On The Complexity of Bounded Time Reachability for Piecewise Affine Systems*

Hugo Bazille[3], Olivier Bournez[1], Walid Gomaa[2,4], and Amaury Pouly[1]

[1] École Polytechnique, LIX, 91128 Palaiseau Cedex, France
[2] Egypt Japan University of Science and Technology, CSE, Alexandria, Egypt
[3] ENS Cachan/Bretagne et Université Rennes 1, France
[4] Faculty of Engineering, Alexandria University, Alexandria, Egypt

Abstract. Reachability for piecewise affine systems is known to be undecidable, starting from dimension 2. In this paper we investigate the exact complexity of several decidable variants of reachability and control questions for piecewise affine systems. We show in particular that the region to region bounded time versions leads to NP-complete or co-NP-complete problems, starting from dimension 2.

1 Introduction

A (discrete time) dynamical system $\mathcal{H}$ is given by some space X and a function $f : X \to X$. A trajectory of the system starting from x_0 is a sequence $x_0, x_1, x_2, \dots$ etc., with $x_{i+1} = f(x_i) = f^{[i+1]}(x_0)$ where $f^{[i]}$ stands for i^{th} iterate of f. A crucial problem in such systems is the *reachability question*: given a system $\mathcal{H}$ and $R_0, R \subseteq X$, determine if there is a trajectory starting from a point of R_0 that falls in R. Reachabilty is known to be *undecidable* for very simple functions f. Indeed, it is well-known that various types of dynamical systems, such as hybrid systems, piecewise affine systems, or saturated linear systems, can simulate Turing machines, see e.g., [1,2,3,4].

This question is at the heart of *verification* of systems. Indeed, a safety property corresponds to the determination if there is a trajectory starting from some set R_0 of possible initial states to the set R of bad states. The industrial and economical impact of having efficient computer tools, that are able to guarantee that a given system does satisfy its specification, have indeed generated very important literature. Particularly, many undecidability and complexity-theoretic results about the hardness of verification of safety properties have been obtained in the model checking community. However, as far as we know, the exact complexity of *natural restrictions* of the reachability question for systems as simple as piecewise affine maps are not known, despite their practical interest.

Indeed, existing results mainly focus on the frontier between decidability and undecidability. For example, it is known that reachability is undecidable for piecewise constant derivative systems of dimension 3, whereas it is decidable for dimension 2 [5]. It is known that piecewise affine maps of dimension 2 can

* This work was partially supported by DGA Project CALCULS.

J. Ouaknine, I. Potapov, and J. Worrell (Eds.): RP 2014, LNCS 8762, pp. 20–31, 2014.

simulate Turing machines [6], whereas the question for dimension 1 is still open and can be related to other natural problems [7,8,9]. Variations of such problems over the integers have recently been investigated [10].

Some complexity facts follow immediately from these (un)computability results: for example, point to point bounded time reachability for piecewise affine maps is P-complete as it corresponds to configuration to configuration reachability for Turing machines.

However, their remain many natural variants of reachability questions which complexity have not yet been established.

For example, in the context of verification, point to point reachability is often not sufficient. On the contrary, region to region reachability is a more general question, which complexity do not follow from existing results.

In this paper we choose to restrict to the case of piecewise affine maps and we consider the following natural variant of the problem.

Continuous Bounded Time. we want to know if region R is reached in less than some prescribed time T, with f assumed to be continuous

Remark 1. We consider piecewise affine maps over the domain $[0,1]^d$, that is to say we do not restrict to the integers as in [10]. That would make the problem rather different. We also assume f to be continuous which makes the hardness result more natural.

In an orthogonal way, control of systems or constructions of controllers for systems often yield to dual questions. Instead of asking if some trajectory reaches region R, one wants to know if all trajectories reach R. The questions of stability, mortality, or nilpotence for piecewise affine maps and saturated linear systems have been established in [11]. Still in this context, the complexity of the problem when restricting to bounded time or fixed precision is not known.

This paper provides an exact characterization of the *algorithmic complexity* of those two types of reachability for discrete time dynamical systems. Let PAF_d denote the set of piecewise-affine *continuous* functions over $[0,1]^d$. At the end we get the following picture.

Problem: `REACH-REGION`
Inputs: a continuous PAF_d f and two regions R_0 and R in $\mathrm{dom}(f)$
Question: $\exists x_0 \in R_0, t \in \mathbb{N}, f^{[t]}(x_0) \in R$?

Theorem 2 ([6]). *Problem `REACH-REGION` is undecidable (and recursively enumerable-complete).*

Problem: `CONTROL-REGION`
Inputs: a continuous PAF_d f and two regions R_0 and R in $\mathrm{dom}(f)$
Question: $\forall x_0 \in R_0, \exists t \in \mathbb{N}, f^{[t]}(x_0) \in R$?

Theorem 3 ([11]). *Problem `CONTROL-REGION` is undecidable (and co-recursively enumerable complete) for $d \geqslant 2$.*

Problem: `REACH-REGION-TIME`
Inputs: a time $T \in \mathbb{N}$ in unary, a continuous PAF_d f and two regions R_0 and R in $\mathrm{dom}(f)$
Question: $\exists x_0 \in R_0, \exists t \leqslant T, f^{[t]}(x_0) \in R$?

Theorem 4. *Problem* `REACH-REGION-TIME` *is* NP*-complete for* $d \geqslant 2$.

Problem: `CONTROL-REGION-TIME`
Inputs: a time $T \in \mathbb{N}$ in unary, a continuous PAF_d f and two regions R_0 and R in $\mathrm{dom}(f)$
Question: $\forall x_0 \in R_0, \exists t \leqslant T, f^{[t]}(x_0) \in R$?

Theorem 5. *Problem* `CONTROL-REGION-TIME` *is co*NP*-complete for* $d \geqslant 2$.

All our problems are region to region reachability questions, which requires new proof techniques.

Indeed, classical tricks to simulate a Turing machine using a piecewise affine maps encode a Turing machine configuration by a point, and assume that all the points of the trajectories encode (possibly ultimately) valid Turing machines configurations.

This is not a problem in the context of point to point reachability, but this can not be extended to region to region reachability. Indeed, a (non-trivial) region consists mostly in invalid points: mostly all points do not correspond to encoding of Turing machines for all the considered encodings in above references.

In order to establish hardness results, the trajectories of all (valid and invalid) points must be carefully controlled. This turns out not to be easily possible using the classical encodings.

Let us insist on the fact that we restrict to continuous dynamics. In this context, this is an additional source of difficulties. Indeed, such a system must necessarily have a sub-region which dynamics cannot be easily interpreted in terms of configurations.

In other words, the difficulty is in dealing with points and trajectories not corresponding to valid configurations or evolutions.

2 Preliminaries

2.1 Notations

The set of non-negative integers is denoted $\mathbb{N}$ and the set of the first n naturals is denoted $\mathbb{N}_n = \{0, 1, \ldots, n-1\}$. For any finite set Σ, let Σ^* denote the set of finite words over Σ. For any word $w \in \Sigma^*$, let $|w|$ denote the length of w. Finally, let λ denote the empty word. If w is a word, let w_1 denote its first character, w_2 the second one and so on. For any $i, j \in \mathbb{N}$, let $w_{i\ldots j}$ denote the subword $w_i w_{i+1} \ldots w_j$. For any $\sigma \in \Sigma$, and $k \in \mathbb{N}$, let σ^k denote the word of length k where all symbols are σ. For any function f, let $f \upharpoonright E$ denote the restriction of f to E and let $\mathrm{dom}(f)$ denote the domain of definition of f. For any set $S \in \mathbb{R}^d$, $\mathring{S}$ denotes the interior of S.

2.2 Piecewise Affine Functions

Let I denote the unit interval $[0,1]$. Let $d \in \mathbb{N}$. A convex closed polyhedron in the space I^d is the solution set of some linear system of inequalities:

$$A\mathbf{x} \leq \mathbf{b} \tag{1}$$

with coefficient matrix A and offset vector $\mathbf{b}$. Let PAF_d denote the set of piecewise-affine continuous functions over I^d. For any $f\colon I^d \to I^d$ in PAF_d, f satisfies:

- f is continuous,
- there exists a sequence $(P_i)_{1\leq i\leq p}$ of convex closed polyhedron with nonempty interior such that $f_i = f \upharpoonright P_i$ is affine, $I^d = \bigcup_{i=1}^{p} P_i$ and $\mathring{P}_i \cap \mathring{P}_j = \emptyset$ for $i \neq j$.

In the following discussion we will always assume that any polyhedron P can be defined by a finite set of linear inequalities, where all the elements of A and $\mathbf{b}$ in (1) are all rationals. A polyhedron over which f is affine we also be called a region.

2.3 Decision Problems

In this paper, we will show hardness results by reduction to known hard problems. We give the statement of these latter problems in the following.

Problem: `SUBSET-SUM`
Inputs: a goal $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$.
Question: $\exists I \subseteq \{1, \ldots, n\}, \sum_{i\in I} A_i = B$?

Theorem 6 ([12]). *`SUBSET-SUM` is NP-complete.*

Problem: `NOSUBSET-SUM`
Inputs: a witness $B \in \mathbb{N}$ and integers $A_1, \ldots, A_n \in \mathbb{N}$.
Question: $\forall I \subseteq \{1, \ldots, n\}, \sum_{i\in I} A_i \neq B$?

Theorem 7. *`NOSUBSET-SUM` is coNP-complete.*

Proof. Basically the same proof as Theorem 6

3 Hardness of Bounded Time Reachability

In this section, we will show that `REACH-REGION-TIME` is an NP-hard problem by reducing it to `SUBSET-SUM`.

3.1 Solving SUBSET-SUM by Iteration

We will now show how to solve the SUBSET-SUM problem with a simple iterated function. Consider an instance $\mathcal{I} = (B, A_1, \ldots, A_n)$ of SUBSET-SUM. We will need to introduce some notions before defining our piecewise affine function. Our first notion is that of configurations, which represent partial summation of the number for a given choice of I.

Remark 8. Without loss of generality, we will only consider instances where $A_i \leqslant B$, for all i. Indeed, if $A_i > B$, it will never be part of a subset sum and so we can simply remove this variable from the problem. This ensures that $A_i < B+1$ in everything that follows.

Definition 9 (Configuration). *A configuration of* $\mathcal{I}$ *is a tuple* $(i, \sigma, \varepsilon_i, \ldots, \varepsilon_n)$ *where* $i \in \{1, \ldots, n+1\}$, $\sigma \in \{0, \ldots, B+1\}$, $\varepsilon_i \in \{0, 1\}$ *for all* i. *Let* $\mathcal{C}_\mathcal{I}$ *be the set of all configurations of* $\mathcal{I}$.

The intuitive understanding of a configuration, made formal in the next definition, is the following: $(i, \sigma, \varepsilon_i, \ldots, \varepsilon_n)$ represents a situation where after having summed a subset of $\{A_1, \ldots, A_{i-1}\}$, we got a sum σ and ε_j is 1 if and only if we are to pick A_j in the future.

Definition 10 (Transition function). *The transition function* $T_\mathcal{I} : \mathcal{C}_\mathcal{I} \to \mathcal{C}_\mathcal{I}$, *is defined as follows:*

$$T_\mathcal{I}(i, \sigma, \varepsilon_i, \ldots, \varepsilon_n) = \begin{cases} (i, \sigma) & \text{if } i = n+1 \\ (i+1, \min\left(B+1, \sigma + \varepsilon_i A_i\right), \varepsilon_{i+1}, \ldots, \varepsilon_n) & \text{otherwise} \end{cases}$$

It should be clear, by definition of a subset sum that we have the following simulation result.

Lemma 11. *For any configuration* $c = (i, \sigma, \varepsilon_i, \ldots, \varepsilon_n)$ *and* $k \in \{0, \ldots, n+1-i\}$,

$$T_\mathcal{I}^{[k]}(c) = \left(i+k, \min\left(B+1, \sigma + \Sigma_{j=i}^{i+k-1} \varepsilon_j A_j\right), \varepsilon_{i+k}, \ldots, \varepsilon_n\right)$$

Proof. By induction.

A consequence of this simulation by iterated function, is that we can reformulate satisfiability in terms of reachability.

Lemma 12. $\mathcal{I}$ *is a satisfiable instance (*i.e., *admits a subset sum) if and only if there exists a configuration* $c = (1, 0, \varepsilon_1, \ldots, \varepsilon_n) \in \mathcal{C}_\mathcal{I}$ *such that* $T_\mathcal{I}^{[n]}(c) = (n+1, B)$.

3.2 Solving a SUBSET-SUM Problem with a Piecewise Affine Function

In this section, we explain how to simulate the function $T_\mathcal{I}$ using a piecewise affine function and some encoding of the configurations for a given $\mathcal{I} = (B, A_1, \ldots, A_n)$.

Definition 13 (Encoding). *Define* $p = \lceil \log_2(n+2) \rceil$, $\omega = \lceil \log_2(B+2) \rceil$, $q = p + \omega + 1$ *and* $\beta = 5$. *Also define* $0^\star = 1$ *and* $1^\star = 4$. *For any configuration* $c = (i, \sigma, \varepsilon_i, \ldots, \varepsilon_n)$, *define the encoding of* c *as follows:*

$$\langle c \rangle = \left(i2^{-p} + \sigma 2^{-q}, 0^\star \beta^{-n-1} + \sum_{j=i}^{n} \varepsilon_i^\star \beta^{-i} \right)$$

Also define the following regions for any $i \in \{1, \ldots, n+1\}$ *and* $\alpha \in \{0, \ldots, \beta - 1\}$*:*

$$R_0 = [0, 2^{-p-1}] \times [0, 1] \qquad R_i = [i2^{-p}, i2^{-p} + 2^{-p-1}] \times [0, \beta^{-i+1}] \quad (i \geqslant 1)$$

$$R_{i,\alpha} = \left[i2^{-p}, i2^{-p} + 2^{-p-1}\right] \times \left[\alpha\beta^{-i}, (\alpha+1)\beta^{-i}\right] \qquad R_i = \cup_{\alpha \in \mathbb{N}_\beta} R_{i,\alpha}$$

$$R^{lin}_{i,1^\star} = \left[i2^{-p}, i2^{-p} + (B + 1 - A_i)2^{-q}\right] \times \left[1^\star\beta^{-i}, 5\beta^{-i}\right] \qquad R^{sat}_{i,1^\star} = R_{i,1^\star} \setminus R^{lin}_{i,1^\star}$$

The rationale behind this encoding is the following. On the first coordinate we put the current number i, "shifted" by as many bits as necessary to be between 0 and 1. Following i, we put σ, also shifted by as many bits as necessary. Notice that there is one padding bit between i and σ; this is necessary to make the regions R_i disjoint from each other. On the second component, we put the description of the variables ε_j, written in basis β to get some "space" between consecutive encodings. The choice of the value 1 and 4 for the encoding of 0 and 1, although not crucial, has been made to simplify the proof as much as possible.

The region R_0 is for initialization purposes and is defined differently for the other R_i. The regions R_i correspond to the different values of i in the configuration (the current number). Each R_i is further divided into the $R_{i,\alpha}$ which correspond to all the possible values of the next ε variable (recall that it is encoded in basis β). In the special case of $\varepsilon = 1$, we cut the region $R_{i,1^\star}$ into a linear part and a saturated part. This is needed to emulate the $\max(\sigma + A_i, B + 1)$ in Definition 10: the linear part corresponds to $\sigma + A_i$ and the saturated part to $B + 1$.

Figure 1 and Figure 2 give a graphical representation of the regions.

Lemma 14. *For any configuration* $c = (i, \sigma, \varepsilon_i, \ldots, \varepsilon_n)$, *if* $i = n + 1$ *then* $\langle c \rangle \in R_{n+1,0^\star}$, *otherwise* $\langle c \rangle \in R_{i,\varepsilon_i^\star}$. *Furthermore if* $\varepsilon_i = 1$ *and* $\sigma + A_i \leqslant B + 1$, *then* $\langle c \rangle \in R^{lin}_{i,1^\star}$, *otherwise* $\langle c \rangle \in R^{sat}_{i,1^\star}$.

We can now define a piecewise affine function which will mimic the behavior of $T_\mathcal{I}$. The region R_0 is here to ensure that we start from a "clean" value on the first coordinate.

Definition 15 (Piecewise affine simulation).

$$f_\mathcal{I}(a, b) = \begin{cases} (2^{-p}, b) & \textit{if } (a,b) \in R_0 \\ (a, b) & \textit{if } (a,b) \in R_{n+1} \\ (a + 2^{-p}, b - 0^\star\beta^{-i}) & \textit{if } (a,b) \in R_{i,0^\star} \\ (a + 2^{-p} + A_i 2^{-q}, b - 1^\star\beta^{-i}) & \textit{if } (a,b) \in R^{lin}_{i,1^\star} \\ ((i+1)2^{-p} + (B+1)2^{-q}, b - 1^\star\beta^{-i}) & \textit{if } (a,b) \in R^{sat}_{i,1^\star} \end{cases}$$

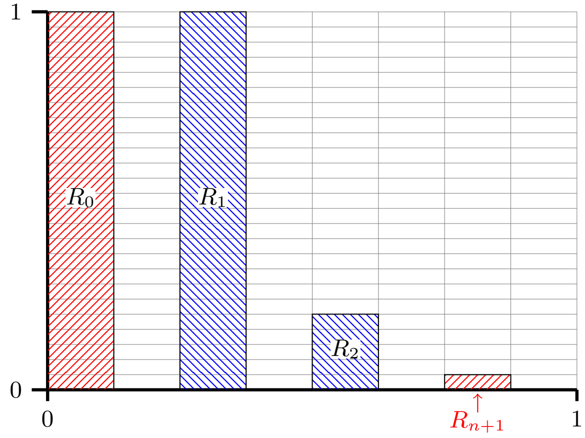

Fig. 1. Graphical representation of the regions

Lemma 16 (Simulation is correct). *For any configuration* $c \in \mathcal{C}_\mathcal{I}$, $\langle T_\mathcal{I}(c)\rangle = f_\mathcal{I}(\langle c\rangle)$.

Notice that we have defined f over a subset of the entire space and it is clear that this subspace is not stable in any way[1]. In order to match the definition of a piecewise affine function, we need to define f over the entire space or a stable subspace (which contains the initial region). We follow this second approach and extend the definition of f on some more regions. More precisely, we need to define f over $R_i = R_{i,0} \cup R_{i,1} \cup R_{i,2} \cup R_{i,3} \cup R_{i,3}$ and at the moment we have only defined f over $R_{i,1} = R_{i,0^\star}$ and $R_{i,4} = R_{i,1^\star}$. Also note that $R_{i,4} = R_{i,4}^{lin} \cup R_{i,4}^{sat}$ and we define f separately on those two subregions.

In order to correctly and continuously extend f, we will need to further split the region $R_{i,3}$ into linear and saturated parts $R_{i,3}^{slo}$ and $R_{i,3}^{shi}$: see Figure 2.

Definition 17 (Extended region splitting). *For* $i \in \{1, \ldots, n\}$ *and* $\alpha \in \{0, \ldots, \beta - 1\}$, *define:*

$$R_{i,3}^{lin} = R_{i,3} \cap \left\{ (a,b) \,\middle|\, b\beta^i - 3 \leqslant \frac{2^{-p-1} + i2^{-p} - a}{2^{-p-1} - (B+1-A_i)2^{-q}} \right\} \qquad R_{i,3}^{sat} = R_{i,3} \setminus R_{i,3}^{lin}$$

It should be clear by definition that $R_{i,3}^{sat} = R_{i,3}^{slo} \cup R_{i,3}^{shi}$ and that the two subregions are disjoint except on the border.

[1] For example $R_{1,1} \subseteq f(R_0)$ but f is not defined over $R_{1,1}$.

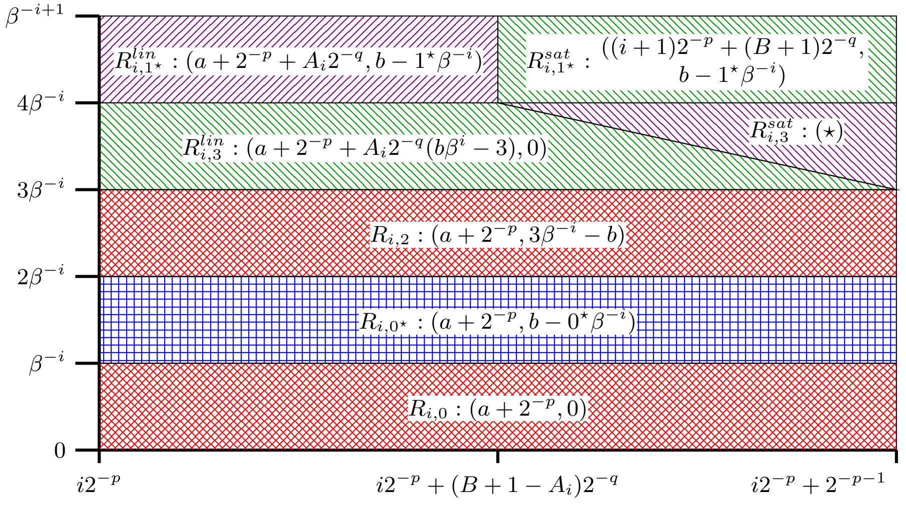

Fig. 2. Zoom on one R_i with the subregions and formulas

Definition 18 (Extended piecewise affine simulation).

$$f_{\mathcal{I}}(a,b) = \begin{cases} (a + 2^{-p}, 0) & \textit{if } (a,b) \in R_{i,0} \\ (a + 2^{-p}, 3\beta^{-i} - b) & \textit{if } (a,b) \in R_{i,2} \\ (a + 2^{-p} + A_i 2^{-q}(b\beta^i - 3), 0) & \textit{if } (a,b) \in R^{lin}_{i,3} \\ ((i + \frac{3}{2})2^{-p} - (b\beta^i - 3)(2^{-p-1} - (B+1)2^{-q}), 0) & \textit{if } (a,b) \in R^{sat}_{i,3} \end{cases}$$

This extension was carefully chosen for its properties. In particular, we will see that f is still continuous, which is a requirement of the piecewise affine functions we consider. Also, the domain of definition of f is f-stable (i.e. $f(\operatorname{dom} f) \subseteq \operatorname{dom} f$). And finally, we will see that f is somehow "reversible".

Lemma 19 (Simulation is continuous). *For any $i \in \{1, \ldots, n\}$, $f_{\mathcal{I}}(R_i)$ is well-defined and continuous over R_i.*

Lemma 20 (Simulation is stable). *For any $i \in \{1, \ldots, n\}$, $f_{\mathcal{I}}(R_i) \subseteq R_{i+1}$. Furthermore, $f(R_0) \subseteq R_1$ and $f(R_{n+1}) \subseteq R_{n+1}$.*

We now get to the core lemma of the simulation. Up to this point, we were only interested in forward simulation: that is given a point, what are the iterates of x. In order to prove the NP-hardness result, we need a backward result: given a point, what are the possible preimages of it. To this end, we introduce new subregions of the R_i which we call *unsaturated*. Intuitively, R_i^{unsat} corresponds to the encodings where $\sigma \leqslant B$, that is the sum did not saturate at $B+1$. We also introduce the R_{fin} region which will be the region to reach. We will be interested in the preimages of R_{fin}.

Definition 21 (Unsaturated regions). *For $i \in \{1, \ldots, n+1\}$, define*

$$R_i^{unsat} = [i2^{-p}, i2^{-p} + B2^{-q}] \times [\beta^{-n-1}, \beta^{-i+1} - \beta^{-n-1}]$$

$$R_{fin} = [(n+1)2^{-p} + B2^{-q} - 2^{-q-1}, (n+1)2^{-p} + B2^{-q}] \times [\beta^{-n-1}, 2\beta^{-n-1}]$$

Lemma 22 (Simulation is reversible). *Let $i \in \{2, \ldots, n\}$ and $(a, b) \in R_i^{unsat}$ Then the only points $\mathbf{x}$ such that $f_{\mathcal{I}}(\mathbf{x}) = (a', b')$ are:*

- $\mathbf{x} = (a - 2^{-p}, b' + 0^\star\beta^{-i+1}) \in R_{i-1,0^\star} \cap R_{i-1}^{unsat}$
- $\mathbf{x} = (a - 2^{-p}, \beta^i - b' + 0^\star\beta^{-i+1}) \in R_{i-1,2} \cap R_{i-1}^{unsat}$
- $\mathbf{x} = (a - 2^{-p} - A_i 2^{-q}, b' + 1^\star\beta^{-i+1}) \in R_{i-1,1^\star}^{lin} \cap R_{i-1}^{unsat}$ *(only if $a \geqslant 2^{-p} + A_i 2^{-q}$)*

The goal of those results in to show if there is a point in R_{fin} which is reachable from R_0 then we can extract, from its trajectory, a configuration which also reaches R_{fin}. Furthermore, we arranged so that R_{fin} contains the encoding of only one configuration:$(n+1, B)$ (see Lemma 12).

Lemma 23 (Backward-forward identity). *For any point $\mathbf{x} \in R_{fin}$, if there exists a point $\mathbf{y} \in R_0$ and an integer k such that $f_{\mathcal{I}}^{[k]}(\mathbf{y}) = \mathbf{x}$ then there exists a configuration $c = (1, 0, \varepsilon_1, \ldots, \varepsilon_n)$ such that $f_{\mathcal{I}}^{[k]}(\langle c \rangle) \in R_{fin}$.*

Lemma 24 (Final region is accepting). *For any configuration c, if $\langle c \rangle \in R_{fin}$ then $c = (n+1, B)$.*

3.3 Complexity Result

We now have all the tools to show that `REACH-REGION-TIME` is an NP-hard problem.

Theorem 25. *`REACH-REGION-TIME` is NP-hard for $d \geqslant 2$.*

Proof. Let $\mathcal{I} = (B, A_1, \ldots, A_n)$ be a instance of `SUBSET-SUM`. We consider the instance $\mathcal{J}$ of `REACH-REGION-TIME` defined in the previous section with maximum number of iterations set to n (the number of A_i), the initial region set to R_0 and the final region set to R_{fin}. One easily checks that this instance has polynomial size in the size of $\mathcal{I}$. The two directions of the proofs are:

- If $\mathcal{I}$ is satisfiable then use Lemma 11 and Lemma 16 to conclude that there is a point $x \in R_0$ in the initial region such that $f_{\mathcal{I}}^{[n]}(x) \in R_{fin}$ so $\mathcal{J}$ is satisfiable.
- If $\mathcal{J}$ is satisfiable then there exists $x \in R_0$ and $k \leqslant n$ such that $f_{\mathcal{I}}^{[k]}(x) \in R_{fin}$. Use Lemma 23 and Lemma 16 to conclude that there exists a configuration $c = (1, 0, \varepsilon_1, \ldots, \varepsilon_n)$ such that $\langle T_{\mathcal{I}}^{[k]}(c) \rangle = f_{\mathcal{I}}^{[k]}(\langle c \rangle) \in R_{fin}$. Apply Lemma 24 and use the injectivity of the encoding to conclude that $T_{\mathcal{I}}^{[k]}(c) = (n+1, B)$ and Lemma 12 to get that $\mathcal{I}$ is satisfiable.

4 Solving of Bounded Time Reachability

In the previous section we focused on what we can do with a reachability problem, and specifically how to solve a NP-hard problem with it. In this section, we take any such reachability problem and focus on how to actually solve it. More precisely we are interested in the complexity of solving the `REACH-REGION-TIME` problem.

4.1 Notations and Definitions

For any $i = 1, \dots, d$, let $\pi_i^d \colon I^d \to I$ denote the i^{th} projection function, that is, $\pi(x_1, \dots, x_d) = x_i$. Let $g_d \colon I^{d+1} \to I^d$ be defined by $g_d(x_1, \dots, x_{d+1}) = (x_1, \dots, x_d)$. For a square matrix A of size $(d+1) \times (d+1)$ define the following pair of projection functions. The first function $h_{1,d}$ takes as input a square matrix A of size $(d+1) \times (d+1)$ and returns a square matrix of size $d \times d$ which is the upper-left block of A. The second function $h_{2,d}$ takes as input a square matrix A of size $(d+1) \times (d+1)$ and returns the vector of size d given by $[a_{1,d+1} \cdots a_{d,d+1}]^T$ (the last column of A minus the last element).

Let s denote the size function, its domain of objects will be overloaded and understood from the context. For $x \in \mathbb{Z}$, $s(x)$ is the length of the encoding of x in base 2. For $x \in \mathbb{Q}$ with $x = \frac{p}{q}$ we have $s(x) = \max(s(p), s(q))$. For an affine function f we define the size of $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ (where all entries of A and $\mathbf{b}$ are rationals) as: $s(f) = \max(\max_{i,j}(s(a_{i,j})), \max(s(b_i)))$. We define the size of a polyhedron r defined by $A\mathbf{x} \leqslant \mathbf{b}$ as: $s(r) = \max(s(A), s(\mathbf{b}))$.

We define the size of a piecewise affine function f as: $s(f) = \max_i(s(f_i), s(r_i))$ where f_i denotes the restriction of f to r_i the i^{th} region.

We define the *signature* of a point $\mathbf{x}$ as the sequence of indices of the regions traversed by the iterates of f on $\mathbf{x}$ (that is, the region trajectory).

4.2 Results

In order to solve a reachability problem, we will formulate it with linear algebra. However a crucial issue here is that of the size of the numbers, especially when computing powers of matrices. Indeed, if taking the n^{th} power of A yields a representation of exponential size, no matter how fast our algorithm is, it will run on exponentially large instances and thus be slow.

First off, we show how to move to homogenous coordinates so that f becomes piecewise linear instead of piecewise affine.

Lemma 26. *Assume that $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ with $A = (a_{i,j})_{1 \leqslant i,j \leqslant d}$ and let $y = A'(\mathbf{x}, 1)^T$ where A' is the block matrix $\begin{pmatrix} A & \mathbf{b} \\ 0 & 1 \end{pmatrix}$. Then $f(x) = g_d(A'(\mathbf{x}, 1)^T)$.*

Remark 27. Notice that this lemma extends nicely to the composition of affine functions: if $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ and $h(\mathbf{x}) = C\mathbf{x} + \mathbf{d}$ then $h(f(x)) = g_d(C'A'(\mathbf{x}, 1)^T)$.

We can now state the main lemma, namely that the size of the iterates of f vary linearly in the number of iterates, assuming that f is piecewise affine.

Lemma 28. *Let $d \geqslant 2$ and $f \in PAF_d$. Assume that all the coefficients of f on all regions are rationals. Then for all $t \in \mathbb{N}$, $s(f^{[t]}) \leqslant (d+1)^2 s(f)pt + (t-1)\lceil \log_2(d+1) \rceil$ where p is the number of regions of f. This inequality holds even if all rationals are taken to have the same denominator.*

Finally, we need some result about the size of solutions to systems of linear inequalities. Indeed, if we are going to quantify over the existence of a solution of polynomial size, we must ensure that the size constraints does not change the satisfiability of the system.

Lemma 29 ([13]). *Let A be a $N \times d$ integer matrix and $\mathbf{b}$ an integer vector. If the $A\mathbf{x} \leqslant \mathbf{b}$ system admits a solution, then there exists a rational solution x_s such that $s(x_s) \leqslant (d+1)L + (2d+1)\log_2(2d+1)$ where $L = \max(s(A), s(b))$.*

Proof. See Theorem 5 of [13]: $s(x_s) \leqslant s\left((2d+1)!2^{L(2d+1)}\right)$.

Putting everything together, we obtain a fast nondeterministic algorithm to solve `REACH-REGION-TIME`. The nondeterministism allows use to choose a signature for the solution. Once the signature is fixed, we can write it as a linear program of reasonable size using Lemma 28 and solve it. The remaining issue is the one of the size of solution but fortunately Lemma 29 ensures us that there is a small solution which can be found quickly.

Theorem 30. *`REACH-REGION-TIME` is in NP.*

5 Other Results

In this section, we give succint proofs of the other result mentioned in the introduction about `CONTROL-REGION-TIME`. The proof is based on the same arguments as before.

Theorem 31. *Problem `CONTROL-REGION-TIME` is coNP-hard for $d \geqslant 2$.*

Proof. The proof is exactly the same except for two details:

- we modify f over R_{n+1} as follows: divide R_{n+1} in three regions: R_{low} which is below R_{fin}, R_{fin} and R_{high} which is above R_{fin}. Then build f such that $f(R_{low}) \subseteq R_{low}$, $f(R_{fin}) \subseteq R_{fin}$ and $f(R_{high}) \subseteq R_{low}$.
- we choose a new final region $R'_{fin} = R_{low}$.

Let $\mathcal{I} = (B, A_1, \ldots, A_n)$ be an instance of NOSUBSET-SUM, let $\mathcal{J}$ be the corresponding instance of CONTROL-REGION-TIME we just built. We have to show that $\mathcal{I}$ has no subset sum if and only if $\mathcal{J}$ is "controlled". This is the same as showing that $\mathcal{I}$ has a subset sum if and only if $\mathcal{J}$ has points never reaching R'_{fin}.

Now assume for a moment that the instance is in SUBSET-SUM (as opposed to NOSUBSET-SUM), then by the same reasoning as the previous proof, there will be a point which reaches the old R_{fin} region (which is disjoint from R'_{fin}). And since R_{fin} is a f-stable region, this point will never reach R'_{fin}.

And conversely, if the control problem is not satisfied, necessarily there is a point which trajectory went through the old R_{fin} (otherwise if would have reached either $R_{low} = R'_{fin}$ or R_{high} but $f(R_{high}) \subseteq R_{low}$). Now we proceed as in the proof of Theorem 25 to conclude that there is a subset which sums to B, and thus $\mathcal{I}$ is satisfiable.

Theorem 32. *Problem* `CONTROL-REGION-TIME` *is in* $coNP$.

Proof. Again the proof is very similar to that of Theorem 30: we have to build a non-deterministic machine which accepts the "no" instances. The algorithm is exactly the same except that we only choose signatures which avoid the final region (as opposed to end by the final region) and are of maximum length (that is $t = T$ as opposed to $t \leqslant T$). Indeed, if there is a such a trajectory, the problem is not satisfied. And for the same reasons as Theorem 30, it runs in non-deterministic polynomial time.

References

1. Koiran, P., Cosnard, M., Garzon, M.: Computability with low-dimensional dynamical systems. Theoretical Computer Science 132, 113–128 (1994)
2. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? Journal of Computer and System Sciences 57, 94–124 (1998)
3. Moore, C.: Generalized shifts: unpredictability and undecidability in dynamical systems. Nonlinearity 4, 199–230 (1991)
4. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. Journal of Computer and System Sciences 50, 132–150 (1995)
5. Asarin, E., Maler, O., Pnueli, A.: Reachability analysis of dynamical systems having piecewise-constant derivatives. Theoretical Computer Science 138, 35–65 (1995)
6. Koiran, P., Cosnard, M., Garzon, M.: Computability with Low-Dimensional Dynamical Systems. Theoretical Computer Science 132, 113–128 (1994)
7. Asarin, E., Schneider, G.: Widening the boundary between decidable and undecidable hybrid systems. In: Brim, L., Jančar, P., Křetínský, M., Kučera, A. (eds.) CONCUR 2002. LNCS, vol. 2421, pp. 193–208. Springer, Heidelberg (2002)
8. Asarin, E., Schneider, G., Yovine, S.: On the decidability of the reachability problem for planar differential inclusions. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 89–104. Springer, Heidelberg (2001)
9. Bell, P., Chen, S.: Reachability problems for hierarchical piecewise constant derivative systems. In: Abdulla, P.A., Potapov, I. (eds.) RP 2013. LNCS, vol. 8169, pp. 46–58. Springer, Heidelberg (2013)
10. Ben-Amram, A.M.: Mortality of iterated piecewise affine functions over the integers: Decidability and complexity. In: STACS, pp. 514–525 (2013)
11. Blondel, V.D., Bournez, O., Koiran, P., Tsitsiklis, J.: The stability of saturated linear dynamical systems is undecidable. Journal of Computer and System Science 62, 442–462 (2001)
12. Garey, M.R., Johnson, D.S.: Computers and Intractability. W. H. Freeman and Co. (1979)
13. Koiran, P.: Computing over the reals with addition and order. Theor. Comput. Sci. 133, 35–47 (1994)