Graph problems arising from parameter identification of discrete dynamical systems

Journal Article

Author(s):
Borchers, Steffen; Bosio, Sandro; Findeisen, Rolf; Haus, Utz-Uwe; Rumschinski, Philipp; Weismantel, Robert

Publication date:
2011-06

Permanent link:
https://doi.org/10.3929/ethz-b-000037714

Originally published in:
Mathematical Methods of Operations Research 73(3), https://doi.org/10.1007/s00186-011-0356-3

Math Meth Oper Res (2011) 73:381–400
DOI 10.1007/s00186-011-0356-3

ORIGINAL ARTICLE

Graph problems arising from parameter identification of discrete dynamical systems

Steffen Borchers · Sandro Bosio · Rolf Findeisen · Utz-Uwe Haus · Philipp Rumschinski · Robert Weismantel

Received: 13 November 2009 / Accepted: 23 February 2011 / Published online: 21 April 2011

Abstract This paper focuses on combinatorial feasibility and optimization problems that arise in the context of parameter identification of discrete dynamical systems. Given a candidate parametric model for a physical system and a set of experimental observations, the objective of parameter identification is to provide estimates of the parameter values for which the model can reproduce the experiments. To this end, we define a finite graph corresponding to the model, to each arc of which a set of parameters is associated. Paths in this graph are regarded as feasible only if the sets of parameters corresponding to the arcs of the path have nonempty intersection. We study feasibility and optimization problems on such feasible paths, focusing on computational complexity. We show that, under certain restrictions on the sets of parameters, some of the problems become tractable, whereas others are NP-hard. In a similar vein, we define and study some graph problems for experimental design, whose goal is to support the scientist in optimally designing new experiments.

Keywords Graph problems · Computational complexity · Dynamical systems · Parameter identification

1 Introduction

Discrete dynamical systems are an important modeling tool for analysis and prediction of physical processes, describing the discrete-time evolution $(x_t)_{t\in\mathbb{N}}$ of the system

S. Borchers · R. Findeisen · P. Rumschinski
Institut für Automatisierungstechnik, Otto-von-Guericke Universität, Universitätsplatz 2, 39106 Magdeburg, Germany

S. Bosio (✉) · U.-U. Haus · R. Weismantel
Institute for Operations Research (IFOR), Eidgenössischen Technischen Hochschule (ETH) Zürich, Rämistrasse 101, 8092 Zürich, Switzerland
e-mail: sandro.bosio@ifor.math.ethz.ch

state by means of a transition function $x_{t+1} = f(x_t, p)$, with $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$, where $p \in \mathbb{R}^{n_p}$ is some fixed parameter vector. Limited knowledge about the system structure, however, frequently results in competing modeling hypotheses, whose parameters are often unknown. A successful analysis of the system under study, resulting in a model that captures its essential behavior, requires discrimination among these alternatives. While model correctness is in general impossible to prove, as this would require infinite experimental evidence, it may be possible to prove that some of the model alternatives are inconsistent with the available experimental data. This is known as *model invalidation*, and amounts to deciding whether there is any parameter value for which the model can reproduce the measurements. A more ambitious goal is that of explicitly finding, for each model alternative, the set of all such parameters, if they exist. This is known as *parameter identification*.

Traditional statistical approaches to parameter identification are Monte Carlo simulation [see e.g. Robert and Casella (2004)] and data fitting. The latter is often performed by optimization of some likelihood criteria, the most common of which is least squares (Marquardt 1963). Some interesting recent works in this field consider the use of collocation methods (Ramsay et al. 2007) approximating the output of the dynamical system, coupled with profiled estimation techniques. Statistical approaches, however, provide results of probabilistic nature. Some studies of necessary conditions for model validity use frequency-domain data (Smith and Doyle 1992) and time-domain data (Evans et al. 2004; Schnell et al. 2006), but are frequently limited to linear models. A recent approach to model invalidation, applicable to a larger class of models, is based on the existence of a *barrier function* separating model trajectories from measurement data (Prajna 2006). However, finding a barrier is a nontrivial task with several degrees of freedom, and its very existence is not guaranteed for all invalid models.

In this paper we present a graph-based combinatorial approach to parameter identification and model invalidation, with a focus on computational complexity. The application of this approach to real-size problems is out of the scope of this paper, and requires algorithmic tools that are subject to ongoing research. The approach consists in building a *transition graph*, a layered graph representing system trajectories for entire parameter regions. Each layer of this graph corresponds to a time index, and its nodes represent the discrete states (state space regions, as obtained by discretization) at the given time. Arcs between nodes of consecutive layers represent the possible discrete-state transitions. Associated to each arc is the subset of parameters allowing the corresponding transition, denoted *transition parameters*. We say that a path in a transition graph is a *transition path* if the intersection of the transition parameters associated to the arcs of the path is nonempty. The parameters in such an intersection, which we denote *path parameters*, are those that are consistent with all the transitions in the path.

Consider a discrete dynamical system and a time-indexed set of experimental measurements. By mapping the measurements onto nodes of the transition graph of the system, model invalidation can be relaxed into the problem of finding a transition path that touches the measurement nodes, and parameter identification into the problem of computing the union of the path parameters over all such paths. Similarly, some experimental design problems can be formulated as optimization problems on transition paths. This approach is applicable to any system class for which the transition parameters can be efficiently derived. This is for example the case for polynomial and rational transition functions, for which a semidefinite programming relaxation approach, introduced in Kuepfer et al. (2007) for stationary systems and extended to discrete dynamical systems in Borchers et al. (2009), allows to efficiently bound the transition parameters.

The remainder of the paper is organized as follows. In Sect. 2 the concept of transition graph is defined, describing its relation with the corresponding dynamical system. The problem of finding a transition path is studied in Sect. 3, showing how its computational complexity depends on the way transition parameters are defined. The application of the framework to parameter identification is discussed in Sect. 4, and some experimental design problems defined upon transition paths are proposed in Sect. 5. Section 6 provides some concluding remarks and outlines future work.

2 Transition graphs

A *discrete dynamical system* is a system of parametric equations $x_{t+1} = f(x_t, p)$ describing the discrete-time evolution of a state vector $x_t \in X$ at the time-index $t \in \mathbb{N}$ for a given parameter vector $p \in P$ by means of a transition function $f : X \times P \to X$. The feasible regions $X \subseteq \mathbb{R}^{n_x}$ and $P \subseteq \mathbb{R}^{n_p}$ for states and parameters are typically polyhedra, and frequently simply boxes.

This paper focuses on parameter identification of discrete dynamical systems, deriving a *transition graph* that represents the possible discrete-time trajectories for entire parameter regions. In order to define the transition graph, we consider a discretization of the state space X (which is typically continuous) by means of a partition into a finite set of regions $\mathscr{D} = \{X_j \subseteq X : j \in D\}$. We refer to an index $j \in D$ of the partition as a *discrete state*, by that indicating the state region $X_j \in \mathscr{D}$. For simplicity we consider a time-invariant space discretization. Our approach can be easily extended to time-variant discretizations, but with higher computational requirements.

For a fixed parameter $p \in P$, each point $x \in X$ is mapped by the transition function to a unique point $f(x, p) \in X$. Conversely, as shown in Fig. 1, a discrete state is not necessarily mapped to a unique discrete state (i.e., the discrete dynamical system becomes non-deterministic when the state space is discretized). The set of discrete states reachable from a discrete state $j \in D$ for a parameter $p \in P$ is given by

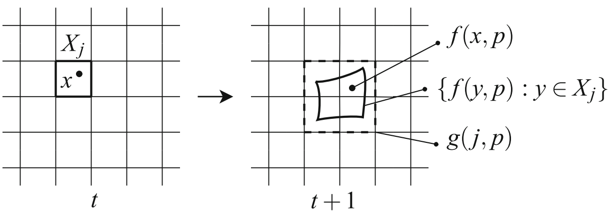

Fig. 1 State and discrete-state transitions. A point x is mapped to a point $f(x, p)$, while a discrete state j is mapped to a set of discrete states $g(j, p)$, which is the smallest set of discrete states containing the image of X_j under the transition function f

$$g(j,p) = \{\ell \in D : y = f(x,p) \text{ for some } x \in X_j, y \in X_\ell\}, \quad g : D \times P \to 2^D,$$

where $2^D = \{D' \subseteq D\}$ is the power set of D. A function taking values in a power set is also called a *multimap*. Given a pair of discrete states $j, \ell \in D$, the *transition parameters* that allow the transition from j to ℓ can then be defined by the multimap

$$\phi(j,\ell) = \{p \in P : \ell \in g(j,p)\}, \quad \phi : D \times D \to 2^P.$$

If $\phi(j,\ell) \neq \emptyset$ we say that (j,ℓ) is a *feasible (discrete-state) transition*.

Given a discretization D, the transition parameters multimap $\phi : D \times D \to 2^P$, and a discrete time interval $T = \{1, \dots, \tau\}$ for some $\tau \in \mathbb{N}$, the resulting discretized dynamical system can be described as a layered digraph $TG(D, \phi, T) = (V, A)$, where the node set

$$V = \{v_j^t : j \in D,\ t \in T\}$$

contains one node for each discrete state and time step, and the arc set

$$A = \left\{ \left(v_j^t, v_\ell^{t+1} \right) : \phi(j,\ell) \neq \emptyset, \quad j,\ell \in D,\ t, t+1 \in T \right\},$$

contains one arc for each feasible transition and time step but τ, where $t, t+1 \in T$ stands for $t \in T \backslash \{\tau\}$. We denote by $V^t = \{v_j^t \in V : j \in D\}$ the node set for layer t, and by $A^t = \left\{ \left(v_j^t, v_\ell^{t+1} \right) \in A : j, \ell \in D \right\}$ the set of arcs between two consecutive layers V^t and V^{t+1} (Fig. 2).

In the remainder we assume that the transition parameter multimap ϕ is given in some compact form, and study the complexity of connectivity problems on transition graphs, and their relation to model invalidation and parameter identification. Let us remark that finding the transition parameters is in general a challenging task by itself, whose solution is out of the scope of this paper. Here we only notice that an approximation of such sets can be obtained by repeatedly solving, within a recursive bisection algorithm, the following nonlinear decision problem.

Problem 1 (*Feasibility problem*) Given $j, \ell \in D$ and $Q \subseteq P$, decide if there exists some $x \in X_j$, $y \in X_l$, and $p \in Q$ for which $y = f(x,p)$.

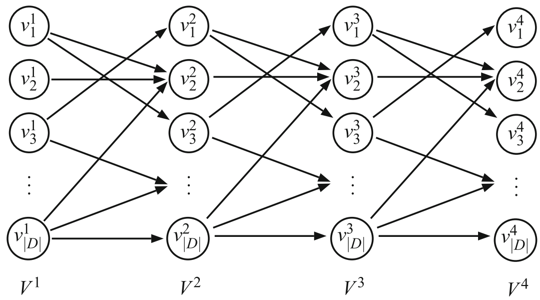

Fig. 2 Example of transition graph with $T = \{1, \dots, 4\}$. Note that the layers have the same structure. More formally, all the subgraphs $\left(V^t \cup V^{t+1}, A^t\right)$ for $t, t+1 \in T$ are isomorphic to each other. This is because the parameters and the transition function are assumed to be time invariant

Solving Problem 1 is in general non-trivial, in particular for highly-nonlinear systems. For a solution approach by semidefinite programming for the case of polynomial and rational transition functions, the reader is referred to Borchers et al. (2009).

3 Connectivity problems for transition graphs

Given a transition graph $TG(D, \phi, T) = (V, A)$ and two nodes $s \in V^t, d \in V^\ell$, with $t < \ell$, an s–d path is an ordered sequence of nodes $B = (v^t_{i_t}, \dots, v^\ell_{i_\ell})$ such that $v^t_{i_t} = s$, $v^\ell_{i_\ell} = d$, and $(v^k_{i_k}, v^{k+1}_{i_{k+1}}) \in A$ for every $t \leqslant k < \ell$. We define the *path parameters* of B as the intersection of the transition parameters associated to the arcs of the path, denoted

$$\phi(B) = \bigcap_{k=t}^{\ell-1} \phi(i_k, i_{k+1}).$$

If $\phi(B) \neq \emptyset$ we say that B is a *transition path*, or equivalently a ϕ-feasible path. We then say that s is ϕ-connected to d if there exists a ϕ-feasible s–d path. Note that ϕ-connectivity is neither symmetric nor transitive, as the graph is directed and the composition of ϕ-feasible paths is not necessarily ϕ-feasible.

Given a path $B = (v^t_{i_t}, \dots, v^\ell_{i_\ell})$, we say that a state trajectory $(x_k)_{t \leqslant k \leqslant \ell}$ *lies inside* B if $x_k \in X_{i_k}$ for all $t \leqslant k \leqslant \ell$. The set $\phi(B)$ contains all parameters $p \in P$ for which there exists a state trajectory $(x_k)_{t \leqslant k \leqslant \ell}$ lying inside B such that $x_{k+1} = f(x_k, p)$ for all $t \leqslant k < \ell$. Note that, due to the state space discretization, there may exist some $p \in \phi(B)$ that does not realize any such trajectory. Emptiness of $\phi(B)$ is thus a sufficient condition for proving that no trajectory lying inside B exists, but not a necessary one.

3.1 ϕ-connectivity for arbitrary graphs

A transition graph is a particular layered digraph, defined upon the transition parameters. More generally, given an arbitrary digraph $G = (V, A)$ and a groundset P, finite or infinite, let us define an *arc multimap* as a multimap $\phi : V \times V \to 2^P$ for which $\phi(u, v) \neq \emptyset$ if and only if $(u, v) \in A$. In other words, an arc multimap is an association of nonempty sets to the arcs of a digraph. In the remainder, we assume that the membership test $p \in \phi(u, v)$ can be done efficiently, if p admits a compact representation. Whenever considering an arbitrary digraph $G = (V, A)$, we also assume that $|A| \geqslant |V|$.

For simplicity of notation, let us denote $\phi(a) = \phi(u, v)$ for an arc $a = (u, v) \in A$, and given a path B in G let $\phi(B)$ be the intersection of $\phi(a)$ for all the arcs in B. Note that for ϕ-connectivity problems we can restrict our attention to simple paths (paths that do not contain a loop), as for any non-simple s–d path B there exists a simple s–d path B' for which $\phi(B) \subseteq \phi(B')$. As simple paths can be represented by the subset of arcs used, we can naturally extend the notation $\phi(B)$ to an arbitrary subset of arcs

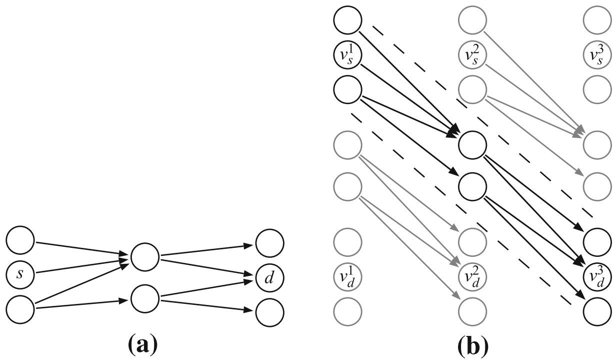

Fig. 3 **a** An example digraph $G = (V, A)$ with 3 layers, with nodes s and d outlined. **b** The transition graph $TG(V, \phi, T)$, which contains the original digraph G as the largest connected component (in evidence in the figure)

$B \subseteq A$ by defining $\phi(B) = \cap_{a \in B} \phi(a)$. We can then formulate ϕ-connectivity for a general graph.

Problem 2 Given a digraph $G = (V, A)$, two nodes $s, d \in V$, a groundset P, and an arc multimap $\phi : V \times V \to 2^P$, decide whether there exists a ϕ-feasible s–d path.

A positive ϕ-connectivity result obtained for an arbitrary digraph G and arc multimap ϕ is clearly directly valid also for transition graphs. Moreover, the following lemma implies that any negative result obtained for the case in which G is layered directly holds for the case of transition graphs.

Lemma 1 *Given a layered digraph $G = (V, A)$, a groundset P, and an arc multimap $\phi : V \times V \to 2^P$, it is possible to build in polynomial time a transition graph that is equivalent to G in terms of ϕ-connectivity.*

Proof Let $T = \{1, \ldots, \tau\}$ be the index set of the layers of G. We show that checking ϕ-connectivity in G is equivalent to checking ϕ-connectivity in the transition graph $TG(V, \phi, T)$.

Let two nodes $s, d \in V$ be given, and assume without loss of generality that s belongs to the first layer, and d to the last one. The transition graph $TG(V, \phi, T)$ has by definition one layer for each $t \in T$. For each layer, it has one node v_u^t for every $u \in V$ and one arc $\left(v_u^t, v_w^{t+1}\right)$ for every $u, w \in V$ for which $\phi(u, w) \neq \emptyset$, that is, for every $(u, w) \in A$. As shown in Fig. 3, this transition graph is composed of "diagonal" connected components (assuming a proper vertical placement of the nodes), the largest of which is isomorphic to the original digraph G. In particular, it is easy to see that there exists a ϕ-feasible s–d path in G if and only if there exists a ϕ-feasible path in $TG(V, \phi, T)$ between nodes v_s^1 and v_d^τ. $\square$

Arbitrary multimaps have a large degree of expression freedom. Indeed, the following simple result shows that in general even testing ϕ-feasibility for a *given* path B is NP-hard. The problem belongs to NP when a compact certificate for $\phi(B) \neq \emptyset$ exists. A rational point $p \in \phi(B)$ with compact encoding is clearly a valid certificate, and can always be found e.g. if the sets ϕ are polyhedral, or if P is a bounded discrete set. In the general case, however, the existence of such a point cannot be guaranteed.

Proposition 1 *Given a path B in a digraph $G = (V, A)$, a groundset P, and an arc multimap $\phi : V \times V \to 2^P$, deciding whether B is ϕ-feasible is* NP*-hard.*

Proof We prove Proposition 1 by reduction from the NP-complete problem of checking emptiness of a binary set $Q = \{\boldsymbol{x} \in \{0, 1\}^n : A\boldsymbol{x} \geqslant b\}$, with $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$. Let $Q_j = \{\boldsymbol{x} \in \{0, 1\}^n : A_j\boldsymbol{x} \geqslant b_j\}$ be the binary set defined by the jth constraint. We assume without loss of generality that $Q_j \neq \emptyset$ for every j (which can be easily checked in polynomial time), as otherwise the instance is infeasible. We then create a digraph $G = (V, A)$ with $V = \{v_1, \ldots, v_{m+1}\}$ and $A = \{(v_j, v_{j+1}) : 1 \leqslant j \leqslant m\}$, which is a path of $m + 1$ nodes, and define an arc multimap $\phi : V \times V \to 2^P$, with $P = \{0, 1\}^n$, by setting $\phi(v_j, v_{j+1}) = Q_j$ for every $1 \leqslant j \leqslant m$. Note that a compact encoding of ϕ requires only A and b. Then, checking if the path $B = (v_1, \ldots, v_{m+1})$ is ϕ-feasible is clearly the same as checking if $Q \neq \emptyset$. □

3.2 Testing ϕ-connectivity for box multimaps

As we have seen in the previous section, for arbitrary multimaps even testing ϕ-feasibility for a given path is NP-hard. Thus, to solve ϕ-connectivity efficiently one has to require some structure on the multimap. One of the simplest structures that can be considered is to require the sets $\phi(u, v)$ to be axis-aligned boxes. In this case testing ϕ-feasibility for a given path becomes a simple component-wise check on a set of boxes. However, even for this very simple structure testing ϕ-connectivity is NP-complete, both for finite and infinite groundsets.

Theorem 1 *Given a digraph $G = (V, A)$, two nodes $s, d \in V$, a groundset P, and an arc multimap $\phi : V \times V \to 2^P$ for which testing ϕ-feasibility is in* P*, deciding whether there exists a ϕ-feasible s–d path is* NP*-complete, even if the sets $\phi(\cdot, \cdot)$ are axis-aligned boxes, G is layered, and has at most three nodes per layer.*

Proof As testing ϕ-feasibility is in P, deciding ϕ-connectivity is in NP. We prove Theorem 1 by reduction from 3-SAT, a well-known NP-complete problem. An instance of 3-SAT consists of n variables $X_1, \ldots, X_n$ and m disjunctive clauses $C_1, \ldots, C_m$, each involving three literals (variables or negated variables). The goal is to find a truth assignment satisfying all the clauses. Let each clause be given as a set $C_t = \{L_1^t, L_2^t, L_3^t\}$, with $L_k^t \in \{X_1, \neg X_1, \cdots, X_n, \neg X_n\}$ for $k \in \{1, 2, 3\}$. We assume without loss of generality that each clause is composed of three distinct literals, and that there is no clause C_t for which $X_i, \neg X_i \in C_t$ for some variable X_i, as otherwise such a clause can be discarded.

Given a 3-SAT instance, we build a layered digraph $G = (V, A)$ as follows. For each clause C_t we build a layer $V^t = \{v_1^t, v_2^t, v_3^t\}$ with three nodes, one for each literal in C_t. For every pair of consecutive layers V^t, V^{t+1} we include an arc $\left(v_k^t, v_\ell^{t+1}\right)$ for every $k, \ell \in \{1, 2, 3\}$. We then add an initial layer $V^0 = \{s\}$ and an ending layer

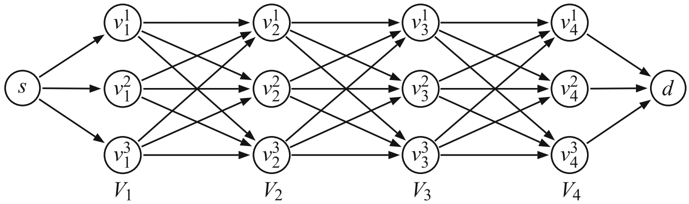

Fig. 4 Digraph obtained by the reduction of a 3-SAT instance with 4 clauses

$V^{m+1} = \{d\}$, each with a single node, and include the arcs (s, v_k^1) and (v_k^m, d) for all $k \in \{1, 2, 3\}$. Note that the resulting digraph, depicted in Fig. 4, depends only on the *number* of clauses.

We set $P = \{0, 1\}^n$, and define the arc multimap $\phi : V \times V \to 2^P$ as follows. Each node $v_k^t \in V$ corresponds to a literal L_k^t, and hence to some variable X_i. If $L_k^t = X_i$ we set $\phi\left(v_k^t, w\right) = \{\boldsymbol{x} \in P : x_i = 1\}$ for all $(v_k^t, w) \in A$. If $L_k^t = \neg X_i$ we set $\phi(v_k^t, w) = \{\boldsymbol{x} \in P : x_i = 0\}$ for all $(v_k^t, w) \in A$. Note that these sets are $(n-1)$-dimensional boxes. Finally, we set $\phi(s, w) = P$ for every $(s, w) \in A$.

It is easy to see that a ϕ-feasible s–d path B cannot touch nodes corresponding to opposite literals, and hence that it yields a (partial) truth assignment satisfying all the clauses. Conversely, given a satisfying truth assignment we can easily derive, although not uniquely, a ϕ-feasible s–d path. Therefore, there exists a ϕ-feasible s–d path if and only if the 3-SAT instance admits a satisfying truth assignment.

The result can be extended to the continuous case simply by choosing $P = [0, 1]^n$. If the boxes must be full-dimensional, we can adapt the proof by defining $\phi(v_k^t, w) = \{\boldsymbol{x} \in P : x_i \geqslant 1 - \epsilon\}$ if $L_k^t = X_i$ and $\phi(v_k^t, w) = \{\boldsymbol{x} \in P : x_i \leqslant \epsilon\}$ if $L_k^t = \neg X_i$, for every $(v_k^t, w) \in A$, for an arbitrary positive constant $\epsilon < 0.5$. Note that, as a result, any non-empty intersection of boxes is full-dimensional as well. $\square$

3.3 Testing ϕ-connectivity for explicit groundsets

Proposition 1 and Theorem 1 show that a compact representation of an arc multimap $\phi : V \times V \to 2^P$ for a digraph $G = (V, A)$ does not require the groundset P nor the sets $\phi(u, v)$ to be given explicitly, and can express exponentially large or unbounded sets. In this section we show that if P is *explicit*, i.e., finite and given explicitly, then ϕ-connectivity can be solved in polynomial time in the size of G and P.

Let us introduce some notation, which will also be used later. Given a digraph $G = (V, A)$, a groundset P, an arc multimap $\phi : V \times V \to 2^P$, and a subset $Q \subseteq P$, we define $G_Q = (V, A_Q)$ as the subgraph of G induced by Q, in which A_Q contains only those arcs $a \in A$ for which $Q \subseteq \phi(a)$. If the subset Q is explicit, then G_Q can be created in polynomial time in $|A|$ and $|Q|$. For simplicity, we denote by G_p the subgraph $G_{\{p\}}$ for every $p \in P$.

Proposition 2 *Given a digraph $G = (V, A)$, an explicit groundset P, and an arc multimap $\phi : V \times V \to 2^P$, ϕ-connectivity can be checked in time $O(|A| \cdot |P|)$.*

Proof Given two nodes $u, v \in V$, it is easy to see that there exists a ϕ-feasible u–v path in G if and only if there exists an u–v path in G_p for at least one $p \in P$. As both constructing the subgraph G_p and searching for an u–v path in G_p cost $O(|A|)$ time, the complexity $O(|A| \cdot |P|)$ directly follows. □

As a straightforward corollary, if the cardinality of P is limited by a polynomial in the size of the graph, then ϕ-connectivity can be solved in polynomial time. This is however not true if P is not explicit, as it is in general NP-hard to enumerate the elements of a set, even assuming there is only one (Valiant and Vazirani 1985).

Given two nodes $u, v \in V$, we are also interested in finding the union of the path parameters over all u–v paths in G. We formally define such sets as the multimap $\Phi : V \times V \to 2^P$ with

$$\Phi(u, v) = \begin{cases} \bigcup_{B \in \mathscr{B}(u,v)} \phi(B) & \text{if } u \neq v \\ P & \text{otherwise,} \end{cases}$$

where $\mathscr{B}(u, v)$ is the set of all simple u–v paths in G (there is no need to restrict to ϕ-feasible paths). If G is the transition graph of a dynamical system, $\Phi(u, v)$ gives an outer-approximation of the set of parameters that can produce a trajectory starting in the region corresponding to node u and ending in the region corresponding to node v (see Sect. 4 for further details). Let us then formalize this problem.

Problem 3 Given a digraph $G = (V, A)$, two nodes $u, v \in V$, a groundset P, and an arc multimap $\phi : V \times V \to 2^P$, compute the set $\Phi(u, v)$.

Node u is ϕ-connected to v if and only if $\Phi(u, v) \neq \emptyset$. The set $\Phi(u, v)$ can indeed be equivalently defined as the set of all elements $p \in P$ for which there exist an u–v path B in G with $p \in \phi(B)$, i.e., for which there exists an u–v path in G_p. This gives the following simple result.

Lemma 2 *Given a digraph $G = (V, A)$, an explicit groundset P, and an arc multimap $\phi : V \times V \to 2^P$, the set $\Phi(u, v)$ for a given pair of nodes $u, v \in V$ can be computed in $O(|A| \cdot |P|)$ time. Moreover, the sets $\Phi(u, v)$ for a given source $u \in V$ and every destination $v \in V$ can also be computed in $O(|A| \cdot |P|)$ time, and the sets $\Phi(u, v)$ for all pairs $u, v \in V$ can be computed in $O(|V| \cdot |A| \cdot |P|)$ time.*

Proof The set $\Phi(u, v)$ for a given pair u, v can be computed in time $O(|A| \cdot |P|)$ by deriving in time $O(|A|)$ the subgraph G_p for each $p \in P$, and then searching in time $O(|A|)$ for an u–v path in G_p. The same complexity applies to the single-source version of the problem, finding in time $O(|A|)$ the set of nodes reachable from u in G_p. The time complexity for the all-pairs case then follows directly. Other complexity bounds for the all-pairs case can be obtained by computing the reachability matrix of G_p, i.e., the incidence matrix of the transitive closure [see e.g. Schrijver (2003)]. □

Due to path composition, it is also easy to verify that the following properties hold for every pair of distinct nodes $u, v \in V$:

$$\begin{aligned}\Phi(u,v) &= \bigcup_{w \in V \setminus \{u\}} \phi(u,w) \cap \Phi(w,v)\\ &= \bigcup_{w \in V \setminus \{v\}} \Phi(u,w) \cap \phi(w,v)\\ &= \phi(u,v) \cup \bigcup_{w \in V \setminus \{u,v\}} \Phi(u,w) \cap \Phi(w,v).\end{aligned}$$

These properties are the analogue of the composition properties for shortest paths. Instead of distance labels and edge weights, we have parameter sets Φ and ϕ respectively, with empty sets playing the role of infinite weights for arcs not belonging to the graph. Then, as path composition operation, instead of cost addition we have set intersection, and instead of taking the minimum among the alternative path labels we take the union. As a consequence, any shortest-path algorithm with complexity $O(f)$ can be adapted to compute the sets Φ in time $O(f \cdot |P|)$. Note that, contrary to the shortest-path case, there is no notion of negative cycle for ϕ-feasible paths, so that the sets Φ are always defined.

For a layered digraph, the transitive closure can be computed more efficiently. In particular, for a transition graph $TG(D, \phi, T) = (V, A)$ the transition parameters are defined for pairs of discrete states, and not for pairs of nodes. For any pair of nodes $v_j^t, v_\ell^k \in V$, with $t < k$, we can write the recursion

$$\begin{aligned}\Phi\left(v_j^t, v_\ell^k\right) &= \bigcup_{i \in D} \Phi\left(v_j^t, v_i^{k-1}\right) \cap \phi(v_i^{k-1}, v_\ell^k)\\ &= \bigcup_{i \in D} \phi\left(v_j^t, v_i^{t+1}\right) \cap \Phi\left(v_i^{t+1}, v_\ell^k\right).\end{aligned}$$

We can then define $\Phi^t(j, \ell) = \Phi(u_j^1, v_\ell^{t+1})$, for $t, t+1 \in T$. It is straightforward to see that $\Phi^1 = \phi$, and that $\Phi(u_j^t, v_\ell^k) = \Phi^{k-t}(j, \ell)$, for every $t, k \in T$ with $t < k$. This gives the following result.

Proposition 3 *Given a transition graph $TG(D, \phi, T) = (V, A)$, with $\phi : D \times D \to 2^P$ for an explicit groundset P, the sets $\Phi(u, v)$ for all pairs $u, v \in V$ can be computed in $O(|D| \cdot |A| \cdot |P|)$ time.*

3.4 Finding ϕ-infeasible paths

Assuming testing ϕ-feasibility is in P, Theorem 1 shows that deciding if a ϕ-feasible s–d path exists is NP-complete, so that to find one, if it exists, one may need to enumerate exponentially many ϕ-infeasible paths (unless P = NP). Interestingly, deciding if a ϕ-infeasible s–d path exists is also NP-complete, and remains NP-complete even when the groundset P is explicit. Note that this is *not* the co-NP version of Problem 2, which would require to prove that *all* the s–d paths are ϕ-infeasible. Indeed, given any instance, solving at least one of these two problems is trivial, as any path is either ϕ-feasible or ϕ-infeasible.

Lemma 3 *Given a digraph $G = (V, A)$, two nodes $s, d \in V$, an explicit groundset P, and an arc multimap $\phi : V \times V \to 2^P$, deciding whether there exists a ϕ-infeasible s–d path is* NP-*complete, even if G is layered and $|P| = O(\sqrt[k]{|V|})$ for some $k \in \mathbb{N}$.*

Proof For an explicit groundset, testing ϕ-feasibility is in P, and hence the problem is in NP. We prove Lemma 3 by reduction from the NP-complete problem of deciding if a given graph has an Hamiltonian path (a path touching each node exactly once). Given a graph $H = (U, E)$, with $|U| = n$, we build an instance of our problem as follows. We create a layered digraph $G = (V, A)$ with n layers $V^t = \{v_u^t : u \in U\}$ of n nodes each, for $1 \leqslant t \leqslant n$, and two additional layers $V^0 = \{s\}$ and $V^{n+1} = \{d\}$. The arc set is defined by introducing an arc (v_u^t, v_w^{t+1}) for every $\{u, w\} \in E$ and $1 \leqslant t < n$, and the arcs (s, v_u^1) and (v_u^n, d) for every $u \in U$. The groundset is $P = U$, and the arc multimap is defined by $\phi(v_u^t, v_w^{t+1}) = U \backslash \{w\}$, $\phi(s, v_u^1) = U \backslash \{u\}$, and $\phi(v_u^n, d) = U$.

An s–d path B in G corresponds to a path in H touching n (not necessarily distinct) nodes, and it is easy to see that $\phi(B)$ contains exactly those nodes that are not touched by the path. Therefore, the path in H corresponding to an s–d path B is simple, and hence Hamiltonian, if and only if $\phi(B) = \emptyset$. In the reduction we have $|P| = O(\sqrt{|V|})$, but it is easy to add an arbitrary polynomial number of nodes and arcs to G without changing the structure of the reduction, thus completing the proof. □

Lemma 3 can be easily extended into the following corollary.

Corollary 1 *Given a digraph $G = (V, A)$, two nodes $s, d \in V$, an explicit groundset P, an arc multimap $\phi : V \times V \to 2^P$, and a subset $Q \subseteq P$, deciding whether there exists an s–d path B such that $\phi(B) = Q$ is* NP-*complete, even if G is layered and $|P \setminus Q| = O(\sqrt[k]{|V|})$ for some $k \in \mathbb{N}$.*

Proof For $Q = \emptyset$, Corollary 1 is exactly Lemma 3. We prove the case $Q \neq \emptyset$ by reduction from Lemma 3. Let the input in Lemma 3 consist of a layered digraph $G = (V, A)$, two nodes $s, d \in V$, an explicit groundset R, and an arc multimap $\theta : V \times V \to 2^R$ with $|R| = O(\sqrt[k]{|V|})$ for some $k \in \mathbb{N}$. The reduction keeps G, creates a groundset $P = R \cup Q$, where we assume without loss of generality that $R \cap Q = \emptyset$, and builds an arc multimap $\phi : V \times V \to 2^P$ defined as $\phi(a) = \theta(a) \cup Q$ for all $a \in A$. It is then trivial to see that there exists an s–d path B with $\phi(B) = Q$ if and only if there exists a θ-infeasible s–d path. The proof is concluded by noting that $P \setminus Q = R$, from which the condition $|P \setminus Q| = O(\sqrt[k]{|V|})$ follows. □

The above corollary shows that an oracle providing an s–d path B with $\phi(B) = Q$ for an arbitrary $Q \subseteq P$ allows to solve ϕ-infeasibility, even if Q is restricted to be non-empty. The following results shows that the converse also holds, and hence that the two problems are equivalent.

Proposition 4 *Given a digraph $G = (V, A)$, two nodes $s, d \in V$, an explicit groundset P, and an arc multimap $\phi : V \times V \to 2^P$, if an oracle for θ-infeasibility for arc multimaps $\theta(a) : V \times V \to 2^P$ is available, then it is possible to find in polynomial time a path B with $\phi(B) = Q$, for any $Q \subseteq P$.*

Proof Consider the subgraph $G_Q = (V, A_Q)$, which contains the arcs $a \in A$ for which $Q \subseteq \phi(a)$. Any s–d path B in G such that $\phi(B) = Q$ is an s–d path in G_Q. We can then define a multimap $\theta : V \times V \to 2^P$ by setting $\theta(a) = \phi(a)\backslash Q$ for every $a \in A_Q$. Note that G_Q and the multimap θ can be constructed efficiently.

It is easy to see that there exists an s–d path B in G with $\phi(B) = Q$ if and only if there exists a θ-infeasible s–d path B in G_Q. The multimap θ, however, is not necessarily an arc multimap for G_Q, as there could be arcs in G_Q for which $\theta(a) = \emptyset$. To apply the oracle, we proceed as follows. For any $a \in A_Q$ with $\theta(a) = \emptyset$, if there exists an s–d path B in G_Q with $a \in B$, which can be found by two connectivity tests, then we have found a path (not necessarily simple) with $\phi(B) = Q$. Otherwise, the arc can be removed. At the end we are left with a smaller graph, for which θ is a proper arc multimap. □

The next result shows that if P is sufficiently small, both problems become easy.

Proposition 5 *Given a digraph $G = (V, A)$, with $|A| = m$, two nodes $s, d \in V$, an explicit groundset P with $|P| = \log k \log m$ for some $k > 0$, and an arc multimap $\phi : V \times V \to 2^P$, then finding a ϕ-infeasible s–d path can be solved in polynomial time.*

Proof If $|P| = \log k \log m$, we can enumerate all the subcollections $\mathcal{Q} \subseteq 2^P$ of the power set of P, whose number is $O(m^k)$. For each $\mathcal{Q} \subseteq 2^P$, let $G_{\mathcal{Q}}$ be the subgraph of G obtained by including only arcs $a \in A$ for which $\phi(a) \in \mathcal{Q}$. Then, there is a ϕ-infeasible s–d path in G if and only if there exists an s–d path in $G_{\mathcal{Q}}$ for a subcollection $\mathcal{Q} \subseteq 2^P$ for which the intersection of all $Q \in \mathcal{Q}$ is empty. □

A special case of Proposition 5 is a groundset with constant cardinality. Note that the assumption $|P| = \log k \log m$ is equivalent to $|P| = \log \log m + k'$. Multiplicative constants as in $|P| = O(\log \log m)$ would result, for the above enumeration, in a super-polynomial complexity. Whether ϕ-infeasibility can be solved for this case, or more in general for $|P| = O(\log m)$, remains open.

3.5 Testing ϕ-connectivity in fixed dimension

Theorem 1 states that testing ϕ-connectivity is NP-complete, even for multimaps ϕ where each set $\phi(u, v)$ is an axis-aligned box. In this section we study the case $P \subseteq \mathbb{R}^d$, where d is assumed to be constant, and we show that if each $\phi(u, v)$ is a (possibly nonconvex) polyhedron in $\mathbb{R}^d$, then ϕ-connectivity can be solved in polynomial time. For simplicity, let us start considering axis-aligned boxes in $\mathbb{R}^d$.

Theorem 2 *Given a digraph $G = (V, A)$, a groundset $P \subseteq \mathbb{R}^d$ with d constant, and an arc multimap $\phi : V \times V \to 2^P$ where each $\phi(u, v)$ is an axis-aligned box, the sets $\Phi(u, v)$ for all pairs $u, v \in V$ can be computed in $O(m^{d+1})$ time, where $m = |A|$.*

Proof Let the arc multimap be $\phi(a) = \{x \in \mathbb{R}^d : s_i^a \leqslant x_i \leqslant e_i^a,\ 1 \leqslant i \leqslant d\}$, for $a \in A$. Given any subset $B \subseteq A$, $\phi(B)$ is the axis-aligned box $\phi(B) = \{x \in \mathbb{R}^d : \max_{a \in B} s_i^a \leqslant x_i \leqslant \min_{a \in B} e_i^a,\ 1 \leqslant i \leqslant d\}$. We assume that $\phi(B)$ is either empty or full-dimensional for every $B \subseteq A$, which is equivalent to assuming that there are no $a, b \in A$ with $\phi(a) \cap \phi(b) \neq \emptyset$ such that $e_i^a = s_i^b$ for some component i. This is without loss of generality, as otherwise we can simply enlarge the boxes in all directions by a positive constant smaller than half the distance between all pairs of non-overlapping boxes.

Let us assume without loss of generality that P is the bounding box of $\cup_{a\in A}\phi(a)$. We show that in time $O(m^{d+1})$ one can determine a collection $\mathscr{D} = \{P_j \subseteq P : j \in D\}$ with $|D| = O(m^d)$ and an arc multimap $\theta : V \times V \to 2^D$ such that $\phi(B) = \cup_{j\in\theta(B)} P_j$ for every $B \subseteq A$. This implies the equivalence of ϕ-connectivity and θ-connectivity on G, and allows to derive Φ in terms of Θ, as $\Phi(u, v) = \cup_{j\in\Theta(u,v)} P_j$ for every $u, v \in V$. Theorem 2 then follows from the application of Lemma 2 and the bound on $|D|$. Note that $\mathscr{D}$ is not required to be a partition of P.

The collection $\mathscr{D}$ and the multimap θ are obtained in time $O(m^{d+1})$ as follows. For each component $1 \leqslant i \leqslant d$, the endpoints s_i^a, e_i^a for all $a \in A$ are sorted, removing duplicates. This gives a monotonic sequence $X_i = (x_k^i)_{1\leqslant k\leqslant r_i}$ of $r_i \leqslant 2m$ points, and a corresponding sequence $Y_i = (y_k^i)_{1\leqslant k<r_i}$ of $r_i - 1$ mid-points $y_k^i = (x_k^i + x_{k+1}^i)/2$. The Cartesian product $Y = Y_1 \times \cdots \times Y_d \subset \mathbb{R}^d$ represents a grid subdivision of P into $|Y| = \Pi_{i=1}^d (r_i - 1) = O(m^d)$ closed boxes P_y, each identified by its center $y \in Y$. For sake of simplicity we directly use y as an index, as Y is a finite set.

For each point $y \in Y$ we derive the subset $M_y = \{a \in A : y \in \phi(a)\}$ of arcs whose box contains y. This problem is known as point location. Each set M_y can be derived in $O(\log^{d-1} m + |M_y|)$ query-time with output-sensitive algorithms (Chazelle 1986; Edelsbrunner and Haring 1986; Chazelle 1988), and the overall time required is at most $O(m^{d+1})$. From the sets M_y one can directly define a collection $\mathscr{D} = \{P_y \subseteq P : y \in Y, M_y \neq \emptyset\}$ and a multimap $\theta : V \times V \to 2^Y$ by $\theta(a) = \{y \in Y : a \in M_y\}$ with the desired properties. □

The collection $\mathscr{D}$ is not minimal. Regions P_y, P_z for which $M_y = M_z$ represent the same intersection, and can be merged. Merging all such regions is equivalent to removing duplicates from a binary matrix, which can be done in linear time (Tomlin and Welch 1986). Note that each resulting region is in general a set of disconnected orthogonal (possibly nonconvex) polyhedra, and in the worst case still $O(m^d)$ such regions are needed.

In the two-dimensional case, the collection obtained after the above post-processing is the planar subdivision defined by the collection of rectilinear polygons (orthogonal polyhedra in $\mathbb{R}^2$) obtained by "cutting" the plane with the sides of the m rectangles $\phi(a)$, for $a \in A$. For this case it is easy to show that the resulting planar subdivision contains at most $2m^2 - 2m + 1$ connected regions, and that this upper bound is tight. Such a tight bound however does not easily generalize to higher dimensions. A problem related to finding the planar subdivision is that of listing all pairwise intersections of m rectangles. Indeed, each subdivision vertex is either a vertex of some rectangle or an intersection of two rectangles. Listing all pairwise intersections is a well-known problem (see e.g. Preparata and Shamos (1985) and the references therein). The straightforward solution of checking all pairs is worst-case optimal, as the number k of intersecting pairs could be as large as $\Omega(m^2)$. However, *output-sensitive* algorithms can solve the problem in $O(m \log m + k)$ time, which is both input and output optimal.

The most common generalization of the pairwise rectangle intersection problem to boxes in $\mathbb{R}^d$ consists in listing all the pairwise box intersections, for which efficient output-sensitive algorithms are known (Six and Wood 1982; Edelsbrunner 1983). However, the space subdivision induced by the boxes cannot be derived by considering only the vertices of *pairwise* intersections. Intersections of up to d boxes must be considered. The subdivision approach considered in this paper, as well as its extensions hereafter, are only intended as a complexity characterization. Although the derived bounds are worst-case optimal, the study of output-sensitive algorithms, as well as of tight upper and lower bounds, would deserve proper attention.

Theorem 2 can be extended to convex polyhedra by defining $\mathscr{D}$ as the cell complex induced by the faces of all polyhedra. Such a complex contains $O(k^d)$ cells and can be derived in $O(k^d)$ time (Edelsbrunner et al. 1986), where k is the total number of inequalities describing the polyhedra. Note that a description as convex hull of n points can be transformed into an inequality description in $O(n \log n + n^{\lfloor d/2 \rfloor})$ time (Chazelle 1993), which is worst-case optimal (McMullen 1970). Theorem 2 can then be extended to nonconvex polyhedra, which are commonly described as a union of convex polyhedra, possibly overlapping. This is done by splitting each arc $a \in A$ into a set of parallel arcs, one for each convex polyhedron defining $\phi(a)$. Nonconvex polyhedra in $\mathbb{R}^2$ and $\mathbb{R}^3$ also allow for a natural face description. In this case, a description in terms of union of convex polyhedra can be found by convex decomposition. Finding a minimum cardinality partition of a simple nonconvex polyhedron in $\mathbb{R}^2$ is easy (Chazelle 1980; Chazelle and Dobkin 1985). In dimension three, and in dimension two if holes are allowed, the problem is NP-hard (Lingas 1982), but approximable with efficient heuristics (see e.g. Chazelle 1984). The decomposition cardinality can decrease if instead of a partition one searches for a cover, but then the minimization becomes NP-hard already in dimension two (O'Rourke and Supowit 1983), and even if holes are not allowed (Culberson and Reckhow 1989).

4 Model invalidation and parameter identification

In this section we apply the results presented in Sect. 3 to model invalidation and parameter identification. Given a discrete dynamical system with transition function $f : X \times P \to X$ and a set of experimental measurements, we say that a parameter $p \in P$ is *consistent* with the measurements if there exists a state trajectory $(x_t)_{t \in \mathbb{N}}$ with $x_{t+1} = f(x_t, p)$ for $t \in \mathbb{N}$ that is compatible with the measurements. Model invalidation aims at proving that no consistent parameter exists. Parameter identification aims at bounding the set $P_F \subseteq P$ of consistent parameters. In our approach, we search for an outer-approximation $\tilde{P}_F \supseteq P_F$, which allows to prove model invalidity whenever $\tilde{P}_F = \emptyset$.

In the remainder, given a discrete time interval $T = \{1, \ldots, \tau\}$, let $M = \{t_1, \ldots, t_e\}$ be the set of time indexes at which the measurements are taken, with $t_1 < t_2 < \cdots < t_e$ and, $t_1 = 1$, $t_e = \tau$ without loss of generality. Furthermore, assume that a state-space discretization $\mathscr{D} = \{X_j : j \in D\}$ of X is defined, and that the corresponding transition parameters $\phi(j, \ell)$ are given for all $j, \ell \in D$. By mapping the measurements onto the nodes of the transition graph $G = TG(D, \phi, T)$, the problem of searching for a consistent parameterization can be relaxed into the problem of searching for a transition path in G that touches the measurement nodes. In this section we formulate this problem more precisely, considering the various forms in which a measurement can be provided.

4.1 Error-bounded state measurements

Real world measurements of continuous quantities always carry some error. If the error can be bounded, the measurement at each $t \in M$ can be given as a subset $X_t^* \subseteq X$ containing the unknown state x_t^*. In the ideal case, the measurement accuracy is sufficient to have, for each measurement X_t^*, a discrete state $j_t^* \in D$ for which $X_t^* \subseteq X_{j_t^*}$. This uniquely identifies a node $v_t^* \in V^t$ of the transition graph, with $v_t^* = v_{j_t^*}^t$. If measurements are available at all time steps ($M = T$), the corresponding measurement nodes directly yield a path $B = (v_t^*)_{t \in T}$ in the transition graph. Model invalidation then amounts to deciding if there exists a state trajectory that lies inside B, which can be relaxed into the problem of checking if B is a transition path, that is, if $\phi(B) \neq \emptyset$. Similarly, parameter identification can be approximated by the set $\tilde{P}_F = \phi(B)$. As shown in Corollary 1, deciding if $\phi(B) \neq \emptyset$ is in general NP-hard, but can be solved efficiently if the sets in the multimap ϕ are polyhedral.

If $M \subset T$, then we have to search for paths in the transition graph that touch all the measurement nodes. Let us solve this case by considering the more general situation in which measurement uncertainty does not yield single nodes, that is, where measurements X_t^* are not strictly contained in any single discrete state. In this case, we can bound each set X_t^* with a set of discrete states (possibly the smallest one), deriving a corresponding set of nodes $V_t^* \subseteq V^t$. Now we can assume without loss of generality that $M = T$, as whenever $t \notin M$ one can simply set $V_t^* = V^t$. Model invalidation can then be relaxed into the problem of finding a path B in the transition graph that touches each set V_t^*, for all $t \in T$. The set of parameters consistent with the measurements can then be approximated by the union of the parameter sets $\phi(B)$ over all such ϕ-feasible paths. Let us formulate these problems in general terms.

Problem 4 Given a layered graph $G = (V, A)$ with layers $V^1, \ldots, V^\tau$, an arc multimap $\phi : V \times V \to 2^P$, and nonempty subsets $V_1^* \subseteq V^1, \ldots, V_\tau^* \subseteq V^\tau$, find a ϕ-feasible path B with $B \cap V_t^* \neq \emptyset$ for all $1 \leqslant t \leqslant \tau$.

Problem 5 Given a layered graph $G = (V, A)$ with layers $V^1, \ldots, V^\tau$, an arc multimap $\phi : V \times V \to 2^P$, and nonempty subsets $V_1^* \subseteq V^1, \cdots, V_\tau^* \subseteq V^\tau$, compute the subset

$$\tilde{P}_F = \bigcup_{B : B \cap V_t^* \neq \emptyset \ \forall t \in T} \phi(B) = \bigcup_{(v_1^*, \ldots, v_\tau^*) \in V_1^* \times \cdots \times V_\tau^*} \bigcap_{k=1}^{\tau-1} \Phi(v_k^*, v_{k+1}^*).$$

Problems 4 and 5 are NP-hard in general, even if the sets ϕ are axis-aligned boxes (see Theorem 1), as they include as special cases Problems 2 and 3 respectively. Indeed, these problems are actually equivalent. Due to the layered structure of the graph, the combinatorial explosion in the above equation can be eliminated by defining, for every $v \in V^*_{t_k}$ with $k > 1$, the recursion

$$\omega(v) = \bigcup_{w \in V^*_{t_{k-1}}} \omega(w) \cap \Phi(w, v),$$

with $\omega(v) = P$ for every $v \in V^*_{t_1}$. Due to this recursion, given the multimap Φ we can compute the parameter approximation as $\tilde{P}_F = \cup_{v \in V^*_{t_e}} \omega(v)$. This allows to apply Theorem 2, as well as its extensions for the case where ϕ is defined by polyhedral sets.

The equivalence of these problems can also be seen as follows. Construct a digraph $\tilde{G}$ by copying G, discarding all nodes in $V^t \setminus V^*_t$, for every $t \in T$, and adding an initial layer $V^0 = \{s\}$ and a final layer $V^{\tau+1} = \{d\}$. Finally, construct a multimap θ by copying ϕ for the arcs that are both in G and $\tilde{G}$, and by setting $\phi(s, u) = P$ for all $u \in V^1$ and $\phi(v, d) = P$ for all $v \in V^\tau$. It is then easy to see that Problem 4 admits a solution if and only if there exists a θ-feasible path in $\tilde{G}$, and that $\tilde{P}_F = \Phi(s, d)$.

4.2 Output functions and unbounded error measurements

In many applications the state cannot be directly measured, as one can only observe some output function of state and parameters. In this case, to apply our framework one needs first to bound the state value, either by inverting the output function (if possible) or by state-estimation techniques. In case of a polynomial output function, the techniques for parameter identification described in Borchers et al. (2009) can be applied. A further important remark is that in many real-case scenarios the measurement error cannot be bounded, due to its stochastic nature. In that case a measurement is given as a probability distribution, or, when not possible, as a *confidence interval*, a set that contains the unknown value with a given probability. In this case our approach remains applicable, but the validity of the corresponding model invalidation and parameter identification results assume a probabilistic sense.

5 Experimental design

If the given measurements yield an unsatisfactory parameter estimate, by repeating the experiment the measurement errors might be reduced, hopefully improving the results. However, experiments cost time and money, and this may be inefficient. Moreover, the poor estimate could be inherent, if the trajectory followed by the experiment is robust to parameter changes, in which case the experiment has to be changed. The goal of experimental design is to devise new experiments, or to support the practitioners in doing so, so as to optimize some target, as e.g. to optimize the resulting parameter identification, or to minimize the cost of the experiments required to obtain a given parameter identification quality.

Numerical analysis and statistical approaches to experimental design are well known for linear dynamical systems (Pázman 1986; Atkinson and Donev 1992), and during the last decade the nonlinear case has attracted significant attention (see e.g.

Bauer et al. (2000), Schittkowski (2007) and the references therein). In this section we consider experimental design from the viewpoint of transition graphs, defining some simple combinatorial optimization problems. In the framework considered here, devising a new experiment amounts to deciding the initial conditions, the duration of the experiment, and the measurement times. Experimental design is however particularly relevant for systems in which an input can be applied during the experiment. The extension of this framework to such systems is out of the scope of this paper, and will be subject of future work.

5.1 Minimum and maximum cardinality ϕ-feasible paths

Consider a graph $G = (V, A)$ and an arc multimap $\phi : V \times V \to 2^P$, with P finite. For a given path B in G, the intersection $\phi(B)$, when G is a transition graph, contains all the parameters $p \in P$ that are consistent with the path. If the experiment were to follow a trajectory lying inside B, then $\phi(B)$ would be the resulting parameter estimate. Given two nodes $s, d \in V$, it is then natural to consider the two problems of finding a ϕ-feasible path B with minimum and maximum cardinality $|\phi(B)|$, respectively. Such paths correspond to the best and worst possible outcome of an experiment, so that when setting up the experiment we would like to stay as close as possible to the first, and as far as possible from the second. Let us define these problems explicitly.

Problem 6 Given a graph $G = (V, A)$, an arc multimap $\phi : V \times V \to 2^P$, and two nodes $s, d \in V$, find a ϕ-feasible s–d path B that minimizes $|\phi(B)|$.

Problem 7 Given a graph $G = (V, A)$, an arc multimap $\phi : V \times V \to 2^P$, and two nodes $s, d \in V$, find an s–d path B that maximizes $|\phi(B)|$.

Note that in Problem 7, due to its objective function, we don't need to explicitly require ϕ-feasibility of the path B. As a simple corollary to Lemma 3, obtained by adding r fictitious elements to the groundset P, we have that finding a ϕ-feasible path B with $|\phi(B)| \leqslant r$ is NP-complete, for any constant r. As a consequence, Problem 6 is NP-hard, even if G is a layered graph and $|P| = \Omega(\sqrt[k]{|V|})$ for any $k > 0$. Hereafter we provide an inapproximability result for Problem 7.

Theorem 3 *Problem 7 cannot be approximated within a factor $|P|^{1-\varepsilon}$ for any $\varepsilon > 0$, unless* NP $=$ ZPP*, even if G is a layered graph with at most two nodes per layer, and P is explicit and has polynomial cardinality in the size of G, where* ZPP *is the class of problems that admit a randomized algorithm with zero probability of error.*

Proof We proceed by reduction from Maximum Independent Set (MIS). Given a graph $H = (U, E)$, in MIS one has to find an independent set (a subset of pairwise non-adjacent nodes) with maximum cardinality. MIS cannot be approximated within a factor $|U|^{1-\varepsilon}$ for any $\varepsilon > 0$, unless NP $=$ ZPP (Håstad 1999). This inapproximability factor will directly carry over to Problem 7.

Given $H = (U, E)$, with $U = \{u_1, \ldots, u_n\}$, we build a layered digraph $G = (V, A)$ with $n + 2$ layers and at most two nodes per layer as follows. The first and last layer contain a single node each, and are denoted $V^0 = \{s\}$ and $V^{n+1} = \{d\}$ respectively.

Then, for each $1 \leqslant t \leqslant n$ we build a layer $V^t = \{v_Y^t, v_N^t\}$ containing two nodes. The arc set is $A = \{(w, z) : w \in V^t, z \in V^{t+1}, 0 \leqslant t \leqslant n\}$, with every pair of consecutive layers inducing a complete bipartite digraph. The groundset is $P = U$, and the arc multimap $\phi : V \times V \to 2^P$ is defined as follows. We set $\phi(s, w) = U$ for every $w \in V^1$. Then, we set $\phi(v_Y^t, w) = U \setminus \delta(u_t)$ and $\phi(v_N^t, w) = U \setminus \{u_t\}$ for every $1 \leqslant t \leqslant n$ and $w \in V^{t+1}$, where $\delta(u) = \{w \in U : \{u, w\} \in E\}$ is the set of nodes adjacent to u in H.

Any s–d path B must touch either v_Y^t or v_N^t for every node $u_t \in U$. We can then correspondingly partition the nodes U into two subsets T_B^Y and T_B^N. It is easy to see that $\phi(B) \subseteq U$ is an independent set, with $\phi(B) = T_B^Y$ if and only if T_B^Y is an independent set as well. Conversely, given an independent set T we can easily construct a path B with $\phi(B) = T$ by setting $T_B^Y = T$ and $T_B^N = U \setminus T$. Given the equivalence of the objective functions, the result directly follows. □

At this point, we have convinced ourselves that finding minimum and maximum cardinality ϕ-feasible paths is hard, unless the multimap and the graph have some special structure. Hereafter we show that Problem 7 is polynomial-time solvable if P has logarithmic cardinality in the size of the graph.

Proposition 6 *If $|P| = O(\log |A|)$, then Problem 7 can be solved in polynomial time.*

Proof Recall that, as outlined in Sect. 3.3, finding an s–d path B with $Q \subseteq \phi(B)$ for any given $Q \subseteq P$ can be easily done by searching for an s–d path in G_Q. Therefore, an s–d path B maximizing $|\phi(B)|$ can be found by enumerating all subsets $Q \subseteq P$, which are polynomially many if $|P| = O(\log m)$, and selecting the largest $Q \subseteq P$ for which there exists an s–d path in G_Q. □

This proof would be extended to Problem 6 if one could find in polynomial-time an s–d path B in G_Q with $\phi(B) = Q$. As mentioned in Sect. 3.4, this is solvable in polynomial time whenever finding a ϕ-infeasible s–d path is. Proposition 5 shows that this holds when $|P| = \log k \log m$ for some $k > 0$. Whether this is true also for the case $|P| = O(\log m)$ remains open. Note that if for some instance class Problem 6 is tractable, then clearly finding a ϕ-infeasibile s–d path becomes tractable as well.

5.2 Worst-case initial conditions and additional measurement times

Let us assume that exact state measurements can be obtained at every time step, and that the duration of the experiment is bounded by $T \in \mathbb{N}$. Designing an optimal worst-case experiment consists in deciding the initial conditions for which the resulting parameter estimate will be in the worst case as good as possible. Let us assume that P is finite and that we can measure the quality of a parameter estimate $\tilde{P}_F$ by its cardinality. Given the transition graph $TG(D, \phi, T)$, designing an optimal worst-case experiment can be formulated as

$$\arg \min_{s \in V^1} \max_{d \in V^T} |\Phi(s, d)|.$$

Similarly, given an experiment with measurements x_t^* for $t \in M$, deciding whether a measurement time should be added can be cast as a min-max problem. Let $\tilde{P}_F$ be the estimate given by the current experiment, as defined in Sect. 4.1. The best additional measurement time is the time $t \in T \setminus M$ for which the worst-case among all possible realizations $w \in V^t$ of the resulting parameter estimate is as good as possible. This can be formulated as

$$\arg \min_{t \in T \setminus M} \max_{w \in V^t} |\tilde{P}_F \cap \Phi(v_{t^-}^*, w) \cap \Phi(w, v_{t^+}^*)|,$$

where t^- and t^+ are the closest measurement times before and after t respectively. Both problems can therefore be solved by evaluating the multimap Φ. These problems can be easily extended to account for a weight function $f(Q)$ evaluating the quality of a parameter estimate $Q \subset P$, as well as to account for bounded measurement errors.

6 Conclusions and future work

In this paper we have considered an interesting combinatorial framework arising from parameter identification and model validation of dynamical systems, which allows also to formulate some experimental design problems. Given a graph and nonempty sets associated to the arcs, we studied how hard it is to find a path for which the sets associated to the arcs of the path have non-empty intersection, and how hard it is to optimize the cardinality of such intersections. This framework has strong connections with dynamic programming, as it allows to add a "memory" to the path, and the resulting combinatorial problems are interesting on their own.

This study focused on establishing the computational complexity of such connectivity problems, considering different possible assumptions on the structure of the sets associated to the arcs. Open questions to be investigated in our future work include an extension of the complexity results to other interesting structures, as well as the development of efficient algorithms for treating the polynomial-solvable cases, and of good heuristics for the hard ones.

From the viewpoint of dynamical systems, the framework considered here provides a necessary condition for the existence of a trajectory in the state space. It would be interesting to find sufficient conditions under which the trajectory can be guaranteed, possibly within a given error. Another important direction of study concerns the extension of this framework to dynamical systems in which a control input can be applied, which is of particular importance in experimental design.

References

Atkinson AC, Donev AN (1992) Optimum experimental designs. Oxford statistical science series. Oxford University Press, Oxford

Bauer I, Bock HG, Körkel S, Schlöder JP (2000) Numerical methods for optimum experimental design in DAE systems. J Comput Appl Math 120(1–2):1–25

Borchers S, Rumschinski P, Bosio S, Weismantel R, Findeisen R (2009) Model discrimination and parameter estimation for dynamical biochemical reaction networks. In: 15th IFAC symposium on system identification

Chazelle B (1980) Computational geometry and convexity. Ph.D. thesis, Yale University

Chazelle B (1984) Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. SIAM J Comput 13(3):488–507

Chazelle B (1986) Filtering search: a new approach to query-answering. SIAM J Comput 15(3):703–724

Chazelle B (1988) A functional approach to data structures and its use in multidimensional searching. SIAM J Comput 17(3):427–462

Chazelle B (1993) An optimal convex hull algorithm in any fixed dimension. Discret Comput Geom 10: 377–409

Chazelle B, Dobkin DP (1985) Optimal convex decompositions. In: Toussaint GT (ed) Computational Geometry. North-Holland, Amsterdam, pp 63–133

Culberson JC, Reckhow RA (1989) Orthogonally convex coverings of orthogonal polygons without holes. J Comput Syst Sci 39(2):166–204

Edelsbrunner H (1983) A new approach to rectangle intersections. I, II Int J Comput Math 13(3–4):209–219, 221–229

Edelsbrunner H, Haring G, Hilbert D (1986) Rectangular point location in d dimensions with applications. Comput J 29(1):76–82

Edelsbrunner H, O'Rourke J, Seidel R (1986) Constructing arrangements of lines and hyperplanes with applications. SIAM J Comput 15(2):341–363

Evans ND, Chappell MJ, Chapman MJ, Godfrey KR (2004) Structural indistinguishability between uncontrolled (autonomous) nonlinear analytic systems. Automatica 40:1947–1953

Håstad J (1999) Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Math 182(1):105–142

Kuepfer L, Sauer U, Parrilo PA (2007) Efficient classification of complete parameter regions based on semidefinite programming. BMC Bioinf 8:12

Lingas A (1982) The power of non-rectilinear holes. In: 9th colloquium on automata, languages and programming, Lecture notes in computer science, vol 140. pp 369–383

Marquardt DW (1963) An algorithm for least-squares of nonlinear parameters. SIAM J Appl Math 11: 431–441

McMullen P (1970) The maximum numbers of faces of a convex polytope. Mathematika 17:179–184

O'Rourke J, Supowit KJ (1983) Some NP-hard polygon decomposition problems. IEEE Trans Inf Theory 29(2):181–190

Pázman A (1986) Foundations of optimum experimental design. Reidel, Dordrecht

Prajna S (2006) Barrier certificates for nonlinear model validation. Automatica 42:117–126

Preparata F, Shamos MI (1985) Computational geometry. Spinger, New York

Ramsay JO, Hooker G, Campbell D, Cao J (2007) Parameter estimation for differential equations: a generalized smoothing approach. J R Stat Soc Ser B 69(5):741–796

Robert CP, Casella G (2004) Monte Carlo statistical methods. Springer, Berlin

Schittkowski K (2007) Experimental design tools for ordinary and algebraic differential equations. Ind Eng Chem Res 46(26):9137–9147

Schnell S, Chappell MJ, Evans ND, Roussel MR (2006) The mechanism distinguishability problem in biochemical kinetics: the single-enzyme, single-substrate reaction as a case study. CR Biol 329:51–61

Schrijver A (2003) Combinatorial optimization. Polyhedra and efficiency. Algorithms and combinatorics, vol 24. Springer-Verlag, Berlin

Six HW, Wood D (1982) Counting and reporting intersections of d-ranges. IEEE Trans Comput 31(3): 181–187

Smith RS, Doyle JC (1992) Model validation: a connection between robust control and identification. IEEE Trans Autom Control 37(7):942–952

Tomlin JA, Welch JS (1986) Finding duplicate rows in a linear programming model. Oper Res Lett 5(1): 7–11

Valiant LG, Vazirani VV (1985) NP is as easy as detecting unique solutions. In: 17th ACM symposium on theory of computing. pp 458–463