb{R}^n$, $Ax \in \mathbb{R}^m$ denotes the usual (row by column) matrix multiplication, A^T the transpose of A, $A_i \in \mathbb{R}^n$ the i-th row of A, and xy the dot product of two vectors (equivalently written as $x^T y$).

In this paper we are interested in discrete-time linear dynamical systems (LDS) of the form $x(t+1) = Ax(t) + b$, for some matrix $A \in \mathbb{R}^{n\times n}$, vector $b \in \mathbb{R}^n$, and $t \in \mathbb{N}$. We call $x = (x_1, \ldots, x_n)^T$ the vector of *state variables* and $x(0) \in \mathbb{R}^n$ the *initial condition*.

Example 1. The following is an example of LDS, described in matrix form.

$$x(t+1) = Ax(t) + b, \qquad \text{where } A = \begin{pmatrix} \frac{6}{10} & -\frac{2}{10} & \frac{2}{10} \\ \frac{2}{10} & \frac{8}{10} & 0 \\ \frac{2}{10} & 0 & \frac{8}{10} \end{pmatrix}, \text{ and } b = \begin{pmatrix} 0 \\ \frac{1}{5} \\ \frac{1}{5} \end{pmatrix}$$

Of these systems we want to study how the dynamics of the state variables vary in the presence of different forms of perturbations.

In this respect, it is relevant to understand what it means for two state variables to have equivalent dynamics. This concept is formally captured by the notion of *backward equivalence* (BE) [15] here cast to LDSs.

Definition 1 (Backward Equivalence). *Let $x(t+1) = Ax(t) + b$ be an LDS with n variables. An equivalence relation $R \subseteq [n] \times [n]$ is a* backward equivalence *if, for all $x \in \mathbb{R}^n$,*

$$\bigwedge_{(i,j)\in R} (x_i = x_j) \implies \bigwedge_{(i,j)\in R} \Big(A_i x + b_i = A_j x + b_j\Big). \tag{1}$$

Example 2. Let $x(t+1) = Ax(t) + b$ be as in Example 1. Then, the equivalence relation $R = \mathtt{id} \cup \{(2,3), (3,2)\}$, where $\mathtt{id} = \{(i,i) \mid 1 \leq i \leq 3\}$ denotes the identity relation, is a BE.

A BE relates state variables with identical solutions whenever these are initialized equally [15, Theorem 3]. Specifically, if R is a BE and $x_i(0) = x_j(0)$ for all $(i,j) \in R$, then $x_i(t) = x_j(t)$ for all $(i,j) \in R$ and $t > 0$. Typically, one is interested in finding the largest BE to reduce the size of the dynamical system by equating state variables with equivalent dynamics. The largest BE always exists and efficient algorithms are available to compute it [15,17].

Example 3. Consider the following perturbation of the LDS from Example 1, where one of the entries of A has been modified by subtracting an $0 < \epsilon < \frac{2}{10}$:

$$x(t+1) = A^\epsilon x(t) + b, \qquad \text{where } A^\epsilon = \begin{pmatrix} \frac{6}{10} & -\frac{2}{10} & \frac{2}{10} \\ \frac{2}{10} - \epsilon & \frac{8}{10} & 0 \\ \frac{2}{10} & 0 & \frac{8}{10} \end{pmatrix}, \text{ and } b = \begin{pmatrix} 0 \\ \frac{1}{5} \\ \frac{1}{5} \end{pmatrix}$$

With this, the relation R from Example 2 is not a BE for $x(t+1) = A^\epsilon x(t) + b$. This shows that BE is not robust w.r.t. perturbations on a dynamical system.

3 Backward Dissimilarity

In this section we introduce *backward dissimilarity* (BD) as a robust alternative to BE. Intuitively, a BD estimates the difference of the dynamics governing two state variables x_i, x_j of an LDS by establishing an invariant condition. If this condition is met in the initial state, it remains preserved throughout the entire temporal evolution of the LDS.

Definition 2 (Backward dissimilarity). *Let $x(t+1) = Ax(t) + b$ be an LDS. A symmetric matrix $D \in \mathbb{R}^{n\times n}_{\geq 0}$ is a* backward dissimilarity *for a non-empty set $I \subseteq \mathbb{R}^n$ of initial conditions if for all $x(0) \in I$ and $t \in \mathbb{N}$*

$$\bigwedge_{1\leq i,j\leq n} \Big(|x_i(t) - x_j(t)| \leq D_{ij}\Big) \implies \bigwedge_{1\leq i,j\leq n} \Big(|x_i(t+1) - x_j(t+1)| \leq D_{ij}\Big). \quad (2)$$

The definition of BD depends on a set I of initial conditions, which can be interpreted as the degree of variability that one allows on the choice of the initial conditions $x(0)$. In this respect, BD can also be used to estimate the difference between dynamics in the presence of perturbations of the initial conditions. Obviously, a BD represents a meaningful invariant of the dynamics only for initial conditions satisfying the premise in (2).

The next example showcases that not all LDSs admit meaningful BDs.

Example 4. Consider the system $\{x_1(t+1) = 2x_1(t) + 1, x_2(t+1) = x_2(t)\}$. Pick any symmetric $D \geq 0$ and assume that initial condition $x(0)$ satisfies the premise in (2), i.e., $|x_1(0) - x_2(0)| \leq D_{1,2}$. Then, the conclusion $|x_1(t) - x_2(t)| \leq D_{1,2}$ cannot hold for all $t \geq 1$ because $|x_1(t) - x_2(t)| \to \infty$ as $t \to \infty$.

Then, it is reasonable to work under the following assumption of boundedness, which provides sufficient conditions for the existence of meaningful BDs.

Working Assumption: *for the set $I \subseteq \mathbb{R}^n$ of initial conditions, there exists $\lambda > 0$ such that, for any $x(0) \in I$, $\|x(t)\|_\infty \leq \lambda$ for all $t \geq 0$.* (WA)

Note that our assumption is not requiring that all trajectories are bounded within the interval $[-\lambda, \lambda]$, but only when the initial conditions are in the set I. In this case the matrix with all entries $D_{ij} = 2\lambda$ is a meaningful BD.

Observe that (WA) is weaker than the notion of stability known from control theory [2] because it only requires that the reachable set is bounded with respect to the given set I of initial conditions. In practice, (WA) is satisfied by many models of physical systems from several domains due to properties related to the preservation of mass or energy [14,17]. Examples include:

- closed chemical systems [25,54], where each state variable represents the concentration of a chemical species, and closedness implies that the overall concentration across all species is constant (hence bounded) over time;
- electrical circuits (e.g., linear RLC networks), where Kirchhoff current and voltage laws prescribe preservation of energy such that the currents or voltages cannot exceed those of the external sources;

– models of closed economy, represented as a linear dynamical system owing to the fundamental work of Leontief [41], which postulates that all goods that are produced are consumed and that expenditure matches income overall.

To benefit the presentation, below we show a direct but loose estimation of λ.

Example 5. Consider the LDS $x(t+1) = A^{\epsilon}x(t) + b$ from Example 3 and fix $\epsilon = \frac{1}{10}$. By assuming the set of initial conditions $I = [-1, 1]^3$, a value for λ that satisfies (WA) is $\lambda = 2$. Indeed, as $A^{\epsilon}b = \frac{4}{5}b$ (i.e., b is an eigenvector of A^{ϵ}) and $\|A^{\epsilon}\|_{\infty} = 1$, the following inequality holds for all $t \in \mathbb{N}$ and $x(0) \in I$:

$$\begin{aligned}\|x(t)\|_{\infty} &= \|(A^{\epsilon})^t x(0) + \textstyle\sum_{i=0}^{t-1}(A^{\epsilon})^i b\|_{\infty} \leq \|A^{\epsilon}\|_{\infty}^t \|x(0)\|_{\infty} + \|\textstyle\sum_{i=0}^{t-1}(\frac{4}{5})^i b\|_{\infty} \\ &\leq 1 + \tfrac{1}{5}\textstyle\sum_{i=0}^{\infty}(\frac{4}{5})^i = 2.\end{aligned}$$

Example 6. Consider the LDS from Example 5 with same set of initial conditions. Then, the corresponding BD is $D = \begin{pmatrix} 0 & 2 & \frac{5}{3} \\ 2 & 0 & 1 \\ \frac{5}{3} & 1 & 0 \end{pmatrix}$. The coefficients of D have been found by a suitable estimation of the differences in the dynamics. For example, the dissimilarity $D_{2,3} = 1$ between the dynamics of x_2 and x_3 can be understood by looking at the following inequalities, with $\lambda = 2$ (*cf.* Example 5):

$$\begin{aligned}|A_2 x(t) - A_3 x(t)| &= |\tfrac{1}{10}(x_1(t) - x_1(t)) + \tfrac{8}{10}(x_2(t) - x_3(t)) + \tfrac{1}{10}\, x_1(t)| \\ &\leq \tfrac{8}{10}|x_2(t) - x_3(t)| + \tfrac{1}{10}\,|x_1(t)| \\ &\leq \tfrac{8}{10}|x_2(t) - x_3(t)| + \tfrac{1}{10}\lambda.\end{aligned}$$

which show that $|x_2(t) - x_3(t)| \leq 1$ implies $|x_2(t+1) - x_3(t+1)| \leq 1$.

Our method to find BDs follows the technique outlined in Example 6: factorizing the difference of the solutions at time $t+1$ in terms of the differences at time t. Such a factorization is always possible and can be found as the solution of a linear program, namely, the *(inhomogeneous) transportation problem* [22,26].

Definition 3 (Transportation problem). *For two vectors $c, d \in \mathbb{R}^n$ and cost matrix $D \in \mathbb{R}^{n \times n}$ we define $\mathcal{T}_{\lambda}(D)(c, d)$ as the optimal value of the following linear program*

$$\begin{aligned}\mathcal{T}_{\lambda}(D)(c,d) = \min_{s \in \mathbb{R}^n, \bar{s} \in \mathbb{R}^n, \omega \in \mathbb{R}^{n \times n}} &\Big[\lambda \textstyle\sum_i (s_i + \bar{s}_i) + \sum_{i,j} D_{ij}\omega_{ij}\Big] \\ \textit{subject to } & \textstyle\sum_j \omega_{ij} + s_i = c_i^+ + d_i^- \qquad i = 1 \dots n \\ & \textstyle\sum_i \omega_{ij} + \bar{s}_j = c_j^- + d_j^+ \qquad j = 1 \dots n \\ & \omega_{ij} \geq 0,\ s_i \geq 0,\ \bar{s}_j \geq 0 \qquad i,j = 1 \dots n\end{aligned}$$

where $c = c^+ - c^-$ and $d = d^+ - d^-$ are decomposed in their positive and negative parts. Intuitively, the decision variables $w_{i,j}$ represent the amount of goods transported from each supplier ($c_i^+ + d_i^-$) to each destination ($c_j^- + d_j^+$), subject to the standard supply and demand constraints, and non-negativity constraints *on the decision variables. The (slack) variables s_i and $\bar{s}_j$ account for possible mismatch between supply and demand. A feasible assignment of the variables $(s, \bar{s}, \omega)$ is called* transportation schedule*, and we call it* perfect *when $s, \bar{s} = 0$.*

We denote by $\Gamma(c,d)$ the polytope containing all transportation schedules for (c,d), and by $\Gamma_V(c,d)$ the (finite) subset of its vertices.

Example 7. We can reinterpret the factorization in Example 6 as an instance of a transportation schedule. Below, we show the (nonzero) assignments of the variables $s, \bar{s}, \omega$, corresponding to a transportation schedule as in Definition 3:

$$\begin{aligned} A_2x(t) &= \overbrace{\tfrac{1}{10}}^{c_1} x_1(t) + \overbrace{\tfrac{8}{10}}^{c_2} x_2(t) \\ &= \underbrace{\tfrac{1}{10}}_{\omega_{1,1}} x_1(t) + \underbrace{\tfrac{8}{10}}_{\omega_{2,3}} x_2(t) \end{aligned} \qquad \begin{aligned} A_3x(t) &= \overbrace{\tfrac{2}{10}}^{d_1} x_1(t) + \overbrace{\tfrac{8}{10}}^{d_3} x_3(t) \\ &= (\underbrace{\tfrac{1}{10}}_{\omega_{1,1}} + \underbrace{\tfrac{1}{10}}_{\bar{s}_1}) x_1(t) + \underbrace{\tfrac{8}{10}}_{\omega_{2,3}} x_3(t) \end{aligned}$$

Then, the factorization from Example 6 matches the expression used as objective function in the transportation problem as shown below for $D_{ij} = |x_i(t) - x_j(t)|$

$$\lambda \textstyle\sum_i (s_i + \bar{s}_i) + \sum_{i,j} D_{ij}\omega_{ij} = \lambda \frac{1}{10} + \frac{8}{10}|x_2(t) - x_3(t)|.$$

BDs are clearly not unique and different factorizations might give better estimations on the difference of the dynamics. The transportation problem finds the tightest one for any two given state variables. By applying this idea to each pair of state variables of the LDS $x(t+1) = Ax(t) + b$, we can define the operator:

$$\Delta_\lambda(D)_{ij} = \mathcal{T}_\lambda(D)(A_i, A_j) + |b_i - b_j|,$$

that, given an estimation $D \in \mathbb{R}^{n\times n}_{\geq 0}$ of the pair-wise differences of the dynamics at time t, provides an updated estimate $\Delta_\lambda(D) \in \mathbb{R}^{n\times n}_{\geq 0}$ for time $t+1$. In the following, we just write Δ when λ is clear from the context (or when the discussion is generic on the value of $\lambda > 0$).

Lemma 1. *The operator Δ is well-defined on the complete lattice $([0,\infty]^{n\times n}, \sqsubseteq)$ and monotone w.r.t. $\sqsubseteq$, where $D \sqsubseteq D'$ iff $D_{ij} \leq D'_{ij}$ for all $1 \leq i, j \leq n$.*

Our first major result ensures that the pre-fixpoints of Δ are BDs for the given set of initial conditions satisfying our working assumptions.

Theorem 1. *Let $x(t+1) = Ax(t) + b$ be an LDS, $I \subseteq \mathbb{R}^n$, and $\lambda > 0$ satisfying assumption* (WA)*. Then, any $D \in \mathbb{R}^{n\times n}_{\geq 0}$ such that $\Delta_\lambda(D) \sqsubseteq D$ is a BD for I.*

Remark 1 (Finite time horizon). Using (WA) makes sense only when one is interested in establishing dissimilarities among trajectories over an infinite time horizon. However, if one is just interested in trajectories over a finite time horizon $[0,T]$, (WA) can be dropped by choosing a λ that bounds the dynamics of the LDS up to time T. In this case, pre-fixpoints of Δ_λ are dissimilarities satisfying Eq. (2) only for $t \in [0, T-1]$.

As $\varDelta_\lambda$ is monotonic, by Knaster-Tarski's fixpoint theorem, it has a least fixpoint, hereafter denoted by δ_λ (or simply δ). The next result, in combination with Theorem 1, tells us that BD is indeed a robust extension of BE.

Theorem 2. *Let R be the greatest BE. Then, $\delta_{i,j} = 0$ iff $(i, j) \in R$.*

4 Computation of Backward Dissimilarity

In this section we discuss how to compute pre-fixpoints of $\varDelta_\lambda$. To this end we provide a procedure which is inspired by the *simple policy iteration* algorithm for Markov decision processes [50, Chapter 6.4].

We will use two key concepts, namely that of *transport policy* and *policy dissimilarity*. A transport policy, similarly to a policy for MDPs, is a map that assigns a factorization strategy (i.e., a transportation schedule) with each pair of state variables. A policy dissimilarity is the smallest dissimilarity that can be proved by using the given factorization strategy.

Definition 4. *A* transport policy *π for $x(t + 1) = Ax(t) + b$ is a map that assigns to each pair of indices (i, j) a transportation schedule $\pi(i, j) \in \varGamma(A_i, A_j)$. If $\pi(i, j) \in \varGamma_V(A_i, A_j)$ for all (i, j), π is referred to as* vertex *transport policy.*

We denote by $\varPi(A, b)$ (resp. $\varPi_V(A, b)$) the set of transport policies (resp. vertex transport policies) for $x(t + 1) = Ax(t) + b$. Note that $\varPi_V(A, b)$ is *finite*.

Given a transport policy $\pi \in \varPi(A, b)$, we define the operator

$$\varDelta^\pi_\lambda(D)_{ij} = \big(\lambda \textstyle\sum_h (s_h + \bar{s}_h) + \sum_{h,k} D_{hk}\omega_{h,k}\big) + |b_i - b_j| \qquad (3)$$

where $\pi(i, j) = (s, \bar{s}, \omega)$. In contrast to $\varDelta_\lambda$ from Theorem 1, given an estimate $D \in \mathbb{R}^{n\times n}_{\geq 0}$ of the pair-wise differences of the dynamics at time t, $\varDelta^\pi_\lambda(D)$ computes an updated estimate for time $t+1$ via the factorization strategy described by π.

As in Lemma 1, one can show that $\varDelta^\pi_\lambda$ is monotone in $([0, \infty]^{n\times n}, \sqsubseteq)$. Thus, by Knaster-Tarski's fixpoint theorem $\varDelta^\pi_\lambda$ has least fixed point.

Definition 5. *Let $\pi \in \varPi(A, b)$ and $\lambda > 0$. The* policy dissimilarity *for π, denoted δ^π_λ, is the least fixed point of $\varDelta^\pi_\lambda$.*

The following result states that for any transport policy $\pi \in \varPi(A, b)$, the pre-fixpoints of $\varDelta^\pi_\lambda$ are also pre-fixpoints of $\varDelta_\lambda$. Thus, by Theorem 1, δ^π_λ is a BD for a set of initial conditions $I \subseteq \mathbb{R}^n$ and $\lambda > 0$ satisfying assumption (WA).

Proposition 1. *Let $\pi \in \varPi(A, b)$. Then $\varDelta^\pi_\lambda(D) \sqsubseteq D$ implies $\varDelta_\lambda(D) \sqsubseteq D$.*

With this in hand, we show that the least fixed point of $\varDelta_\lambda$ is the minimum policy dissimilarity ranging over all vertex transport policies.

Theorem 3. $\delta_\lambda = \min\{\delta^\pi_\lambda \mid \pi \in \varPi_V(A, b)\}$.

As in the classical policy iteration algorithm, our procedure starts from some suitable transport policy. At each iteration, the current policy is replaced by one

```
SimplePolicyIteration(A, b, λ, R)
1  let D_ij = 0 if (i, j) ∈ R and D_ij = 1 if (i, j) ∉ R.
2  for each (i, j) ∈ [n] × [n]
3      if (i, j) ∈ R
4          π[i, j] = k_λ(D)(A_i, A_j)
5      else π[i, j] = (A_i^+ + A_j^-, A_i^- + A_j^+, 0)
6  while ∃(i, j). Δ_λ(δ_λ^π)_ij < (δ_λ^π)_ij
7      π[i, j] = k_λ(δ^π)(A_i, A_j)          // update the policy as in Lemma 2
8  return δ_λ^π                              // δ_λ^π is a fixed point of Δ_λ
```

Fig. 1. Algorithm for computation of dissimilarity invariants.

having a smaller policy dissimilarity. The algorithm terminates when such an update no longer yields a better dissimilarity (*cf.* 6–8 in Fig. 1).

To update the policy we use a map k_λ that, given $D \in \mathbb{R}^{n \times n}_{\geq 0}$ and $c, d \in \mathbb{R}^n$, returns an optimal vertex transportation schedule in the sense of Definition 3:

$$k_\lambda(D)(c, d) \in \operatorname*{arg\,min}_{(s, \bar{s}, \omega) \in \Gamma_V(c,d)} \lambda \textstyle\sum_i (s_i + \bar{s}_i) + \sum_{i,j} D_{ij} \omega_{ij}. \tag{4}$$

The following lemma shows how k_λ can be used to update the current policy at one selected pair of indices (i, j) to obtain a smaller dissimilarity.

Lemma 2. *Let $\pi \in \Pi(A, b)$. If $\Delta_\lambda(\delta_\lambda^\pi)_{ij} < (\delta_\lambda^\pi)_{ij}$ for some i, j, then $\delta_\lambda^{\bar{\pi}} \sqsubset \delta_\lambda^\pi$ where $\bar{\pi} = \pi[(i, j)/k_\lambda(\delta^\pi)(A_i, A_j)]$.*

Now we are only left to explain how our algorithm constructs the initial transport policy (*cf.* lines 2–5 in Fig. 1). The easiest transport policy π one can think of is the one that assigns to each pair of indices (i, j) the transportation schedule $\pi(i, j) = (A_i^+ + A_j^-, A_i^- + A_j^+, \mathbf{0})$. However, the smaller is the initial policy dissimilarity, the faster the algorithm will converge. Thus we provided the algorithm with the option to pass a BE R as input, and construct the initial policy π in such a way that $(\delta_\lambda^\pi)_{ij} = 0$ for each $(i, j) \in R$. Since computing BEs can be done efficiently by using *ad hoc* techniques [4, 17], allowing for this option might improve the overall execution time of the algorithm. By default, in our implementation we pass the largest BE for the considered LDS.

Theorem 4 (Computation of BD). *Let R be a BE for $x(t + 1) = Ax(t) + b$ and $\lambda > 0$. The procedure* SimplePolicyIteration(A, b, λ, R) *terminates and its return value D satisfies $\Delta_\lambda(D) = D$ and $R \subseteq \{(i, j) \mid D_{ij} = 0\}$.*

Complexity Analysis. Let n be the number of state variables of the LDS. The transportation problem of Definition 3 is an instance of the uncapacitated min-cost network flow problem, which can be solved by employing Orlin's algorithm in $O(n^3 \log n)$ time [47]. Thus, the construction of the initial policy takes $O(|R|\, n^3 \log n)$ time to solve $|R|$ transportation problems. Each iteration requires one to compute the policy dissimilarity δ_λ^π, solve at most n^2 transportation problems to find the pair (i, j) to update, and solve one last transportation problem to perform the policy update.

δ_λ^π can be computed by solving the following linear program in n^2 variables:

$$\begin{aligned} &\min_{D} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} \\ &\text{subject to } \Delta_\lambda^\pi(D)_{ij} = D_{ij} && (i = 1\,..\,n, j = 1\,..\,n) \\ &D_{ij} \geq 0 && (i = 1\,..\,n, j = 1\,..\,n) \end{aligned}$$

The above LP can be solved by using the interior point method in $O(n^{4.5} \log(n/\alpha))$ time [59] , where α is the relative accuracy (in our implementation $\alpha = 10^{-6}$). As a consequence, the worst-case runtime of a single iteration of the algorithm is $O(n^{4.5} \log(n/\alpha) + n^5 \log n) = O(n^5 \log n)$. Similarly to the policy iteration algorithm for MDPs [21], the number of iterations is, in the worst case, exponential in the number of state variables. However, in Sect. 7 our experiments show that, in practice, the algorithm converges after few iterations.

5 Related Work

For Markov chains, bisimilarity distances [23] and near-lumpability [11,27] generalize probabilistic bisimulation [40], a.k.a. ordinary lumpability [10]. BD, instead, generalizes exact lumpability, which is complementary to ordinary lumpability [10,45]. Moreover, BD applies to arbitrary LDS and not just Markov chains. In the field of nonlinear dynamical systems [36,52], algorithms for establishing algebraic invariants are available for polynomial dynamical systems [7,8,17,30]. However, the aforementioned invariants describe Lebesgue null sets, thus they belong to the class of exact relations as opposed to the robust relations sought by BD. Approximate extensions of [8,17] have been considered [9,19,57] but require an initial condition and/or finite-time horizons, while the computation of expressive semi-algebraic invariants [43,52] is challenging. Instead, common over-approximation approaches where the reachable set is over-approximated by geometrically convenient objects such as zonotopes [1,29] or Taylor models [20] are complementary to the our approach because they provide absolute estimates of the reachtube. BD, instead, provides estimates of the form $|x - y| \leq D$ which are not concerned with x or y *per se* but with their relative difference $x - y$. By extending the LDS with auxiliary variables $z_{ij} = x_i - x_j$, we remark that [1,20] can be used to over-approximate differences of variables. However, the underlying over-approximations would still depend on the size of the set of initial conditions, e.g., the diameter, rather than differences $x_i - x_j$ only. In light of the fact that BDs are expressible as pre-fixpoints of a monotone operator, the lattice-theoretic PDR-inspired algorithms proposed in [39] may provide an alternative solution to the policy-iteration algorithm considered in this paper and are subject of future work.

A comparable, but different approach is bisimulation of linear control systems (LCS) [32,34]. LCS bisimulation can be seen as a general case of BD because it allows bounding the dynamics of LCS with respect to an arbitrary output map. Tailored to the case of estimating a BD, it leads to the computation of a Lyapunov function [51] that bounds, from below and from above, the set of points satisfying the BD constraints $|x_i - x_j| \leq D_{ij}$, with D_{ij} being the dissimilarity computed by BD for variables i and j. In the rest of the section we formalise how BD compares to LCS bisimulation.

Bisimulation of LCSs. In its more general form, bisimulation for linear control systems (LCSs) considers the problem of comparing two linear control systems in the form $x^i = A^i x^i + B^i u^i, y^i = C^i x^i$, with $i = 1, 2$ where x^i are state variables, y^i are output variables, and u^i are controls (with matrices A^i, B^i and C^i of appropriate sizes [32]). Here we review a definition of LCS bisimulation tailored to the purpose of computing dissimilarities between two variables of an LDS. In particular, we do not consider controls; this implies that we assume LDS with no affine term, i.e., $b = 0$ from now on.

Definition 6 (LCS Bisimulation). *Given an LDS $x(t+1) = Ax(t)$ with $A \in \mathbb{R}^{n\times n}$, we call $\mathcal{B}^{ij} \subseteq \mathbb{R}^n$ an LCS bisimulation for threshold $\delta > 0$ and pair $i \neq j$ if whenever $x \in \mathcal{B}^{ij}$, then $|x_i - x_j| \leq \delta$ for all $Ax \in \mathcal{B}^{ij}$.*

Bisimulations can be constructed by means of bisimulation functions.

Definition 7 (Bisimulation function). *Given an LDS $x(t+1) = Ax(t)$ with $A \in \mathbb{R}^{n\times n}$ and a pair $i \neq j$, a function $V^{ij} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is called bisimulation function if for every $x \in \mathbb{R}^n$ we have*

$$a)\ V^{ij}(x) \geq V^{ij}(Ax) \qquad \text{and} \qquad b)\ V^{ij}(x) \geq |x_i - x_j|.$$

It can be easily seen that $\{x \in \mathbb{R}^n \mid V^{ij}(x) \leq \delta\}$ is an LCS bisimulation with respect to any threshold $\delta > 0$ [32]. Indeed, the first condition essentially requires that V^{ij} is decreasing along the solutions of the LDS, while the second one demands that $V^{ij}(x)$ bounds $|x_i - x_j|$.

While the computation of bisimulation functions V^{ij} is in general challenging [33], the case where V^{ij} is quadratic can be addressed by solving a system of linear matrix inequalities [6,32]. Building on this, the following can be proven.

Theorem 5. *Let D be a symmetric non-negative matrix and $x(t+1) = Ax(t)$ asymptotically stable at $x = 0$. There is an $r > 0$ such that for any $x(0) \in \mathbb{R}^n$ satisfying $\|x(0)\|_2 = r$ and $\bigwedge_{k\neq l} |x_k(0) - x_l(0)| \leq D_{kl}$, for any pair $i \neq j$ one can compute in polynomial time lower and upper estimates of the quadratic bisimulation at $x(0)$, that is, $\underline{L}_{ij} \leq V^{ij}(x(0)) \leq \overline{L}_{ij}$.*

The above result ensures in particular that a BD D provides tighter estimates than the quadratic bisimulation V^{ij} if $D_{ij} \leq \underline{L}_{ij}$.

6 Applications: The Thermostat Case Study

We showcase possible applications of BDs on a simple thermostat model from [24] but applications in performance modelling [55,56,60], engineering [13] and biology [18,28] are also possible. The room temperature depends on neighboring rooms, outdoor temperature, and room-specific air conditioning settings,

	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$
$\bar{x}_1$	0	0.71	0.56
$\bar{x}_2$	0.71	0	0.9
$\bar{x}_3$	0.56	0.9	0

	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{y}_1$	$\bar{y}_2$
$\bar{x}_1$	0	0.71	0.56	0.15	0.7
$\bar{x}_2$	0.71	0	0.9	0.84	0.27
$\bar{x}_3$	0.56	0.9	0	0.71	1.2
$\bar{y}_1$	0.15	0.84	0.71	0	0.57
$\bar{y}_2$	0.7	0.27	1.2	0.57	0

Fig. 2. (Left) BD for the thermostat example. (Right) BD for the union thermostat.

controlled by a linear feedback control [2]. Formally, we consider the LDS $x(t+1) = Ax(t) + b + u(t)$ where $x_i(t)$ is the temperature in room $i = 1, 2, 3$ at time t, b accounts for the outside temperate, while $u_i(t)$ is the control input applied in room i at time t. The matrix A and vector b are as shown below[1]

$$A = \begin{pmatrix} 0.9910 & 0.0050 & 0 \\ 0.0050 & 0.9830 & 0.0055 \\ 0 & 0.0055 & 0.9915 \end{pmatrix}, \qquad b = \begin{pmatrix} 1.6 \\ 1.2 \\ 1.6 \end{pmatrix}. \tag{5}$$

The rest of the section demonstrates how to compute a BD for the above LDS and showcase how it can be used to perform approximate model reduction.

Computing Backward Dissimilarities. In the case when $u(t) = 0$ for all $t \geq 0$ (i.e., all thermostats are kept off) the temperature converges to 4 °C in all rooms. The constant control input u^*, that ensures all three room attain a temperature of 20 °C, can be obtained by solving $x^* = f(x^*, u^*)$, where $f(x, u) = Ax + b + u$ and $x^* = (20, 20, 20)^T$.

An estimation of λ can be achieved as in Example 5 by noting that $\|A\|_\infty < 1$. However, we point out that a sharper estimation of λ is often possible when a reference point x^* is given. Indeed, we can perform a simple change of variable $\bar{x} = x - x^*$ and $\bar{u} = u - u^*$ corresponding to a shifting of the graph. Since $\bar{x}(t+1) + x^* = f(\bar{x}(t) + x^*, \bar{u}(t) + u^*)$ the dynamics $\bar{x}(t+1) = g(\bar{x}(t), \bar{u}(t))$ of the shifted system can be obtained via

$$g(\bar{x}(t), \bar{u}(t)) = \bar{x}(t+1) = f(\bar{x}(t) + x^*, \bar{u}(t) + u^*) - x^* = A\bar{x}(t) + \bar{u}(t), \tag{6}$$

where the last identity uses $u^* = (I - A)x^* - b$. Observing that g has no additive constant and $\|A\|_\infty < 1$, we can set $\bar{u}(t) = 0$ and $\lambda = 5$, provided that $\|\bar{x}(0)\|_\infty = \|x(0) - x^*\|_\infty \leq 5$. With this in place, for $x^* = (20, 20, 20)^T$ we obtain the BD D depicted in Fig. 2 (left), where we see that the variables x_1 and x_3 are the least dissimilar ($D_{13} = 0.56$).

Approximate Data Recovery. Assume that the thermometer in room 1 is malfunctioning. We can exploit the BD D in Fig. 2 (left) to recover good estimates for the missing readings $x_1(t)$ from those of the other rooms, because

$$\max(x_2(t) - D_{12}, x_3(t) - D_{13}) \leq x_1(t) \leq \min(x_2(t) + D_{12}, x_3(t) + D_{13})$$

[1] The matrix A was obtained by discretizing the original model [24, Eq. 8] with time step 10^{-2} and by additionally perturbing the matrix entries in the order of 10^{-3}, to replicate a typical real-case scenario where fragile model symmetries do not occur.

ensuring that $x_1(t)$ is at most 0.56 °C far from $x_3(t)$, for all $t \in \mathbb{N}$.

Approximate Model Reduction. The thermostat LDS defined in (5) has no model symmetries, thus exact model reduction technique based on BE [15] does not apply in this scenario. In what follows, we show how to exploit the BD D in Fig. 2 (left) to perform approximate reduction of the thermostat LDS. Specifically, we proceed in two steps:(1) we perform hierarchical clustering using D as underlying distance to obtain a partitioning of the rooms which is then used to construct a reduced LDS that approximates the dynamics of the thermostat LDS;(2) finally, we employ BD to compare the approximated system against the original one—the smaller the measured dissimilarities between corresponding variables the higher the quality of the approximated model.

By performing hierarchical clustering using the BD D as underlying distance, we obtain the partitioning $\mathcal{H} = \{\{1,3\},\{2\}\}$. Using the reduction algorithm from [15] for $\mathcal{H}$, we obtain the LDS $\bar{y}(t+1) = B\bar{y}(t)$ where $B = \left(\begin{smallmatrix} 0.9910 & 0.0050 \\ 0.0105 & 0.9830 \end{smallmatrix}\right)$ Informally, this LDS approximates the original dynamics by lumping $\bar{x}_1$ and $\bar{x}_3$.

The quality of the reduced model is evaluated by providing a backward dissimilarity for the "union" LDS $(\bar{x}(t+1), \bar{y}(t+1)) = (A\bar{x}(t), B\bar{y}(t))$ and $\lambda = 5$[2] provided that $\|(x(0), y(0)) - (x^*, y^*)\|_\infty \leq 5$ for $y^* = (20, 20)^T$. The resulting BD is depicted in Fig. 2 (right). We can see that $\bar{y}_2$ is a good approximation of $\bar{x}_2$ scoring a dissimilarity of 0.27 °C. Likewise, $\bar{y}_1$, which represents the aggregation of $\bar{x}_1$ and $\bar{x}_3$, scores a dissimilarity of 0.15 °C with $\bar{x}_1$ and 0.71 °C with $\bar{x}_3$.

The approximated LDS $\bar{y}(t+1) = B\bar{y}(t)$ models the intended variables aggregation. Notably, by performing hierarchical clustering of the variables in the union LDS using the BD in Fig. 2 (right) as underlying distance, we retrieve the proposed variable lumping.

7 Experimental Comparison with LCS Bisimulation

This section numerically compares the quality of the dissimilarity estimations against those obtained by LCS bisimulation showing that BD provides tighter bounds on a number of case studies taken from network science, for the control of network dynamics [44], and on a selection of discrete-time Markov chains from the QComp repository [35].

Set-Up. We consider two classes: discrete-time Markov chains (DTMCs) from the QComp repository [35] and controlled dynamical networks from the Netzschleuder network catalogue [48]. For a DTMC with transition probability matrix A^T, we simply considered the LDS $x(t+1) = Ax(t)$, which gives the equation of motion of the transient probability distribution. Because A^T is a row stochastic matrix, we can pick $\lambda = 1$ for any measure $x(0)$.

With regard to controlled dynamical networks, we use a standard set-up from network science [44]. Specifically, given an adjacency matrix A, we consider the LDS $x(t+1) = (A - K)x(t)$ where K implements a control strategy that steers the dynamics toward a desired target state (chosen to be $x = 0$ without loss

[2] The choice of λ is justified by $\max(\|A\|_\infty, \|B\|_\infty) < 1$.

Table 1. Comparison BD - LCS bisimulation. Column n provides the number of variables in the model, while *Time* and *Iter.* are the runtime in seconds and the number of iterations performed by the two techniques. For BD, we also provide (column *0*) the number (nontrivial) BE-equivalent variables. The actual *DOM* models are Correa2013d, DMasure1934a, and Mwamende2009a. Parameters for the (parametric) *DTMC* models are: *haddad*, $\texttt{n} = 5$, $\texttt{p} = 0.6$; *brp*, $\texttt{n} = 1$, $\texttt{max} = 3$; *herman*, $\texttt{n} = 5$.

Model		*LCS bisim.*		*BD*			*Ratios LCS bisim./BD*			
Name	n	*Time*	*Iter.*	*Time*	*Iter.*	*0*	*Min*	*Avg low*	*Avg up*	*Max*
DTMCs https://qcomp.org/benchmarks/										
haddad	11	11.8	12	3.0	3	2	4.60E–2	4.79E–1	4.99E+0	7.37E+1
brp	25	866.9	10	13.8	3	2	7.71E–2	2.71E+2	3.51E+3	3.07E+5
herman	32	4095.6	8	61.3	2	136	2.41E–1	5.50E–1	2.85E+0	5.47E+0
MOVIEGALAXIES https://networks.skewed.de/net/moviegalaxies										
328	9	2.3	9	4.2	4	0	1.58E+0	6.37E+0	8.57E+0	8.19E+1
293	13	17.2	9	15.8	4	0	1.73E+0	4.29E+0	5.81E+0	2.54E+1
347	15	23.6	9	24.4	4	9	2.64E+0	1.48E+1	2.02E+1	2.85E+2
17	22	252.0	9	146.4	5	16	1.41E+0	6.00E+0	8.09E+0	6.23E+1
33	25	407.1	9	904.8	6	3	1.06E+0	3.41E+0	4.66E+0	1.91E+1
804	29	761.6	9	804.6	5	15	8.13E–1	4.38E+0	5.93E+0	6.08E+1
DOM https://networks.skewed.de/net/dom										
Cor	6	0.8	8	2.5	4	0	2.86E+0	2.17E+1	3.31E+1	1.89E+2
DMas	13	16.1	10	6.9	4	3	1.92E–1	5.26E–1	1.49E+0	5.46E+0
Mwa	19	108.5	12	18.3	4	0	2.27E–1	2.45E–1	6.06E–1	1.41E+0
AMBASSADOR https://networks.skewed.de/net/ambassador										
2000	16	26.3	9	34.7	5	2	3.41E+0	9.33E+0	1.27E+1	3.10E+1
2005	16	28.9	10	3.1	2	92	2.57E+0	2.57E+0	3.49E+0	3.49E+0
1990 1994	16	25.8	9	27.6	5	2	2.13E+0	5.49E+0	7.49E+0	1.45E+1

of generality), computed by the celebrated LQR formula by solving a so-called Riccati matrix equation [6,42]. For each network, we computed K numerically by using the MATLAB command `idare`. By using this setting we can consistently ensure asymptotic stability of the resulting LDS.

Results. The results of our comparison of BD and LCS bisimulation are shown in Table 1. To ensure a representative evaluation of the bounds, benchmarks were drawn from four different classes of networks. For each model we report the number of variables; the runtimes and the number of iterations for the computation of the LCS bisimulation and for our BD policy-iteration algorithm, implemented on a prototype which uses ERODE [16] to compute BEs, and Google's OR-Tools library [49] to solve the linear programs involved in Algorithm 1. All experiments were run on a common laptop.

We compare the dissimilarity estimations of BD and LCS bisimulation as follows. For each pair of variable indices (i, j), we build the ratios involving the interval estimation $[\underline{L}_{ij}; \bar{L}_{ij}]$ of the LCS bisimulation (*cf.* Lemma 5) and the BD D_{ij}, and report the following statistics:

minimum ratio: $\min\left\{\underline{L}_{ij}/D_{ij} \mid (i,j) \text{ s.t. } D_{ij} \neq 0\right\}$;
average lower bound ratio: $\text{avg}\left\{\underline{L}_{ij}/D_{ij} \mid (i,j) \text{ s.t. } D_{ij} \neq 0\right\}$;
average upper bound ratio: $\text{avg}\left\{\bar{L}_{ij}/D_{ij} \mid (i,j) \text{ s.t. } D_{ij} \neq 0\right\}$;
maximum ratio: $\max\left\{\bar{L}_{ij}/D_{ij} \mid (i,j) \text{ s.t. } D_{ij} \neq 0\right\}$;

where all the above sets do not include elements such that $D_{ij} = 0$, which are related by BE (*cf.* Theorem 2). The number of distinct BE pairs is reported in the table. These statistics summarize the variability of the relative tightness of the two dissimilarity estimations for each pair of variables—ratios larger than one indicate tighter BD estimations. We observe:

1. In 9 of out the 15 models, the minimum ratio was larger than one: this indicates that across all entries the BD estimation was consistently tighter than the LCS bisimulation.
2. In the models where the minimum ratio was smaller than one, LCS bisimulation is not consistently tighter than BD. This is because the corresponding maximum ratio is larger than one, even considerably so (e.g., up to five orders of magnitude for the model *brp*).
3. The average ratios tend to favor BD. Interestingly, in one model where the average ratios do favor LCS bisimulation (*Mwa*), Algorithm 1 was one order of magnitude faster.
4. The relative runtime behavior seems to be model-dependent. In the case where the BD runtime is most significantly larger (model *33*), the resulting BD is consistently tighter across all pairs. The largest runtime ($\geq$ 30 min) is reported for the model *herman*.
5. LCS bisimulation was not able to detect BE (i.e., no model returned $\underline{L}_{ij} = 0$ for $i \neq j$).
6. Across all the considered models, the BD-policy iteration algorithm terminated after at most 6 iterations. This suggests that, despite the exponential worst-case runtime upper bound, the method performs well in practice.

8 Conclusion

We presented backward dissimilarity (BD), a robust relation between variables of a linear dynamical system which bounds the differences between their solutions at all time points. A fixed-point characterization of backward dissimilarity allowed for the development of a policy-iteration algorithm to compute BDs. Although each iteration of the algorithm has polynomial time complexity, in the worst-case the computation may require exponentially many iterations as a function of the number of variables. Nevertheless, experimental results showed that the actual number of iterations is typically limited to a few. Compared to an alternative method based on bisimulation for linear control systems, BD offers tighter estimates and similar runtimes in most cases.

Acknowledgments. This work was partially supported by Poul Due Jensen Foundation grant no. 883901; the S40S Villum Investigator Grant nr. 37819 from Villum Fonden, the Fsc regional Tuscan project AUTOXAI2 J53D21003810008, the project SERICS (PE00000014) and the Tuscany Health Ecosystem (THE) project with CUP B83C22003920001, under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

1. Althoff, M.: An introduction to CORA 2015. In: Proceedings of the Workshop on Applied Verification for Continuous and Hybrid Systems (2015)
2. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM (2005). https://doi.org/10.1137/1.9780898718713
3. Bacci, G., Bacci, G., Larsen, K.G., Mardare, R.: Converging from branching to linear metrics on Markov chains. Math. Struct. Comput. Sci. **29**(1), 3–37 (2019)
4. Bacci, G., Bacci, G., Larsen, K.G., Tribastone, M., Tschaikowski, M., Vandin, A.: Efficient local computation of differential bisimulations via coupling and up-to methods. In: 36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, 29 June–2 July 2021, pp. 1–14. IEEE (2021). https://doi.org/10.1109/LICS52264.2021.9470555
5. Bacci, G., Bacci, G., Larsen, K.G., Mardare, R.: On the metric-based approximate minimization of Markov chains. In: ICALP. LIPIcs, vol. 80, pp. 104:1–104:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017)
6. Barraud, A.Y.: A numerical algorithm to solve A'XA – X = –Q. In: CDC, pp. 420–423 (1977). https://doi.org/10.1109/CDC.1977.271607
7. Bartocci, E., Kovács, L., Stankovič, M.: Automatic generation of moment-based invariants for prob-solvable loops. In: Chen, Y.-F., Cheng, C.-H., Esparza, J. (eds.) ATVA 2019. LNCS, vol. 11781, pp. 255–276. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31784-3_15
8. Boreale, M.: Algebra, coalgebra, and minimization in polynomial differential equations. In: Esparza, J., Murawski, A.S. (eds.) FoSSaCS 2017. LNCS, vol. 10203, pp. 71–87. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54458-7_5
9. Boreale, M.: Algorithms for exact and approximate linear abstractions of polynomial continuous systems. In: Prandini, M., Deshmukh, J.V. (eds.) HSCC, pp. 207–216. ACM (2018) .https://doi.org/10.1145/3178126.3178137
10. Buchholz, P.: Exact and ordinary lumpability in finite Markov chains. J. Appl. Probab. **31**(1), 59–75 (1994)
11. Buchholz, P.: Lumpability and nearly-lumpability in hierarchical queueing networks. In: Proceedings of 1995 IEEE International Computer Performance and Dependability Symposium, pp. 82–91. IEEE (1995)
12. Cardelli, L., Pérez-Verona, I.C., Tribastone, M., Tschaikowski, M., Vandin, A., Waizmann, T.: Exact maximal reduction of stochastic reaction networks by species lumping. Bioinformatics **37**(15), 2175–2182 (2021). https://doi.org/10.1093/bioinformatics/btab081
13. Cardelli, L., Tribastone, M., Tschaikowski, M.: From electric circuits to chemical networks. Natural Comput. (2019) https://doi.org/10.1007/s11047-019-09761-7
14. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Efficient syntax-driven lumping of differential equations. In: Chechik, M., Raskin, J.-F. (eds.) TACAS 2016. LNCS, vol. 9636, pp. 93–111. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49674-9_6

15. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. In: POPL (2016)
16. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: ERODE: a tool for the evaluation and reduction of ordinary differential equations. In: Legay, A., Margaria, T. (eds.) TACAS 2017. LNCS, vol. 10206, pp. 310–328. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54580-5_19
17. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Maximal aggregation of polynomial dynamical systems. Proc. Natl. Acad. Sci. (PNAS) **114**(38), 10029–10034 (2017)
18. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Syntactic Markovian bisimulation for chemical reaction networks. In: Aceto, L., Bacci, G., Bacci, G., Ingólfsdóttir, A., Legay, A., Mardare, R. (eds.) Models, Algorithms, Logics and Tools. LNCS, vol. 10460, pp. 466–483. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63121-9_23
19. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Guaranteed error bounds on approximate model abstractions through reachability analysis. In: McIver, A., Horvath, A. (eds.) QEST 2018. LNCS, vol. 11024, pp. 104–121. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99154-2_7
20. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Taylor model flowpipe construction for non-linear hybrid systems. In: RTSS, pp. 183–192 (2012)
21. Condon, A.: The complexity of stochastic games. Inf. Comput. **96**(2), 203–224 (1992). https://doi.org/10.1016/0890-5401(92)90048-K
22. Dantzig, G.B.: Application of the simplex method to a transportation problem. In: Koopmans, T. (ed.) Activity analysis of production and allocation, pp. 359–373. Wiley, New York (1951)
23. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Metrics for labelled Markov processes. Theor. Comput. Sci. **318**(3), 323–354 (2004). https://doi.org/10.1016/j.tcs.2003.09.013
24. Fehnker, A., Ivančić, F.: Benchmarks for hybrid systems verification. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 326–341. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24743-2_22
25. Feinberg, M.: Chemical reaction network structure and the stability of complex isothermal reactors—I. The deficiency zero and deficiency one theorems. Chem. Eng. Sci. **42**(10), 2229–2268 (1987).https://doi.org/10.1016/0009-2509(87)80099-4. https://www.sciencedirect.com/science/article/pii/0009250987800994
26. Ford, L.R., Fulkerson, D.R.: Solving the transportation problem. Manag. Sci. **3**(1), 24–32 (1956)
27. Franceschinis, G., Muntz, R.R.: Bounds for quasi-lumpable Markov chains. Perform. Eval. **20**(1–3), 223–243 (1994)
28. Gast, N., Bortolussi, L., Tribastone, M.: Size expansions of mean field approximation: transient and steady-state analysis. ACM SIGMETRICS Perform. Eval. Rev. **46**(3), 25–26 (2019)
29. Ghorbal, K., Goubault, E., Putot, S.: The zonotope abstract domain taylor1+. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 627–633. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02658-4_47
30. Ghorbal, K., Platzer, A.: Characterizing algebraic invariants by differential radical invariants. In: Ábrahám, E., Havelund, K. (eds.) TACAS 2014. LNCS, vol. 8413, pp. 279–294. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54862-8_19
31. Giacalone, A., Jou, C.C., Smolka, S.A.: Algebraic reasoning for probabilistic concurrent games. In: IFIP WG 2.2/2.3, pp. 443–458 (1990)

32. Girard, A., Pappas, G.: Approximate bisimulations for constrained linear systems. In: 44th IEEE Conference on Decision and Control, pp. 4700–4705 (2005). https://doi.org/10.1109/CDC.2005.1582904
33. Girard, A., Pappas, G.: Approximate bisimulations for nonlinear dynamical systems. In: 44th IEEE Conference on Decision and Control, pp. 684–689 (2005). https://doi.org/10.1109/CDC.2005.1582235
34. Girard, A., Pappas, G.J.: Approximation metrics for discrete and continuous systems. IEEE Trans. Autom. Control **52**(5), 782–798 (2007). https://doi.org/10.1109/TAC.2007.895849
35. Hartmanns, A., Klauck, M., Parker, D., Quatmann, T., Ruijters, E.: The quantitative verification benchmark set. In: Vojnar, T., Zhang, L. (eds.) TACAS 2019. LNCS, vol. 11427, pp. 344–350. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17462-0_20
36. Hasani, R., et al.: Closed-form continuous-depth models. arXiv preprint arXiv:2106.13898 (2021)
37. Islam, M.A., et al.: Model-order reduction of ion channel dynamics using approximate bisimulation. Theor. Comput. Sci. **599**, 34–46 (2015). https://doi.org/10.1016/j.tcs.2014.03.018
38. Katoen, J.-P., Kemna, T., Zapreev, I., Jansen, D.N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 87–101. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71209-1_9
39. Kori, M., Ascari, F., Bonchi, F., Bruni, R., Gori, R., Hasuo, I.: Exploiting adjoints in property directed reachability analysis. In: Enea, C., Lal, A. (eds.) CAV 2023, Part II. LNCS, vol. 13965, pp. 41–63. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37703-7_3
40. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. Inf. Comput. **94**(1), 1–28 (1991)
41. Leontief, W.: Input-Output Economics. OUP E-Books, Oxford University Press (1986)
42. Liberzon, D.: Calculus of Variations and Optimal Control Theory: A Concise Introduction. Princeton University Press, Princeton (2011)
43. Liu, J., Zhan, N., Zhao, H.: Computing semi-algebraic invariants for polynomial dynamical systems. In: EMSOFT, pp. 97–106 (2011)
44. Liu, Y.Y., Slotine, J.J., Barabasi, A.L.: Controllability of complex networks. Nature **473**, 167–73 (2011). https://doi.org/10.1038/nature10011
45. Marin, A., Rossi, S.: On the relations between lumpability and reversibility. In: MASCOTS, pp. 427–432 (2014). https://doi.org/10.1109/MASCOTS.2014.59
46. De Nicola, R., Montanari, U., Vaandrager, F.: Back and forth bisimulations. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 152–165. Springer, Heidelberg (1990). https://doi.org/10.1007/BFb0039058
47. Orlin, J.: A faster strongly polynomial minimum cost flow algorithm. In: STOC, pp. 377–387 (1988)
48. Peixoto, T.P.: The Netzschleuder network catalogue and repository (2020). https://doi.org/10.5281/zenodo.7839981
49. Perron, L., Furnon, V.: Or-tools. https://developers.google.com/optimization/
50. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics, Wiley, Hoboken (1994). https://doi.org/10.1002/9780470316887

51. Salamati, M., Soudjani, S., Majumdar, R.: Approximate time bounded reachability for CTMCs and CTMDPs: a Lyapunov approach. In: McIver, A., Horvath, A. (eds.) QEST 2018. LNCS, vol. 11024, pp. 389–406. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99154-2_24
52. Sankaranarayanan, S., Dang, T., Ivančić, F.: Symbolic model checking of hybrid systems using template polyhedra. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 188–202. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78800-3_14
53. Song, L., Zhang, L., Godskesen, J.C.: Bisimulations and logical characterizations on continuous-time Markov decision processes. In: McMillan, K.L., Rival, X. (eds.) VMCAI 2014. LNCS, vol. 8318, pp. 98–117. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54013-4_6
54. Tognazzi, S., Tribastone, M., Tschaikowski, M., Vandin, A.: EGAC: a genetic algorithm to compare chemical reaction networks. In: Genetic and Evolutionary Computation Conference, GECCO, pp. 833–840 (2017). https://doi.org/10.1145/3071178.3071265
55. Tribastone, M., Mayer, P., Wirsing, M.: Performance prediction of service-oriented systems with layered queueing networks. In: Margaria, T., Steffen, B. (eds.) ISoLA 2010. LNCS, vol. 6416, pp. 51–65. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16561-0_12
56. Tschaikowski, M., Tribastone, M.: Tackling continuous state-space explosion in a Markovian process algebra. Theor. Comput. Sci. **517**, 1–33 (2014). https://doi.org/10.1016/j.tcs.2013.08.016
57. Tschaikowski, M., Tribastone, M.: Approximate reduction of heterogenous nonlinear models with differential hulls. IEEE Trans. Autom. Control **61**(4), 1099–1104 (2016). https://doi.org/10.1109/TAC.2015.2457172
58. Tschaikowski, M., Tribastone, M.: Spatial fluid limits for stochastic mobile networks. J. Perform. Eval. **109**, 52–76 (2017). https://doi.org/10.1016/j.peva.2016.12.005
59. Vaidya, P.M.: Speeding-up linear programming using fast matrix multiplication (extended abstract). In: 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October–1 November 1989, pp. 332–337. IEEE Computer Society (1989). https://doi.org/10.1109/SFCS.1989.63499
60. Wirsing, M., et al.: Sensoria patterns: augmenting service engineering with formal analysis, transformation and dynamicity. In: Margaria, T., Steffen, B. (eds.) ISoLA 2008. CCIS, vol. 17, pp. 170–190. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88479-8_13