Alcom-FT Technical Report Series
ALCOMFT-TR-01-191

# A step towards a complexity theory for dynamical systems

Marco Gori[1]* and Klaus Meer[2]**

[1] Dipartimento di Ingegneria dell'Informazione
Università di Siena
Via Roma 56, I-53100 Siena, Italy
[2] Syddansk Universitet Odense
Dept. of Mathematics and Computer Science
Campusvej 55, DK-5230 Odense M, Denmark
meer@imada.sdu.dk

**Abstract.** Recent years have seen an increasing interest in the study of continuous-time computational models. However, not so much has been done with respect to setting up a complexity theoretic framework for such models. The present paper intends to go a step into this direction. We consider problems over the real numbers which we try to relate to Lyapunov theory for dynamical systems: The global minimizers of particular energy functions are supposed to give solutions of the problem. The structure of such energy functions leads to the introduction of problem classes **U** and **NU**; for the systems we are considering they parallelize the classical complexity classes **P** and **NP**. We then introduce a notion of reducibility among problems and show existence of complete problems for **NU** and for **PU**, a polynomial hierarchy of continuous-time problems.
For previous work on the computational capabilities of continuous-time systems see the surveys by Cris Moore [9] and by Pekka Orponen [10]. Our paper presents a step into the direction of creating a general framework for a complexity theory of continuous-time systems as outlined in [10]. It is closely related to work done by Hava.T. Siegelmann, A. Ben-Hur and Shmuel Fishman [12, 11].

**Key words:** Dynamical systems, analog computations, complexity, optimization

## 1 Introduction

Continuous problem solving is a fundamental issue at the cross-road of many disciplines. The idea of conceiving the process of computation as the evolution of the state of a dynamical system has been explored by many researchers using different approaches, see the very interesting survey by Pekka Orponen [10] and the literature cited in there. However, Orponen writes in the conclusion of his survey about the state of the art dealing with analog systems from a complexity point of view: 'The most significant unexplored area is surely the computational complexity-theory of continuous-time systems'. And a bit further on: 'In

* Partially supported by PAR (Piano Ateneo Ricerca)- University of Siena
** Klaus Meer was partially supported by EU ESPRIT project NeuroCOLT2 and by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

parallel, one should develop the appropriate notions of complexity classes, reductions, and hard problems for continuous-time computation'.

The present paper is devoted to do a step into the latter direction. Our goal is to introduce two classes **U** and **NU** of such continuous-time problems which, to a certain extent, parallelize the well-known discrete complexity classes **P** and **NP**. In our approach, energy functions can be seen as algorithms and energy functions with suitable properties are the analogue of efficient algorithms. These properties are related to the structure of local minima of the energies. The philosophy behind is that problems whose solutions are provided by global minima of energies are easier to solve if the structure of local minima is not too complicated than if it is arbitrarily complex. Note, however, that even the minimization of a simply structured energy can create problems, as was shown in [13].

We then define a reducibility notion; as main results we obtain the existence of complete problems for our class **NU** and for a generalization of the latter to a polynomial hierarchy **PU** of continuous-time problems.

Previous approaches related to ours are the ones by Siegelmann and Fishman [12] and by Siegelmann, Ben-Hur, and Fishman [11]. They are mainly interested in defining the notion of time-complexity for computations related to analog systems

$$\frac{dx}{dt} = F(x(t)),$$

where $t \in \mathbb{R}, x(t)$ is a $n$-dimensional vector and $F$ is a $n$-dimensional vector field of $x$. In [12], especially systems with Lyapunov energy functionals are investigated. This can be seen as a special case of our framework. We define a problem $\Pi$ as a family of subsets in different spaces $\mathbb{R}^n \times \mathbb{R}^k$; for an input $d \in \mathbb{R}^n$ we want to find a $y \in \mathbb{R}^k$ such that $(x, y) \in \Pi$. For example, $\Pi$ could be the problem to find an equilibrium point of a given dynamical system. Solutions of $\Pi$ are supposed to be related to global minimizers of an energy function.

Complexity issues related to our approach are twofold. On the one hand side, the complexity of a problem is influenced by the structure of the related energies. This leads to the introduction of our complexity classes $\mathbf{U}, \mathbf{NU}$ and **PU**, to completeness notions and will be the main issue in this paper. The second contribution to the complexity reflects the work that has to be done when actually following a trajectory by using optimization methods. This part, which is beyond the scope of this extended abstract, is guided by common phenomena like conditioning etc. Our point of view is that a reasonable foundation of a complexity theory for continuous-time systems has to incorporate both complexity parts.

In order to obtain a significant notion of hardness, several aspects are taken into account: The underlying idea for defining the class **U** is the requirement that optimal points of unimodal energy functions are related to our system. This can be seen as an analogue of **P** since unimodal energies provide an easy structure of locally minimal values. A class **NU** similar to **NP** is defined by means of systems with (potentially) non-unimodal energies.

In addition, we pose conditions on the computability of the energies. Since we work in a real number framework it is natural to consider a real number computational model for evaluating energies; we shall use Straight-Line programs SLP and the Blum-Shub-Smale (shortly: BSS) machine. We suppose the reader to be familiar with these models, see [3, 1].

The use of algebraic SLPs implies that energies can be computed by operations from the set $\mathcal{O} := \{+, -, \times\}$. However, our reasoning does not crucially depend on this particular set. Other functions like exponentials, trigonometric functions etc. might be included. This could be interesting in order to combine our approach to analog devices which use more general functions among their operations, for example the ones studied in [4].

The present results should be considered as a first step towards introducing and studying complexity classes and notions like completeness for problems arising in relation with analog computations. In the concluding section we shall outline further directions along the approach developed here.

## 2 Definition of a problem

Let us begin by defining the notion of a problem. Let $\mathbb{R}^\infty := \bigoplus_{i\geq 1} \mathbb{R}^i$ denote the set of finite sequences of real numbers ($\mathbb{R}^\infty$ is a real counterpart of $\Sigma^*$ for finite alphabets $\Sigma$).

**Definition 1.** *A* PROBLEM $\Pi$ *is a relation in* $\mathbb{R}^\infty \times \mathbb{R}^\infty$. SOLVING A PROBLEM *means that on input* $d \in \mathbb{R}^n$ *for some* $n \in \mathbb{N}$, *a vector* $y \in \mathbb{R}^k$ *for some* $k \in \mathbb{N}$ *is computed such that* $(d, y) \in \Pi$. *Here, we require the output dimension* $k$ *to be polynomially related to* $n$, *i.e. there exists a polynomial* $p$ *such that* $k = p(n)$ *for all* $n \in \mathbb{N}$.

*Notation: We will frequently write* $\Pi(d)$ *to denote a solution* $y$ *such that* $(d, y) \in \Pi$, *even though* $\Pi$ *might not be a function.*

*Example 1.* As already mentioned, a typical example of a problem in our setting is a dynamical system $\frac{dx}{dt} = F(x(t))$ for which an equilibrium point is searched. This can be formalized in different ways. One possibility is to consider $F$ as the first component of a problem $\Pi$ and to look for a $y$ which is an equilibrium point of $F$. Thus

$$\Pi = \{(F, y) \in \mathbb{R}^\infty \times \mathbb{R}^\infty \mid y \text{ is an equilibrium of } \frac{dx}{dt} = F(x(t))\} \ .$$

The above definition requires $F$ to be representable in a certain way as a point in $\mathbb{R}^\infty$. This is, for example, the case if $F$ is given as a rational function. For dealing with vector fields $F$ involving more complicated functions than just those composed by arithmetic operations one has to consider an enlarged operation set $\mathcal{O}$.

We are interested in studying problems closely related to specific energy functions. The latter are defined for every dimension of an input for the problem. Their global minimizers are supposed to yield (by a simple additional computation) a solution of the corresponding problem for the input under consideration.

The energy functions are basically given by an infinite family $\{E_n\}_{n\in\mathbb{N}}$. Every $E_n$ is a function depending on two blocks $d$ and $w$ of variables. The block $d \in \mathbb{R}^n$ later on is taken to be the input of a problem $\Pi$. The block $w \in \mathbb{R}^m$ will be related to a solution $y$ of $\Pi$ for input $d$. Once more, the dimension $m$ should be polynomially related to $n$, that is $m = q(n)$ for a polynomial $q$. This polynomial is given together with the family $\{E_n\}$.

Another main point of the definition is the way how these energies can be computed.

**Definition 2.** *(Straight-line programs) For an operation set* $\mathcal{O}$ *a* STRAIGHT-LINE PROGRAM *of input dimension* $T$ *over* $\mathcal{O}$ *is a sequence* $\beta_1, \ldots, \beta_\ell$ *of operations defined as follows. Every* $\beta_i$ *is either of the form* $\beta_i := c$ *for a constant* $c \in \mathbb{R}$ *or of the form* $\beta_i := \beta_j \circ \beta_k$, *where* $\circ \in \mathcal{O}$ *and* $j, k \in \{-T+1, \ldots, 0, 1, \ldots i-1\}$. *For any* $T$*-dimensional real input* $x_1, \ldots, x_T$, *to the first* $T$ *registers* $\beta_{-T+1}, \ldots, \beta_0$ *we assign the values* $\beta_{-T+1} := x_1, \beta_{-T+2} := x_2, \ldots, \beta_0 := x_T$. *A computation of the program then proceeds in the obvious manner assigning the corresponding values to the* $\beta_i, i \geq 1$. *The result is supposed to be computed in* $\beta_\ell$. *The* SIZE *or* COMPUTATION TIME *of the program is the number* $\ell$ *of operations performed.*

In our framework, the input variables are chosen as $d_1, \ldots, d_n$ and $w_1, \ldots, w_{q(n)}$. Thus, $T = n + q(n)$. We are interested in computing a real valued function $E_n : \mathbb{R}^{n+q(n)} \to \mathbb{R}$.

*Remark 1.* Instead of computing just a single real number, SLPs can also be used to compute vector functions. In that case some registers $\beta_{i_k}, 1 \leq k \leq t, i_k \in \{-T+1, \ldots, \ell\}$, have to be predetermined. The $t$-dimensional result of such a computation then can be found in these predetermined registers after performing the computation. We shall make use of such more general SLPs later on in the definition of SLP reductions.

We want to combine SLPs in a uniform way in order to relate them to functions from $\mathbb{R}^\infty \to \mathbb{R}^\infty$.

**Definition 3.** *A family $\mathcal{E} := \{E_n\}_{n\in\mathbb{N}}$ of SLPs, every $E_n$ of input dimension $n+q(n) \in \mathbb{N}$ for a fixed polynomial $q$, is called to be* UNIFORMLY POLYNOMIALLY BOUNDED *if there exists a BSS machine $M$ which, on input $n \in \mathbb{N}$, computes a description (i.e. a point in $\mathbb{R}^\infty$) of $E_n$ and runs in polynomial time with respect to $n$. We call such a family an* SLP ENERGY FAMILY.

Clearly, every member $E_n$ of an SLP energy family has a polynomial size in $n$. The result of $E_n$ on an input $(d, w)$ of dimension $n + q(n)$ can be computed in polynomial time in the BSS model over $\mathcal{O}$. Note that if the allowed operations are $\mathcal{O} := \{+, -, \times\}$ all functions $E_n$ are $C^\infty$ (because they are polynomials).

## 3 The classes U and NU

In relation with energy functions we shall now define two complexity classes **U** and **NU** of problems. They can be seen as a continuous counterpart of the classes **P** and **NP** in classical complexity theory. The idea is to consider problems $\Pi$ whose solutions can be obtained as a global minimizer of a related energy function.

In case of class **U** we assume the corresponding energy functions to be unimodal for every fixed input. That is, there is only one unique locally minimal function value which at the same time is a globally minimal value (i.e. every local minimizer is a global one). In case of **NU**, energies can be arbitrary, i.e. they can have several local and global minimizers.

**Definition 4.** *Let $\Pi$ be a problem.*

*a) $\Pi$ belongs to the class* **U** *if there exists an SLP energy family $\{E_n\}_n$ together with another family $\{N_n\}$ of SLPs which is uniformly given by a polynomial time BSS machine such that the following is true:*
  - *i) There is a fixed polynomial $q$ such that every $E_n$ is a map $E_n : \mathbb{R}^n \times \mathbb{R}^{q(n)} \to \mathbb{R}$;*
  - *ii) For any fixed $d \in \mathbb{R}^n$ the function $w \to E_n(d, w)$ is unimodal;*
  - *iii) If $w^*$ is a global minimizer of $w \to E_n(d, w)$ for given $d$, then we can compute a solution $\Pi(d)$ using the SLP $N_{q(n)}$. More precisely, $(d, N_{q(n)}(w^*)) \in \Pi$.*

*b) Problem $\Pi$ belongs to class* **NU** *if items i) and iii) above hold, but $w \to E_n(d, w)$ has not to be unimodal. Clearly, it is* $\mathbf{U} \subseteq \mathbf{NU}$.

*Note that the family $\{N_n\}$ in a) satisfies the definition of an energy family as well. However, it is used in a different way because we are not interested in finding a minimizer of the functions computed by the $N_n$.*

The idea behind this definition is as follows. If a solution $(d, y)$ of a problem instance $d$ is - up to an easy computation represented by the function $N_{q(n)}$ - given by a global minimizer of an energy $w \to E_n(d, w)$, then it is reasonable to focus on the structure of the set of local minimizers and on methods to approach a global one. Therefore, we also require the energies to be efficiently computable; a single function evaluation of $E_n$ should not dominate the allover efforts needed to solve the problem.

One might also think about using the BSS model instead of SLPs in the above definition of energies. However, this would introduce new aspects; on the one hand side, the principle optimization idea in that case would be changed to *constrained* optimization problems. On the other hand, the feature of how to handle equality tests is avoided when using SLPs. Of course, both aspects do not prvent from the study of such more general energy functions.

It is well known from optimization theory that minimization becomes the harder the more complex the structure of locally minimal values is. For unimodal energies, any optimization process which eventually reaches a local minimum is fine. For energies of problems in **NU**, however, methods can get stuck in local minima. Here, one has to rely on guessing good starting points close enough to a global minimizer. This situation parallelizes the nondeterministic guess of a solution for an instance of an **NP**-complete problem in classical complexity theory.

Finding good energies can be regarded as counterpart to finding polynomial time algorithms. The above arguments lead to the conjecture that **U** is a proper subclass of **NU**.

## 4 A few words about solving problems

As mentioned in the introduction the general program we want to follow has two equally important parts. The first is devoted to define problems in the framework of analog computation. The second part then deals with solution algorithms. In this paper, we focus on the first step and try to lay a foundation of a complexity theory for such problems. The underlying philosophy here is that the description of the energies, which represent a problem, is much easier then the solution of a problem. This is reflected in the above definitions by the use of SLPs working in polynomial time. The actual difficulty of solving problems results both from the structure of local minima of the energies and the minimization process.

Concerning the latter part, at least we want to outline here the further direction of investigations. In our computational model, the deterministic terminal attractor machine, we consider a basic part of an algorithm to follow a trajectory of a differential equation into an equilibrium point of the system. Then, depending on whether this is already a global minimum or not, the computation has to be continued.

A way to deal with solution algorithms is to discretize the dynamical system and analyze the behavior of classical algorithms like steepest descent for the energies. Given a problem and a related energy family $\{E_n\}$, for an input $d \in \mathbb{R}^n$ we can consider the differential equation (terminal attractor machine)

$$\frac{dw(t)}{dt} = -\frac{E_n(d,0)}{\sigma} \cdot \frac{D_w E_n(d, w(t))}{\|D_w E_n(d, w(t))\|^2} \ , \ \sigma > 0 \text{ a constant.}$$

The evolution of $E_n(d, w(t))$ can be seen from

$$\frac{E_n(d, w(t))}{dt} = (D_w E_n(d, w(t)))' \cdot \frac{dw(t)}{dt}$$

which, in turn, yields

$$\frac{E_n(d,w(t))}{dt} = -\frac{E_n(d,0)}{\sigma}.$$

Finally, $E_n(d,w(t)) = E_n(d,0)(1-\frac{t}{\sigma})$, which reaches the terminal value $w_t = \sigma$. Hence, the terminal attractor machine based on the above differential equation returns an equilibrium point in constant time $\sigma$. The structure of the related energy then determines the further ongoing. For example, the equilibrium could already be a global minimum, or at least a descending direction could easily be found, or one could get stuck in a local minimum.

Of course, after studying the structure of energies a precise analysis of the complexity of such a process has to be performed as well. This is actually initiated in [12] for some special cases and certainly represents one of the main topics for future research in this area. Such investigations will involve both approximation issues and the conditioning of problems. Most likely, it will then be reasonable to introduce further subclasses of **U** and **NU** which reflect the dependence of solution algorithms not only on the structure of local minima, but also on the intended accuracy and the conditioning of the problem.

It is important to recall that the major effort of solving problems is behind the optimization process and the structure of minima. Therefore, it makes sense to investigate a completeness notion for **NU**, even though we do not precisely define here the complexity of following trajectories (but one can take [12] as a way to attack this issue).

## 5 Reducibility notions and completeness results

We turn to the introduction of a reducibility notion among problems in **NU**. It is inspired by the above remarks on the core difficulty of solving a problem.

Following the traditional setting there is a straightforward notion of reducibility using the definition of class **U**, see Definition 6 below. However, we shall first introduce another notion and then show that it is stronger than the straightforward one. Then, we show the existence of complete problems under that stronger reducibility notion.

**Definition 5.** *a) Let $\Pi_1$ and $\Pi_2$ be two problems. We say that $\Pi_1$ is* SLP-REDUCIBLE IN POLYNOMIAL TIME *to $\Pi_2$ if there exist two functions $\phi$ and $\phi^*$ from $\mathbb{R}^\infty \to \mathbb{R}^\infty$, both computable in polynomial time by a uniform SLP in the BSS model of computation, such that*

$$\forall d \in \mathbb{R}^\infty \quad \Pi_1(d) = \phi^*\left(\Pi_2\left(\phi(d)\right)\right)$$

*b) A problem $\Pi \in$ **NU** is* **NU**-COMPLETE *if every other problem in **NU** is SLP-reducible in polynomial time to $\Pi$.*

The following Lemma is almost obvious taking into account the related properties of reductions in the BSS model.

**Lemma 1.** *Let $\Pi_1, \Pi_2$ be problems.*

*i) SLP-reducibility in polynomial time is a transitive relation among problems.*
*ii) If $\Pi_2 \in$ **U** and $\Pi_1$ is SLP-reducible in polynomial time to $\Pi_2$, then $\Pi_1 \in$ **U**.*
*iii) If $\Pi_1$ is **NU**-complete and SLP-reducible in polynomial time to $\Pi_2 \in$ **NU**, then $\Pi_2$ is also **NU**-complete.*

*iv)* $\mathbf{U} = \mathbf{NU} \;\Leftrightarrow\;$ *there is a* $\mathbf{NU}$*-complete problem in* $\mathbf{U}$.

*Proof.* ad i): Follows just by writing down the definition.

ad ii): Let $\Pi_2$ be in $\mathbf{U}$ and suppose $\{E_n\}$ and $\{N_n\}$ to be families of SLPs proving this fact. Let $\phi$ and $\phi^*$ be SLPs which witness the reducibility of $\Pi_1$ to $\Pi_2$. Now, let $d$ be an input for $\Pi_1$. We have $\Pi_1(d) = \phi^*(\Pi_2(\phi(d)))$. If $\tilde{d} := \phi(d)$ is of dimension $s$ (polynomially bounded in $n$), then $\Pi_2(\tilde{d}) = N_s(y)$ for a global minimizer $y$ of $E_s(\tilde{d}, w)$. We have to define a unimodal SLP energy family $\{E_n^*\}$ together with a polynomially computable uniform family $\{N_n^*\}$ of SLPs such that these families prove membership of $\Pi_1$ in $\mathbf{U}$. The SLPs $N_m^*$ are defined as the composition of the SLPs $\phi^* \circ N_m$. The energy functions $E_s^*$ are given as SLPs for the functions $w \to E_n(\phi(d), w)$. Note that because $\phi$ is supposed to be an SLP, $E_s^*$ can be constructed as an SLP as well; the precomputation of $\phi(d)$ starting with $d$ can be done uniformly by an SLP for every dimension $n$.

It is now easy to see that these families of SLPs witness membership of $\Pi_1$ in $\mathbf{U}$.

ad iii): The construction of item ii) works as well if the energies are not unimodal.

ad iv): Follows from ii) and the reducibility definition. □

As already mentioned, in analogy to the classical $\mathbf{P}$-reduction used to define $\mathbf{NP}$-completeness one might study the following notion of reducibility among problems in $\mathbf{NU}$ as well:

**Definition 6.** *Let* $\Pi_1, \Pi_2$ *be two problems in* $\mathbf{NU}$. $\Pi_1$ *is* $\mathbf{U}$-REDUCIBLE TO $\Pi_2$ IN POLYNOMIAL TIME *if there exist functions* $\Phi_U, \Phi_U^*$ *satisfying condition a) in Definition 5 and belonging to class* $\mathbf{U}$.

Thus, when considered themselves as problems by taking the graph relation, the reduction functions are related to unimodal energies whose minima give the function values. It turns out that SLP-reducibility is a stronger notion:

**Theorem 1.** *SLP-reducibility among problems in* $\mathbf{NU}$ *implies* $\mathbf{U}$*-reducibility.*

*Proof.* Let $\Pi_1$ be SLP-reducible to $\Pi_2$ in polynomial time. There exist functions $\phi, \phi^*$ which are computable in polynomial time by uniform families of SLPs and which satisfy the SLP-reducibility conditions of Definition 5. We shall show that both $\phi$ and $\phi^*$ can be taken to realize the $\mathbf{U}$-reduction of $\Pi_1$ to $\Pi_2$. Towards this end, it remains to show $\phi, \phi^* \in \mathbf{U}$. Let $M$ be a BSS machine over $\mathcal{O}$ which generates the SLPs related to $\phi$, say $\{P_n\}_n$. Let $d \in \mathbb{R}^n$ be an input for $\Pi_1$. Using the well known representation of the computation of an SLP, see [3], we formalize the calculation of $P_n(d)$ : Suppose the sequence of computational steps to be of the form

$$\begin{aligned}
\beta_1 &= \beta_{j_1} \circ_1 \beta_{k_1} \quad ; \; j_1, k_1 \in \{-n-m+1, \ldots, 0\}, \circ_1 \in \mathcal{O} \\
\beta_2 &= \beta_{j_2} \circ_2 \beta_{k_2} \quad ; \; j_2, k_2 \in \{-n-m+1, \ldots, 1\}, \circ_2 \in \mathcal{O} \\
&\;\vdots \\
\beta_\ell &= \beta_{j_\ell} \circ_\ell \beta_{k_\ell} \quad ; \; j_\ell, k_\ell \in \{-n-m+1, \ldots, \ell-1\}, \circ_\ell \in \mathcal{O} \; .
\end{aligned}$$

The value computed by the SLP is $\beta_\ell = P_n(d) = \phi(d)$.

We can assign a polynomial $f_d$ in unknowns $\beta_i, 1 \le i \le \ell$ to the computation of the SLP by replacing every computational step $\beta_i = \beta_{j_i} \circ_i \beta_{k_i}$ by the term $(\beta_i - \beta_{j_i} \circ_i \beta_{k_i})^2$ and finally summing up the $\ell$ many terms. If division is included as well, we replace $\beta_i = \frac{\beta_{j_i}}{\beta_{k_i}}$ by $(\beta_i \cdot \beta_{k_i} - \beta_{j_i})^2$. The following properties of $f_d$ are important:

i) $f_d(\beta) \geq 0 \; \forall \beta$

ii) $\beta^*$ is a local minimizer of $\beta \to f_d(\beta)$ if and only if $\beta^*_\ell = \phi(d)$. To prove this claim consider the partial derivatives $\frac{\partial f_d}{\partial \beta_i}, 1 \leq i \leq \ell$ of $f_d$. The first order necessary condition $D_\beta f_d = 0$ for a minimizer is satisfied by a $\beta$ iff $d$ and $\beta$ are coupled through the computation of the SLP $P_n$ on input $d$; setting the partial derivatives equal to zero precisely reflects this computation.

iii) $f_d$ is a polynomial of degree 4 which is computable in polynomial time (in the BSS model) in the size of input $d$.

Since $\phi$ is uniformly computable for all $d \in \mathbb{R}^\infty$, the above reduction gives a uniformly computable family of energies $E_n(d, \beta) := f_d(\beta)$ such that, for fixed $d \in \mathbb{R}^n$, the function $E_n(d, \beta)$ is unimodal (see ii) above). Moreover, if $\beta^*$ is a minimizer of $\beta \to E_n(d, \beta)$ the projection $N_n(d, \beta^*) := \beta^*_\ell$ gives $\phi(d)$. We can therefore choose $\phi$ itself as $\Phi_U$ and obtain $\Phi_U \in \mathbf{U}$. The argument for $\Phi^*_U$ is the same. □

Now, we shall show that **NU** has complete problems. The complete problem we establish is a general one in the sense that the proof works for almost arbitrary sets of operations. It is therefore of a theoretical flavor.

At the end of the section the existence of more natural complete problems will be discussed.

For sake of simplicity, the operation set again will be chosen as $\mathcal{O} := \{+, -, \times\}$.

**Theorem 2.** *There exist* **NU***-complete problems w.r.t. the set $\mathcal{O}$ chosen above.*

*Proof.* The construction works along the lines for deriving complete problems for general discrete complexity classes by defining a problem related to the calculations of a universal machine. Let $f : \mathbb{R}^\infty \to \mathbb{R}^\infty$ be a polynomial time computable function in the BSS model. The corresponding BSS machine can be coded as a vector in some $\mathbb{R}^k$. Thus, we can identify a function like $f$ with its code-vector in some $\mathbb{R}^k$.

For the **NU**-complete problem we are going to define, we restrict ourselves to functions $f$ which represent a uniform description of an SLP energy family. Therefore, the restrictions of $f$ to fixed input dimensions $s$ are supposed to be computable by an SLP. They are functions with two variable blocks $(d, w) \in \mathbb{R}^{s_1} \times \mathbb{R}^{s_2}, s_1 + s_2 := s$.

We define a problem $\Pi$ as follows: $\Pi$ is a relation in $(\mathbb{R}^\infty \times \mathbb{R}^\infty) \times \mathbb{R}^\infty$. An input $(f, d)$ to $\Pi$ is interpreted as the code of a function $f$ giving an SLP energy family, together with an input $d \in R^{s_1}$ for the second block of arguments of $f$ (in the above sense). A solution for $\Pi$ is a vector $y \in \mathbb{R}^{s_2}$ such that $y$ is a global minimizer of the function $w \to f(d, w)$.

We first have to show membership of $\Pi$ in **NU**. Towards this end, we define an SLP energy family $\{E_n\}_n$ and a polynomial time computable function $G$ such that for every input $(f, d)$ a global minimum $y$ of the function $w \to E_n(f, d, w)$ satisfies $(f, d, G(y)) \in \Pi$. Moreover, $G$ has to represent uniformly a SLP family. The definition of the components $E_n$ of the energy family $\mathcal{E}$ is given by $E_n(f, d, w) := f(d, w)$ and $G$ can be chosen on all dimensions as the identity. Note that a description of $f$ is given by a BSS program working in polynomial time in $n$. Therefore, the family $\{E_n\}$ satisfies the dimension condition of Definition 4, part a),i). Moreover, it is an SLP energy family: Given $n \in \mathbb{N}$, a (uniform) description of the SLP $E_n$ can be obtained using the uniform program computing $f$.

Now, let $\tilde{\Pi}$ be a problem in **NU**. Let $\tilde{f} : \mathbb{R}^\infty \to \mathbb{R}^\infty$ be the corresponding function giving the SLP energy family for $\tilde{\Pi}$. We again identify $\tilde{f}$ with its code in $\mathbb{R}^\infty$. For input $d$ for $\tilde{\Pi}$ we define $\phi(d) := (\tilde{f}, d)$ and take $(\tilde{f}, d)$ as input for problem $\Pi$. Since the code $\tilde{f}$ is constant with respect to problem $\tilde{\Pi}$, the function $\phi$ can be realized on all dimensions via an SLP and is polynomial time computable. Next, let $y$ be a solution of $\Pi$ on input $(\tilde{f}, d)$, i.e. $(\tilde{f}, d, G(y)) \in \Pi$. By definition of $\Pi$ the vector $y = G(y)$ is a solution of $\Pi_1$ on input $d$. Thus, taking $\phi^*$ to be the identity finishes the proof. $\square$

Looking for more natural problems, one can prove the following one to capture the difficulty of class **NU**.

**Definition 7.** *Let* **QP** *denote the following quadratic optimization problem: Given a linear objective function $f$ together with finitely many constraints $h_1(x) = 0, \ldots, h_m(x) = 0$, where the $h_i$ are polynomials of degree at most 2, find a solution point of* $\min\{f(x) | h_i(x) = 0, 1 \leq i \leq m\}$. *Thus,* **QP** *is the problem defined by tuples $(f, h_1, \ldots, h_m, x)$ such that $x$ is a global minimizer of the constrained optimization problem* $\min f(x)$ *subject to* $h_i(x) = 0$.

**Theorem 3.** *Let the set $\mathcal{O}$ be chosen as before. Then* **QP** *is* **NU***-hard with respect to $\mathcal{O}$.*

*Proof.* The proof is reminiscent to the $\mathbf{NP}_{\mathbb{R}}$-completeness proof for the so called 4-Feasibility problem in [2]. Consider a problem $\Pi \in \mathbf{NU}$. Let $M$ be the BSS machine over $\mathcal{O}$ generating the SLP energy family $\{E_n\}$. Let $d$ be an input in $\mathbb{R}^n$ for $\Pi$. We have to construct a function $\phi$ computable in polynomial time by a uniform family of SLPs; on input $d$ the result $\phi(d)$ has to represent a linear objective function together with a system of constraints. The latter have to be finitely many quadratic polynomial equations in variables $w$ such that from a minimum $w^*$ of the linear objective function under the constraints we can compute in polynomial time a vector $\phi^*(w^*)$ satisfying $(d, \phi^*(w^*)) \in \Pi$. The function $\phi^*$ has to be computable in polynomial time by a uniform family of SLPs as well.

First, consider the following decision problem related to the energies $E_n$: Given an input $d \in \mathbb{R}^n$ and a real $\beta$, does there exist an input $w$ such that $E_n(d, w) = \beta$? Since the $E_n$ are given by a polynomial time computable BSS program, this problem belongs to class $\mathbf{NP}_{\mathbb{R}}$ over the reals. We use it to construct the side constraints of the optimization problem $\phi(d)$. Towards this end we apply the $\mathbf{NP}_{\mathbb{R}}$-completeness theorem in [2]; there, it is shown that the problem of deciding whether a system of polynomials of degree 2 has a solution is $\mathbf{NP}_{\mathbb{R}}$ complete over the reals. The linear objective function of the instance of **QP** then is taken to be $\beta$. By using the proof of Theorem 1 it is not hard to see that the entire instance can be computed by an SLP. Since a similar argument occurs below in the proof of Theorem 4 we do not elaborate it here.

The final step of the proof is to determine $\phi^*$. The above reasoning shows that $\phi^*$ can be chosen once again as the identity. $\square$

The problem **QP** would be complete as well if we could establish its membership in **NU**. It is not obvious how to prove that since **QP** is a constrained optimization problem. An answer seems to be closely related to the study of external penalty functions in optimization theory, see [7].

## 6 A polynomial hierarchy

If we generalize the optimization problems associated to the energy functions we obtain as well an analogue of the polynomial hierarchy for problems in our framework. The (probably) different levels of that hierarchy arise by splitting the variables of the energy into more and more blocks and alternating the min and max operator on these blocks. For example, problems in the next level above **NU** are defined as relations $\Pi$ such that for input $d$ a vector $y$ satisfying $(d, y) \in \Pi$ can be obtained by computing a solution for $\max_w \min_v E_n(d, w, v)$.

The set $\mathcal{O}$ is once more chosen as $\{+, -, \times\}$.

**Definition 8.** *(Polynomial hierarchy) A problem $\Pi$ belongs to the continuous-time polynomial hierarchy* **PU** *if the following holds: there exist an SLP energy family $\{E_n\}_n$ and a function $N : \mathbb{R}^\infty \to \mathbb{R}^\infty$ computable by a uniform family of SLPs in polynomial time such that:*

*i) There is a fixed polynomial $q$ such that every $E_n$ is a map $E_n : \mathbb{R}^n \times \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \ldots \times \mathbb{R}^{n_k} \to \mathbb{R}$, where $\sum_{i=1}^{k} n_i = q(n)$.*

*ii) If $w_1^*$ is a solution for the choice of variables $w_1$ in the optimization problem*

$$\min_{w_1} \max_{w_2} \ldots \min_{w_k} \; E_n(d, w_1, \ldots, w_k)$$

*for given $d$, then we can compute a solution of $\Pi$ on input $d$ as $(d, N(w_1^*))$.*
*Here, of course the last optimization operation is* min *if $k$ is odd and* max *if $k$ is even. The problem belongs as well to* **PU** *if the optimization starts with* max.

In [2] it was shown that deciding the existence of a zero for a degree 4 polynomial is **NP**-complete over the reals. The following proof, which shows the existence of a complete problem for **PU**, focuses on reductions computable with SLPs instead of general BSS reductions.

**Theorem 4.** *The following problem is* **PU***-complete under SLP reductions: Given a polynomial $f$ of degree 4 in $n$ blocks $w, X_2, \ldots, X_n$ of variables, compute a minimizer of the function*

$$w \to \max_{X_2} \min_{X_3} \ldots \max_{X_n} f(w, X_1, \ldots, X_n)$$

*Again, the last optimization operation is* min *if $n$ is odd and* max *if $n$ is even.*

*Proof.* The proof is similar to the one of the corresponding statements about the classical polynomial hierarchy. We will outline the idea by dealing with a problem from **NU** and by showing, how we can reduce it to a problem on the next higher level of our hierarchy, i.e. to a $\min - \max$ problem for polynomials of degree 4.

Let a problem $\Pi \in$ **NU** be given. The related energies are denoted by $\{E_n\}$. For an input $d \in \mathbb{R}^n$ we are interested in computing a minimizer for $w \to E_n(d, w)$. Consider the decision problem: given a $z$ and a $d$, is $z$ a global minimizer of $w \to E_n(d, w)$?

Following the proof of Theorem 1, there is a polynomial $f_n(d, w, \beta)$ of degree 4 such that $f_n(d, w, \beta) \geq 0 \; \forall d, w, \beta$ and $f_n(d, w, \beta) = 0$ iff $\beta$ represents the intermediate results of the computation of SLP $E_n$ on input $(d, w)$. Note that $f_n$ can be computed (uniformly in $d$) by an SLP, and these SLPs are uniformly given. This holds because $\{E_n\}_n$ is an SLP energy family.

The above decision problem is in *co*-$\mathbf{NP}_{\mathbb{R}}$ over the reals. Therefore, it can be reduced to a polynomial $F_{n,d,z}(w, \beta^{(1)}, \beta^{(2)}, u_1, u_2)$ of degree 4 such that $z$ is <u>not</u> a minimizer of $w \to E_n(d, w)$ iff $F_{n,d,z}$ has a real zero. Here, the point is that this reduction can be performed uniformly in $z, d$ by an SLP (a general BSS reduction already exists according to [2]). In fact, $F_{n,d,z}$ is given as

$$f_n(d, w, \beta^{(1)}) + f_n(d, z, \beta^{(2)}) + (\beta_\ell^{(2)} - \beta_\ell^{(1)} - u_1^2)^2 + (u_1 \cdot u_2 - 1)^2 \ .$$

If $F_{n,d,z}(w, \beta^{(1)}, \beta^{(2)}, u_1, u_2) = 0$ the first two terms guarantee that $\beta^{(1)}$ and $\beta^{(2)}$ represent the correct computation for $E_n(d, \bullet)$ on inputs $w$ and $z$, respectively; the last two terms guarantee the result $\beta_\ell^{(1)}$ for $w$ to be smaller than the result $\beta_\ell^{(2)}$ for $z$.

Finally, we see that computing the max-min of $F_{n,d,z}(w, \beta^{(1)}, \beta^{(2)}, u_1, u_2)$ helps to compute the minimum of $E_n(d, \bullet)$. Denote $x := (w, \beta^{(1)}, \beta^{(2)}, u_1, u_2)$ and $F_{n,d}(z, x) := F_{n,d,z}(x)$. Then, we know that $\min\limits_{z,x} F_{n,d}(z, x)$ is $\geq 0$. A point $z$ is the minimum for $E_n(d, \bullet)$ if $x \to F_{n,d}(z, x)$ has only values $> 0$, i.e. if $\min\limits_{x} F_{n,d}(z, x) > 0$. Thus, we have to find a $z$ for which this minimum is as large as possible. This means we have to compute $\max\limits_{z} \min\limits_{x} F_{n,d}(z, x)$. If it is attained in $\tilde{z}$ and the value is $> 0$, then $\tilde{z}$ is the minimum of $w \to E_n(d, w)$. The reduction functions are computable by uniform SLP families.

The argument works the same for more alternations of min and max. The reduction increases the number of operators by 1; thus, the resulting problem is again an instance of our problem under consideration. □

## 7 Conclusions and further research directions

The previous sections presented a step towards a possible foundation of analog versions of notions like $\mathbf{P}, \mathbf{NP}$, reducibility and completeness in the framework of using dynamical systems as continuous-time computational devices.

Our approach focuses on the structure of energy functions which describe problem solutions through its minima. This structure is mainly determined by the operation set over which the BSS model is considered. As long as we deal with arithmetic operations, the energy functions naturally are related to rational functions. One could easily enlarge the set $\mathcal{O}$ in order to obtain more general classes of energies, for example including trigonometric functions. Proofs given here can be extended along the lines of [6] and [5]. Such enlarged sets of operations are particularly important in some of the ongoing related work. In [13, 14], neural networks with nonlinear activation functions are considered as dynamical systems admitting Lyapunov functions. In [4], analog computational devices using particular analytical functions are investigated w.r.t. their computational power. A next step thus would be to work out precisely how far the present approach can be extended.

Another important research direction is closely related to [12]. Our results are not very sensitive with respect to a concrete notion of complexity for continuous-time systems as long as the main contribution to the complexity of solving problems is both determined by the structure of local minima and the minimization processes used. It is crucial to further develop the complexity theory of such processes in order to approach equilibrium points of dynamical systems. Of course, a tremendous amount of work has been done in numerical function optimization which could be useful here. Such an approach might follow the outline given in section 4 and the research done in [12, 11]. Note that even in presence of a Lyapunov function a dynamical system can be quite powerful from a computational point of view, see [13, 14]. Therefore, we believe that the further study should result in finer partitioning of the classes $\mathbf{U}$ and $\mathbf{NU}$.

The arising structural questions provide interesting further problems: are the classes **U** and **NU** distinct, and what about the relation between **NU** and **PU**. Optimization theory should provide a way to attack these questions and perhaps will help to prove the distinctness of these classes (which we conjecture to be the case). However, the related questions seem to be not at all trivial: for which kind of functions are there easily computable perturbations such that the resulting function has the same globally minimal value but a much easier structure of locally minimal values?

Let us point out that even though some aspects of our approach parallelize main features in classical complexity theory, this does not hold for all features. For example, in our setting it is not obvious how to check whether a given energy actually is unimodal, whereas the corresponding problem is not difficult for polynomial time Turing machines. This leads to questions in recursion theory, f.e. concerning representability. It could be interesting to study recursion theoretic issues also for analog computations. This again might link our approach to the work by Moore [8].

Our final remark addresses the relation of our classes to real number complexity classes in the model of Blum, Shub and Smale. Dealing with functions $f : \mathbb{R}^\infty \to \mathbb{R}^\infty$ let $\mathrm{FP}_{\mathbb{R}}$ denote those which are computable in polynomial time in the BSS model.

**Theorem 5.** $\mathrm{FP}_{\mathbb{R}} \subseteq \mathbf{U}$ □

The proof combines the one of Theorem 1 with the proof in [2] that $\mathrm{NP}_{\mathbb{R}}$ has complete problems. These proofs then are extended in order to show that polynomial time computations can be expressed by unimodal energy functions. Some care has to be taken w.r.t. additional variables introduced in the reduction proof of [2]. Due to the problems which can occur even when minimizing unimodal functions we conjecture the relation $\mathrm{FP}_{\mathbb{R}} \subseteq \mathbf{U}$ to be proper. Concerning nondeterministic computations over the reals the relations to **NU** are not clear; this is due to the fact that a reasonable notion of functions computable in nondeterministic polynomial time in the BSS model is missing. This is an interesting topic on its own.

## 8 Acknowledgement

We thank P. Orponen for pointing us to relevant literature.

## References

1. L. Blum, F. Cucker, M. Shub, S. Smale: Complexity and Real Computation. Springer-Verlag (1998).
2. L. Blum, M. Shub, S. Smale: On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. Bulletin American Mathematical Society **21** (1989), 1–46.
3. P. Bürgisser, M. Clausen, M.A. Shokrollahi: Algebraic Complexity Theory. Grundlehren der mathematischen Wissenschaften vol. 315 Springer Verlag (1997).
4. M.L. Campagnolo, C. Moore, J.F. Costa: An analog characterization of the subrecursive functions. In: P. Kornerup (ed.): Proc. of the 4th conference on Real Numbers and Computers, Odense University (2000), 91–109.
5. A. Hemmerling: Computability of String Functions Over Algebraic Structures. Mathem. Logic Quarterly 44, nr. 1 (1998), 1–44.

6. N. Megiddo: A general NP-completeness Theorem. In : From Topology to Computation: Proceedings of the Smalefest, Springer-Verlag, New-York, Berlin (1993), 432–442.
7. M. Minoux: Mathematical Programming, Theory and Algorithms. Wiley and Sons (1986).
8. C. Moore: Recursion Theory on the Reals and Continuous-time Computation. Theoretical Computer Science 162 (1996), 23–44.
9. C. Moore: Finite-Dimensional Analog Computers: Flows, Maps, and Recurrent Neural Networks. In: Proc. of the 1st International Conference on Unconventional Models of Computation, Auckland, Springer (1998).
10. P. Orponen: A survey of continuous-time computation theory. Advances in Algorithms, Languages, and Complexity (eds. D.-Z. Du, K.-I Ko), Kluwer Academic Publishers, Dordrecht (1997), 209–224.
11. H.T. Siegelmann, A. Ben-Hur, S. Fishman: Computational Complexity for Continuous Time Dynamics. Physical Review Letters 83(7) (1999), 1463–1466.
12. H.T. Siegelmann, S. Fishman: Computation by Dynamical Systems. Physica D 120, (1998), 214–235.
13. J. Šima, P. Orponen: Computing with continuous-time Liapunov systems. Proc. of the 33rd Annual ACM Symposium on Theory of Computing, Crete, Association for Computing Machinery, New York (2001), 722–731.
14. J. Šima, P. Orponen: Exponential Transients in Continuous-Time symmetric Hopfield nets. To appear in: Proc. ICANN'2001: International Conference on Artificial Neural Networks, Vienna, (2001).