



HAL
open science

Constraint Based Computation of Periodic Orbits of Chaotic Dynamical Systems

Alexandre Goldsztejn, Laurent Granvilliers, Christophe Jermann

► **To cite this version:**

Alexandre Goldsztejn, Laurent Granvilliers, Christophe Jermann. Constraint Based Computation of Periodic Orbits of Chaotic Dynamical Systems. 19th International Conference on Principles and Practice of Constraint Programming (CP'13), 2013, Sweden. pp.774-789. hal-00867986

HAL Id: hal-00867986

<https://hal.science/hal-00867986v1>

Submitted on 1 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Constraint Based Computation of Periodic Orbits of Chaotic Dynamical Systems

Alexandre Goldsztejn¹, Laurent Granvilliers², and Christophe Jermann²

¹ CNRS, LINA (UMR-6241)

² Université de Nantes, LINA (UMR-6241)

2 rue de la Houssinière, 44300 Nantes, France

firstname.surname@univ-nantes.fr

Abstract. The chaos theory emerged at the end of the 19th century, and it has given birth to a deep mathematical theory in the 20th century, with a strong practical impact (e.g., weather forecast, turbulence analysis). Periodic orbits play a key role in understanding chaotic systems. Their rigorous computation provides some insights on the chaotic behavior of the system and it enables computer assisted proofs of chaos related properties (e.g., topological entropy).

In this paper, we show that the (numerical) constraint programming framework provides a very convenient and efficient method for computing periodic orbits of chaotic dynamical systems: Indeed, the flexibility of CP modeling allows considering various models as well as including additional constraints (e.g., symmetry breaking constraints). Furthermore, the richness of the different solving techniques (tunable local propagators, search strategies, etc.) leads to highly efficient computations. These strengths of the CP framework are illustrated by experimental results on classical chaotic systems from the literature.

Keywords: Chaotic dynamical systems; periodic orbits; topological entropy; numerical constraint satisfaction; symmetry breaking.

1 Introduction

A dynamical system is defined by a *state space* X (here $X \subseteq \mathbb{R}^d$, so a state is a vector of d reals) and an *evolution function*, which describes how the state $x_t \in X$ changes as time t passes. Continuous time dynamical systems (i.e., $t \in \mathbb{R}$) usually involve differential equations; in this case the evolution function is often called a flow. In this paper, we focus on discrete time dynamical systems (i.e., $t \in \mathbb{Z}$). They arise either from discrete models or discretizing continuous time dynamical systems (e.g., using the Poincaré map). In this case, the evolution function is a *map* $f : X \rightarrow X$, and the evolution of an initial condition $x_0 \in X$ is computed by $x_{k+1} = f(x_k)$, giving rise to (*forward*) *orbits* (x_0, x_1, x_2, \dots) . Understanding the infinitely complex structure of orbits generated by very simple systems is the goal of the chaos theory.

The first evidence of chaotic behavior was found by Poincaré while partially solving seemingly simple three-body problem at the end of the 19th century: The two-body problem, consisting of computing the trajectory of two masses following Newton's gravitational laws, is easily solved and fully understood. Poincaré proved that infinitely

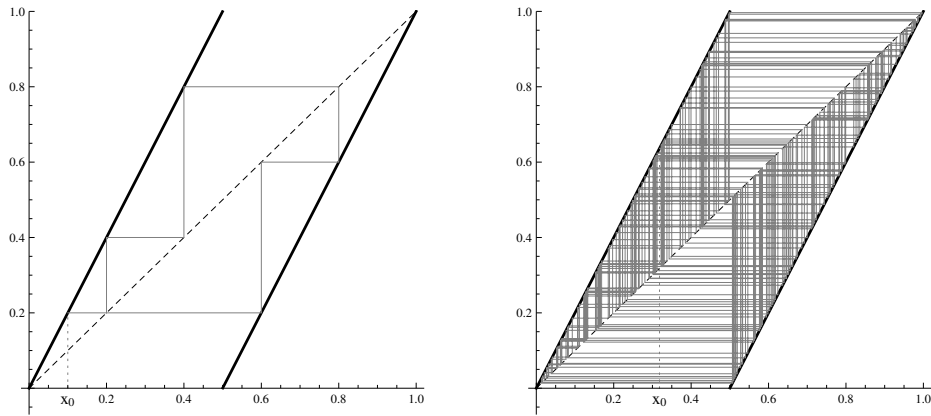


Fig. 1. The graph of the Dyadic map in thick black. Left: The orbit of $x_0 = 0.1$ in gray. Right: The orbit of $x_0 = \frac{1}{\pi}$ in gray.

complex trajectories arise when three bodies are considered, leading to the modern theory of chaos. One main discovery of Poincaré was the critical importance of periodic orbits in chaotic dynamical systems. Deep theoretical developments have followed during the 20th century (Markov partitions and corresponding symbolic dynamics, measure preserving maps and ergodicity, hyperbolicity, etc.) providing an increasingly accurate understanding of chaotic systems. Starting from the middle of that century, extensive simulations with computers (starting with Lorenz butterfly chaotic attractor) offered many illustrations of those chaotic behaviors, allowing these ideas to be disseminated toward the general public.

Formally defining a chaotic dynamical system is difficult: There exist several such definitions (Li-York chaos, Devaney chaos, positive topological entropy, see [20,6,21]), which are not equivalent and whose relationship is a current research topic. The common idea that chaos is the exponential sensitivity to initial conditions is wrong: Consider, e.g., the dynamical system defined by $X = \mathbb{R}^d$ and $f(x) = 2x$. Therefore $x_k = 2^k x_0$, so two different initial conditions diverge exponentially fast³ while this simple linear system is definitely not chaotic. However, enforcing some kind of exponential divergence between the orbits of neighbor initial conditions within a *bounded* state space X leads to very complex systems. Such systems need to be simultaneously expanding (so as to show divergence) and contracting (since the state space is bounded). This leads to hyperbolic dynamical systems, which are consistently contracting in some directions and expanding in the other directions, the most well-understood chaotic behavior.

The Dyadic map is among the most simple systems showing hyperbolic chaos. It is defined by $X = [0, 1)$ and $f(x) = 2x \bmod 1$. Multiplying by two is expanding, while taking modulo 1 enforces a contraction back to $[0, 1)$. Its graph is shown in Fig. 1, together with the orbit of $x_0 = 0.1$ (respectively $x_0 = \frac{1}{\pi}$) on the left (respectively

³ Indeed $d(x_k, y_k) = d(2^k x_0, 2^k y_0) = 2^k d(x_0, y_0)$.

right) hand side graphic. Orbits of this map are easily visualized: An initial point x is moved vertically toward the graph of f (the thick line in Fig. 1) in order to reach $f(x)$, and then horizontally toward the line $x = y$ (the dashed line in Fig. 1) in order to reach $x = f(x)$; Repeating the process from the new point yields the orbit. On the left hand side graphic, $f(0.1) = 0.2$, and the orbit of 0.2 is periodic with period 4 (i.e., $f^4(0.2) = f(f(f(f(0.2)))) = 0.2$). This map has a striking interpretation when considering the binary representation of a real number $x = 0.b_1b_2b_3 \dots \in [0, 1)$, each b_i being a bit in $\{0, 1\}$: Indeed, multiplying x by two shifts left its binary representation (yielding $b_1.b_2b_3 \dots$), and the modulo 1 then removes the first bit on the left (yielding $0.b_2b_3 \dots$). Hence the Dyadic map is in direct correspondence with the shift map on one-sided infinite bit sequences. This is a simple example of the powerful tool symbolic dynamics represents for investigating chaotic dynamical systems. It has several important consequences here: First, it is well known that a real number is rational if and only if the binary representation of its fractional part is periodic after a given bit (the same actually holds in any base). For example, the binary representation of 0.1 is $0.00011001100110011 \dots$. Hence any rational number will converge to a periodic orbit after a finite number of applications of the Dyadic map; the period of this orbit is equal to the period of the binary representation. It follows there are exactly $2^n - 1$ initial states⁴ yielding orbits of period n , called period- n orbits thereafter, and they are equally distributed within $[0, 1)$. On the contrary, irrational numbers are not periodic (the right hand side graphic of Fig. 1 shows the first 200 iterates of the orbit of $\frac{1}{\pi}$, which is seemingly random). Second, when computing an orbit using a computer, a finite binary representation has to be used. Therefore, any finite precision simulation has to converge toward zero. The 200 iterates of the orbit of $\frac{1}{\pi}$ shown in Fig. 1 have been computed using a 200-bit precision arithmetic.

The topological entropy is a real number associated to a dynamical system, which is meant to characterize the exponential divergence of orbits within a bounded state space. Suppose one can distinguish two points only if their distance is greater than $\epsilon > 0$, and consider a set $E \subseteq X \subseteq \mathbb{R}^d$ such that one can distinguish all points, so the mutual pairwise distance of the points in E is at least ϵ (such a set is called ϵ -separated). In this case, non intersecting balls of radius $\frac{\epsilon}{2}$ can be put around each point of E , entailing the cardinality of E to be at most $V_X/V_{\frac{\epsilon}{2}}$, where V_X and $V_{\frac{\epsilon}{2}}$ are respectively the volume of X and the volume⁵ of the ball of radius $\frac{\epsilon}{2}$. A map f can improve the situation by separating initial points that were too close to be distinguished, leading to the definition of (n, ϵ) -separated sets: A set $E \subseteq X$ is (n, ϵ) -separated if two different points in E yield orbits that are separated by at least ϵ within n iterations of the map. Formally, for all $x, y \in E$ with $x \neq y$, $\max_{0 \leq k \leq n} d(f^k(x), f^k(y)) \geq \epsilon$. The maximal cardinality of (n, ϵ) -separated sets is denoted by $s(n, \epsilon)$. As mentioned above, $s(0, \epsilon) \leq V_X/V_{\frac{\epsilon}{2}}$, while iterating the map can only help distinguishing more points, so $s(n, \epsilon)$ is increasing with respect to n . The growth rate of $s(n, \epsilon)$ shows how quick the map separates points. In particular, whenever the growth rate is exponential for some $\epsilon > 0$, i.e., $s(n, \epsilon) \approx ae^{bn}$ for some non-negative real constants a and b , the topological

⁴ There are 2^n binary representation of period n , but $0.1111 \dots$ is a periodic binary representation equal to 1 hence outside $[0, 1)$.

⁵ Volumes are generalized by the Lebesgue measure in space of dimension greater than 3.

entropy of the map f is defined as $h_X(f) = b$. More precisely,

$$h_X(f) = \limsup_{\epsilon \rightarrow 0} \limsup_{n \rightarrow \infty} \frac{\log s(n, \epsilon)}{n}, \quad (1)$$

where the first limit is used because $s(n, \epsilon)$ is non decreasing in ϵ , and supremum limits are used in order to take into account irregular exponential growths. When the topological entropy is strictly positive, the cardinality of maximal (n, ϵ) -separated sets grows exponentially with n . Therefore, the minimal distance between points in a maximal (n, ϵ) -separated set decreases exponentially with n , while the map f still allows separating them by at least ϵ in at most n iterations. Hence the map induces an exponential expansion in spite of the bounded state space. Having a strictly positive topological entropy is the characterization of chaos that is most often used.

For the Dyadic map, one can easily see that the set of points that yield period- n orbits is $(n, 0.5)$ -separated: Indeed, two such points differ in (at least) one bit among their n first bits, say the k^{th} bit. Hence, iterating the map $k - 1$ times brings those two different bits at the first (fractional) place, so the distance between their $(k - 1)^{\text{th}}$ iterates is at least 0.5. Now, since there are $2^n - 1 \approx e^{n \log 2}$ such points yielding period- n orbits, the topological entropy of the Dyadic map is at least $\log 2$. As seen on this example, the topological entropy is closely related to the exponential growth of the number P_n of period- n orbits with respect to n . More generally, under the hypothesis that the system satisfies the axiom A hypothesis [20] (roughly speaking, it is hyperbolic), its topological entropy is equal to

$$h_X(f) = \limsup_{n \rightarrow +\infty} \frac{\log(P_n)}{n}. \quad (2)$$

Numerous techniques have been developed to provide computer assisted proofs of chaos related properties: E.g., the famous answer to Smale's 14th problem [29], and [1,19,3,30,27,13]. Proving that a dynamical system is chaotic is generally done by finding out a subsystem with known topological entropy (often by identifying some specific periodic orbits), leading to a certified lower bound on its topological entropy. Roughly speaking, the system is proved to be as complex as a known chaotic dynamical system. An upper bound on the topological entropy provides an estimate of the accuracy of the certified lower bound, but such upper bounds are difficult to obtain: [27] provides such an upper bound for one dimensional maps. On the other hand, [9,10,11] proposed to compute all periodic orbits up to a given period with certified interval techniques, hence inferring an approximation of the topological entropy using Eq. (2).

We show here that using CP for rigorously computing periodic orbits is a convenient and efficient approach: By benefiting of constraint propagation and symmetry breaking, a simple model can be used, while avoiding heavy preprocessing (Section 2). Furthermore, the CP framework allows tuning the propagation strength and the search strategy (Section 3) so as to achieve more efficient resolution (experiments on well-known chaotic systems are reported in Section 5). We show in particular that the solving process can be tuned for small periods, impacting the resolution for higher periods.

2 Modeling the problem

After briefly recalling the basics of numerical constraint modeling, we introduce NCSPs whose solutions provide the periodic orbits of discrete time dynamical systems. Two standard models described in [9] are discussed with respect to numerical constraint solving. The flexibility of CP modeling allows considering alternative models.

2.1 NCSPs and interval arithmetic

Numerical constraint satisfaction problems (NCSPs) have variables representing real quantities, whose domains are thus subsets of \mathbb{R} . Their constraints are typically equations and inequalities on these quantities. For practical reasons, the domains are handled as intervals and the assignments are not enumerated. Instead, domains are split and filtered until a prescribed precision is reached. Interval arithmetic [24] allows enclosing the results of set-wise operations, and accounts for floating-point computational errors.

In this paper we denote $x = (x_1, \dots, x_n)$ the variables, considered to be a n -dimensional vector for convenience. We also denote x a real assignment of the variables, i.e., a point $(x_1, \dots, x_n) \in \mathbb{R}^n$. Intervals are denoted using bold-faced letters. Hence, the domains of x are denoted $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, considered as a n -dimensional vector of intervals, also called a *box*. We denote \mathbf{f} an interval extension of a function f , i.e., a function which computes an interval $\mathbf{f}(x)$ enclosing all the possible values of $f(x)$ for any real $x \in \mathbf{x}$. This definition naturally extends to function vectors f .

Interval arithmetic suffers from two problems: The *dependency problem* by which multiple occurrences of the same sub-expression are considered independent (e.g., $x - x$ evaluates to 0 for any real in $x \in [0, 1]$, but its interval evaluation at $\mathbf{x} = [0, 1]$ is $[-1, 1]$); And the *wrapping effect* by which the exact evaluation of an expression on an interval is in general poorly approximated using a single interval (e.g., $\frac{1}{x}$ evaluated at any $x \in [-1, 1]$ yields a real in $(-\infty, -1] \cup [1, +\infty)$ but its interval evaluation at $\mathbf{x} = [-1, 1]$ results in $(-\infty, +\infty)$). In addition, the practical use of floating-point computations induces the necessity of rigorous encapsulation of rounding-errors. These issues lead to potentially large over-approximations and must be carefully handled.

2.2 Folded models of periodic orbits

Given a map f on a state space $X \subseteq \mathbb{R}^d$, we can characterize a period- n orbit with the fixed-point relation

$$x = \underbrace{f \circ f \circ \dots \circ f}_{n \text{ times}}(x) = f^n(x). \quad (3)$$

Imposing it as a constraint on variables x with domains⁶ X results in the NCSP *folded model* whose solutions are the initial states $x \in X$ of period- n orbits.

⁶ In theory X may not be representable as a box, and the domains should be set to the smallest enclosing box. In practice however, the state spaces of classical chaotic maps are boxes.

Example 1. The famous *Logistic map* [23] is defined as $f(x) = rx(1 - x)$ on $X = [0, 1]$. It models the evolution of a population (x is the ratio to a maximum population) depending on a parameter $r \in \mathbb{R}^+$ representing a combined rate of reproduction and starvation. Despite its very simple formulation, this map has a chaotic behavior for some values of its parameter, e.g., $r := 4$. The folded model for period-2 orbits with this setting has a single variable x and a single constraint $x = f(f(x)) = -256x^4 + 512x^3 - 320x^2 + 64x$. Its four solutions are 0 , $\frac{3}{4}$ and $(5 \pm \sqrt{5})/8$, the two first ones being in fact fixed-points (period-1 orbits), the others constituting the only period-2 orbit.

Folded models present two major drawbacks when addressed with interval-based constraint solving methods. First, as soon as the map function contains more than one occurrence of a variable, the numbers of operations and occurrences of this variable in the constraint grow exponentially with the period. Though the factorized expression can still be compactly represented with a DAG, this cripples its interval evaluation by exacerbating both the dependency problem and the wrapping effect. This is even worse for the evaluation of the derivatives of the constraint, required to use interval Newton operators for proving the existence of real periodic orbits within boxes. Second, their solutions are the initial states of periodic orbits, but any point in a periodic orbit is an initial state for this orbit. Hence, as exemplified above, they have n solutions for each period- n orbit.

It is worth noting that the constraint can sometimes be simplified. For instance, that of the Dyadic map (see Section 1) can be rewritten $x = 2^n x \bmod 1$, and the Logistic map function can be reformulated as $f(x) = \frac{r}{4} - r(x - \frac{1}{2})^2$. Such simplifications may reduce the over-approximations of interval arithmetic. Still, the intrinsic complexity of the model remains as an initial box forcibly grows exponentially in size with the iterations of the map due to its chaotic nature.

2.3 Unfolded models of periodic orbits

The NCSP *unfolded model* aims at finding complete periodic orbits at once. Its variables (x_0, \dots, x_{n-1}) represent the consecutive n states in an period- n orbit, each x_k being itself a vector of d variables with domains X . The constraints establish the links between consecutive points

$$x_{(k+1) \bmod n} = f(x_k) \quad k \in \{0, \dots, n-1\}. \quad (4)$$

Example 2. The unfolded model for the period-2 orbits of the Logistic map with $r := 4$ is composed of the variables (x_0, x_1) and the constraints $x_1 = 4x_0(1 - x_0)$ and $x_0 = 4x_1(1 - x_1)$. Its four solutions are $(0, 0)$, $(\frac{3}{4}, \frac{3}{4})$, $(\frac{5-\sqrt{5}}{8}, \frac{5+\sqrt{5}}{8})$ and $(\frac{5+\sqrt{5}}{8}, \frac{5-\sqrt{5}}{8})$. It is now obvious the first ones are fixed points, and the others represent the same orbit.

This model has the strong advantage that constraint expressions remain identically complex (as many operators and variable occurrences) when n grows, making it much more appropriate for constraint methods. However, it has $n \times d$ variables instead of d variables in the folded model, and its search space thus grows exponentially with the period n . This drawback must be balanced with the fact having the n states as variables allows connecting the states in the same orbit, defining more freely strong

pruning operators involving several states, and splitting at any state during the search, definite advantages when taking into account the explosive nature of chaotic maps.

2.4 Other models of periodic orbits

The flexibility of the CP framework makes it possible to consider alternative models to the two classical ones presented above. For instance, both folded and unfolded models naturally have a functional form, but it is sometimes interesting, e.g., in order to reduce variable occurrences, to manipulate symbolically each constraint as a relation. This can yield *relational* unfolded models of the form

$$F(x_{(k+1)\bmod n}, x_k) = 0 \quad k \in \{0, \dots, n-1\} \quad (5)$$

whose interest will be illustrated in Section 5. It is also possible to reduce the search space by considering as variables only a fraction of the states in a periodic orbit, yielding *semi-unfolded models*. This could allow experimentally seeking an efficient trade-off between the folded and unfolded models, though in this paper we will focus only on those extremes in order to clearly illustrate their strengths and weaknesses.

2.5 Taking into account additional properties

A nice feature of CP is its ability to include additional knowledge on the considered problem as constraints or within initial domains, yielding a variety of complemented models whose efficacy can then be tested.

Periodic orbits have an inherent cyclic state symmetry. It is difficult to handle it in folded models, but it naturally boils down to a cyclic variable symmetry in unfolded models, and can then be (partially) broken using the lex-leader constraints relaxation proposed in [14]:

$$x_{0,0} \leq x_{k,0} \quad k \in \{1, \dots, n-1\}, \quad (6)$$

where $x_{k,0}$ represents the first coordinate of state k . Note that the symmetry could be broken using any other coordinate. Though inducing only a partial symmetry breaking, these additional constraints reduce optimally the search space.

Example 3. The additional partial symmetry breaking constraint for the unfolded model whose solutions are the period-2 orbits of the Logistic map is $x_0 \leq x_1$. It allows discarding the fourth solution, $(\frac{5+\sqrt{5}}{8}, \frac{5-\sqrt{5}}{8})$, as it is symmetric to the third one. It also halves the search space which is computationally very interesting.

Another property of the considered problem is that period- m orbits for any factor m of n are solutions of any NCSP model for period- n orbits. E.g., the two fixed-points of the Logistic map are solutions of models for any period n . In theory, additional constraints of the form $x_i \neq x_j$ for all $0 \leq i < j < n$ would discard these *factor orbits*, but such constraints cannot be filtered with interval solving methods and are thus useless.

Many chaotic maps have been extensively studied and a lot of knowledge has been accumulated about them. For instance, the *trapping region* of a map f on X , i.e., the

state subspace $X' \subseteq X$ whose image through the map $f(X')$ is strictly enclosed in X' , may be known to be enclosed within an ellipsoid or a polytope. Since periodic orbits starting within a trapping region must belong entirely to this trapping region, we can restrict the search to the enclosing shape using some additional inequality constraints.

Another example is the *non-wandering part* (NWP) of a map f on X , i.e., the set of points $x \in X$ such that any neighborhood U of x verifies $f^n(U) \cap U \neq \emptyset$ for some $n > 0$. This set comprises all periodic orbits and can be approximated using a simple subdivision algorithm: Consider the directed graph whose vertices are the boxes in a regular ϵ -precise subdivision of X and whose arcs $x \rightarrow x'$ verify $f(x) \cap x' \neq \emptyset$; Removing iteratively sinks and sources in this graph yields an ϵ -precise approximation of the NWP of f . This paving can be used to setup the domain of the initial state of an orbit, as proposed in [9]. Its size however grows quickly with ϵ and it is difficult to predict the appropriate precision without a dedicated study of the considered map.

3 Solving the problem

The standard complete constraint solving method is the branch&prune algorithm. It iteratively selects a box, prunes it using local consistency enforcing operators and interval methods (jointly designated as *contractors* in the following), checks if it contains a single solution and, otherwise, splits it into sub-boxes to be further processed. In this section we discuss the appropriate components of a branch&prune algorithm for solving NCSP models of periodic orbits of chaotic maps.

3.1 Pruning periodic orbits domains

The basic pruning algorithm for NCSPs is an AC3-like fixed-point loop over simple, and inexpensive, contractors like, e.g., BC3-revise [5], HC4-revise [4] or MOHC-revise [2]. It is however sometimes needed to resort to stronger contractors in order to avoid too much splitting, on trade-off with more demanding computations at each node of the search-tree. This can be achieved using for instance a fixed-point of 3B (or more generally k B) [22] or CID [28] operators. Finally, it is essential in this work that the returned solutions are proven to enclose a unique periodic orbit of the considered map, otherwise no valid reasoning on the map (e.g., its topological entropy) could be derived. For this purpose, it is typical to use an interval Newton operator [25], providing in addition a more global consistency.

In this paper we consider essentially two pruning procedures: *BC5*, a fixed-point of HC4-revise and BC3-revise⁷ contractors followed by an interval Newton application; and *BC5+CID(k)*, i.e., BC5 involving in addition CID(k) contractors⁸ during the fixed-point phase.

⁷ Typically generated for variables with multiple occurrences only.

⁸ One CID(k) contractor for a variable x slices the domain of x into k parts, computes a fixed-point of HC4-revise contractors for all constraints and variables on each slice, and eventually takes the hull of all the pruned slices.

3.2 Splitting periodic orbits domains

The standard splitting strategy for NCSPs is *round-robin* with bisection, which selects each time the next variable and splits its domain interval at its midpoint. Another typical strategy is *maxdom* which selects the variable with the largest domain.

The unfolded model for periodic orbits has a specific structure since variables are grouped into state coordinates and correspond to consecutive points in the orbit. We can thus consider dedicated splitting techniques, e.g., *initial-state* which splits only the coordinates of the initial state x_0 in the orbit, counting on pruning operators to reduce the domains of the other states. This idea was advanced in [9] as a mean of reducing the dimension of the search space.

Due to the explosive nature of chaotic maps, we think however that splitting all the states domains should pay-off. This will be confirmed in section 5 where we compare classical splitting strategies (*round-robin* and *maxdom*) on all variables to the dedicated *initial-state* splitting strategy.

3.3 Post-processing solution boxes

The branch&prune algorithm we have described outputs two types of boxes: Safe boxes which have been successfully certified to enclose a unique periodic orbit, and unsafe boxes which are not certified but have reached the prescribed maximum precision for the computation. When the partial symmetry breaking constraints (6) are used, boxes for which the corresponding strict inequalities are not certainly satisfied are also considered unsafe. Indeed such boxes may each contain a representative of the same periodic orbit. Note however that this never happened in the experiments reported in Section 5.

Unsafe boxes must be properly handled so as to allow rigorously counting the number of real periodic orbits. For this purpose, we apply a post-process that tries to certify them using a specific version of the interval-based Newton operator with *inflation* [18]. This operator acts like an interval local search algorithm, iteratively shifting and inflating slightly an initial box x so as to find a close box x' that can be certified. If it succeeds, x' replaces x in the solution set, after checking it does not enclose a periodic orbit already found in another safe solution box. Possibly symmetric boxes must be merged before applying this post-process. In case unsafe boxes remain after this process, only a lower bound on the number of real periodic orbits is obtained.

4 Related work

In [9,10,11], an interval-based method dedicated to computing periodic orbits of chaotic maps is proposed. It amounts to a bisection algorithm which splits the domains of the orbit, using interval forward evaluation of the map along the orbit to discard boxes that provably do not contain any periodic solution, and applying an interval Newton operator to certify that a box contains a single solution. When the map is symbolically invertible, it uses both forward and backward interval evaluation along the orbit in a fixed-point manner. Several key ingredients are identified in [9] as essential to the efficiency of this method: The usage of the unfolded model and of the map symbolic inverse, the *initial-state* splitting strategy, and some preprocessing using the non-wandering part and some

trapping region enclosure. Most of them are made unnecessary or even counterproductive by the CP framework, as illustrated by the experiments reported in the next section.

Non rigorous local methods for computing periodic orbits have also been investigated (see e.g., [8,26,7,12] and references therein). They usually work on the unfolded model, in a similar way as multiple shooting method for boundary value problems. Being incomplete, they are not used for estimating the topological entropy, although being useful for other purposes like computing a sample of longer periodic orbits.

5 Experiments

Constraint programming is implemented to handle several classical maps having different characteristics. Several issues are analyzed. What is the best way to model orbits? What are the good pruning and splitting techniques? Is it possible to take advantage of dedicated methods in the CP framework?

More precisely, we aim at comparing unfolded models with folded models, and the Cartesian form with the polar form of complex maps. Several splitting techniques (*maxdom*, *round-robin*, *initial-state*) are investigated. Different local consistency techniques are studied, in particular BC5 and BC5+CID(k). To this end, four standard maps are considered, namely Dyadic, Logistic, Hénon, and Ikeda. All techniques have been implemented in Realpaver [15] using default parameter settings. All experiments have been conducted on an Intel Core i7-620M 2.66GHz measured at 1666 MIPS by the Whetstone test.

We have implemented the previously introduced dedicated methods in our branch&prune algorithm, namely the non wandering part pre-paving, and trapping region constraints. In fact, we have observed that these methods do not change significantly the overall performances of the solving process. For instance, the solving time varies in proportion to $\pm 10\%$ (tested for Hénon and Ikeda) when the non wandering part is taken as input. It appears that propagation and split are together able to eliminate inconsistent regions of the search space without resorting to such methods.

For each problem, we found the theoretical number P_n of periodic orbits of the unidimensional maps, or the same number of periodic orbits as [9] for the Hénon and Ikeda maps. This number grows exponentially with n , i.e., $P_n \approx ae^{bn}$ where b approximates the topological entropy. As a consequence, the solving time t of the branch&prune algorithm must also grow exponentially with n . In fact, we aim at observing for a given map and a given strategy that $t \approx ce^{dn}$, where $d \geq b$ must hold since the solving process is complete. Therefore the difference $(d - b)$ quantifies the overall quality of this strategy. In the following, we will use a logarithmic scale on t and P_n to plot the results, the growth constants b and d corresponding to the slopes of the curves.

Remark: The experiments are carried out only for orbits of prime periods. Hence, possible issues of factor orbits and symmetry breaking are discarded, thus simplifying the post-processing phase and the interpretation of results. Following this approach still permits to compare the different techniques and to calculate accurate approximations of the topological entropy.

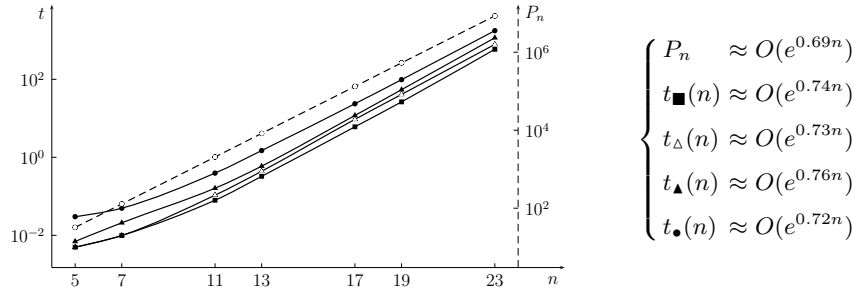


Fig. 2. Finding orbits of Dyadic and Logistic using BC5 with *maxdom*. Left: \circ is the number of solutions P_n ; the other curves represent the solving times of Dyadic's unfolded model and Logistic's unfolded model using the factorized expression (\blacksquare), Dyadic's folded model (Δ), Logistic's folded model using the factorized expression (\blacktriangle), and Logistic's unfolded model using the original expression (\bullet). Right: Empirical asymptotic laws of these different techniques.

5.1 Unidimensional maps

The two aforementioned unidimensional maps (Dyadic and Logistic) are interesting to illustrate the impact of modeling on the solving performance. Their folded models are simple enough, their number of operations growing linearly with n . The expression of Logistic can be factorized (the factorized form is used to generate the folded model). Dyadic is discontinuous due the modulo operation.

The topological entropy of these maps is equal to $\log 2$ since they have respectively 2^n (Logistic) and $2^n - 1$ (Dyadic) solutions. Their orbits are easily calculated by BC5 with *maxdom*, the number of splitting steps matching the number of solutions.

The results are depicted in Fig. 2. The topological entropy is the slope of the dashed line P_n . One can remark that the other curves corresponding to different models tend to become parallel to P_n , showing that the cost of calculating one solution is constant for all of them. Strikingly, the branch&prune algorithm behaves similarly when processing the folded models (curves Δ and \blacktriangle) and the unfolded models (curve \blacksquare). In fact, the unfolded models exploit symmetry breaking constraints that reduce P_n by a factor n . However, pruning the folded models is easier since only one BC3-revise operator is applied at each node of the search tree, while pruning the unfolded models calculates a fixed-point of n HC4-revise operators (one per constraint) followed by an application of the interval Newton operator. Logistic's original unfolded model is worse (curve \bullet), since it requires applying BC3-revise operators due to the multiple variable occurrences.

Discontinuous or non differentiable functions, involving e.g., the modulo operation, are seemingly taken into account with no additional cost. However, they possibly interfere with the certification procedure. For instance, solving Dyadic's unfolded model produces two non certified boxes. The first box encloses the fixed-point $(0, \dots, 0)$, which is located on the domain boundary. The second box $([1 - \epsilon, 1], \dots, [1 - \epsilon, 1])$ contains no solution but it cannot be discarded by interval methods.

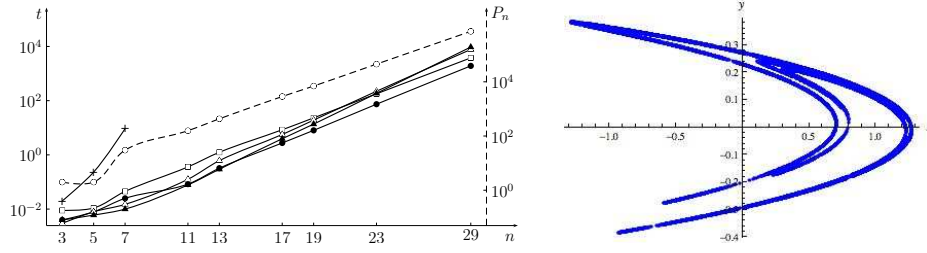


Fig. 3. Hénon map. Left: \circ is the number of solutions P_n ; \bullet is the solving time t of BC5 with *maxdom*; \triangle differs from \bullet in the use of the *round-robin* strategy; \blacktriangle differs from \bullet in the splitting of the initial state alone; \square differs from \bullet in the use of CID(3) operators; $+$ differs from \bullet in the use of the folded model. Right: Period-23 orbits, which clearly shows the well known strange attractor of the Hénon map.

5.2 Hénon map

The Hénon map [16] is defined as $f(x, y) = (y + 1 - ax^2, bx)$, the standard parameter values $a := 1.4$ and $b := 0.3$ leading to a chaotic behavior. Given $x_k, y_k \in [-2, 3]$, $0 \leq k \leq n - 1$, the unfolded model is as follows:

$$\begin{cases} x_{(k+1) \bmod n} = y_k + 1 - ax_k^2 \\ y_{(k+1) \bmod n} = bx_k \end{cases} \quad (7)$$

The results are depicted in Fig. 3. The number of solutions P_n (dashed curve) gives an approximation of the topological entropy as $\log(P_n)/n \approx 0.46$. As expected, the folded model (curve $+$) is not tractable since its size grows exponentially with n . The other techniques are all able to isolate and certify all the solutions in reasonable time for the considered periods, corroborating the results in [9]. The best splitting technique is *maxdom* (curve \bullet), compared to *round-robin* and *initial-state* (curves \triangle and \blacktriangle). Enforcing BC5+CID(3) (curve \square) seems to slow-down the solving phase but the growth constant is decreased from 0.55 to 0.51, demonstrating a better asymptotic behavior. In other words, we have $t_{\bullet}(n) \approx O(e^{0.55n})$ and $t_{\square}(n) \approx O(e^{0.51n})$.

We have also extracted from [9] the growth constant of the solving time t_G obtained from the best implemented method, which is approximatively equal to 0.58, i.e., $t_G(n) \approx O(e^{0.58n})$. Hence, on this problem the CP approach compares favorably in terms of complexity to the dedicated approach of [9].

5.3 Ikeda map

The Ikeda map [17] is defined as

$$f(z) = a + b \exp\left(i\kappa - \frac{i\alpha}{1 + |z|^2}\right) z \quad (8)$$

where z is a complex number. The classical setting $a := 1$, $b := 0.9$, $\alpha := 6$ and $\kappa := 0.4$ yields a chaotic behavior. This map can be transformed into a two-dimensional unfolded model⁹ over the real numbers in two ways: The Cartesian form $z = x + iy$

⁹ The folded model of this map is far too complex to be tractable by interval solving methods.

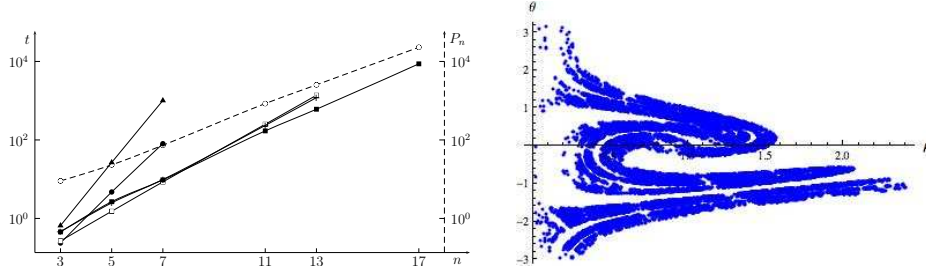


Fig. 4. Ikeda map. Left: \circ is the number of solutions P_n ; \blacktriangle is the solving time t of BC5 with *maxdom* applied to the Cartesian model; \bullet differs from \blacktriangle in the use of the polar model; \square improves \bullet with CID(3) operators; \blacksquare improves \bullet with CID(9) operators; $+$ differs from \blacksquare by *initial-state*. Right: Period-17 orbits, which clearly shows the well known strange attractor of the Ikeda map, although in polar coordinates here.

yields

$$\begin{cases} x_{(k+1) \bmod n} = a + b(x_k \cos u_k - y_k \sin u_k) \\ y_{(k+1) \bmod n} = b(x_k \sin u_k + y_k \cos u_k) \\ u_k = \kappa - \alpha / (1 + x^2 + y^2) \end{cases} \quad (9)$$

for $k = 0, \dots, n - 1$ and the polar form $z = \rho e^{i\theta}$ leads to the relational model

$$\begin{cases} \rho_{(k+1) \bmod n} \cos(\theta_{(k+1) \bmod n}) = a + b(\rho_k \cos(u_k)) \\ \rho_{(k+1) \bmod n} \sin(\theta_{(k+1) \bmod n}) = b(\rho_k \sin(u_k)) \\ u_k = \theta_k + \kappa - \alpha / (1 + \rho_k^2). \end{cases} \quad (10)$$

The domains can be defined as $x_k, y_k \in [-10, 10]$, $\rho_k \in [0, 10\sqrt{2}]$ and $\theta_k \in [-\pi, \pi]$ for every k . In both models, new variables $u_k \in (-\infty, +\infty)$ are added to share projections on common sub-expressions appearing in the constraints, hence augmenting the contraction power of interval constraint propagation. These variable domains are however never split, thus not increasing the size of the search space.

The results are depicted in Fig. 4. The number of solutions P_n (dashed curve) gives an approximation of the topological entropy as $\log(P_n)/n \approx 0.60$. We first compare the Cartesian model (curve \blacktriangle) with the polar model (curve \bullet) both handled by BC5 with *maxdom*. The growth constants for these models are respectively equal to 1.83 and 1.46, i.e. $t_{\blacktriangle}(n) \approx O(e^{1.83n})$ and $t_{\bullet}(n) \approx O(e^{1.46n})$, promoting the use of the polar model. However, even using the polar model, the solving strategy BC5 with *maxdom* remains very inefficient with respect to the approximate topological entropy. This led us to enforce stronger consistency techniques in order to decrease the number of splitting steps by an exponential factor.

The solving time is much improved by means of BC5+CID(3) (curve \square with growth constant 0.85) and especially BC5+CID(9) (curve \blacksquare with growth constant 0.66, i.e. $t_{\blacksquare}(n) \approx O(e^{0.66n})$), considering the polar model. Finally, as observed for the Hénon map, the other tested splitting strategies are counterproductive. In particular, this phenomenon is illustrated by replacing *maxdom* with *initial-state* and solving the polar

model with BC5+CID(9) (curve + with growth constant 0.81, i.e. $t_+(n) \approx O(e^{0.81n})$, to be compared to ■).

6 Discussion

Compared to the dedicated method proposed in [9,10,11], the CP framework offers a much more flexible, easy to deploy and to use environment. However, this high flexibility entails choosing the best combination of model and solving strategy. The results reported in Section 5 suggest that this choice can be performed as follows: The different combinations can be implemented to calculate period- n orbits for small values of n (e.g., with a timeout of a few minutes). On the basis of these results, the law $t \approx ce^{dn}$ can be approximated for each combination, by estimating the constants c and d , and the best combination can be used to solve the problem with greater periods.

A quantitative comparison of the respective efficiencies of the CP framework and the method of [9] is difficult to assess, since [9] does not focus on this aspect. Nevertheless, the asymptotic complexity, which does not depend on the computer, can be extracted from the results reported in [9] for the Hénon map: The time needed to compute all n -periodic orbits follows $t_G(n) \approx O(e^{0.58n})$. Our experiments on the Hénon map have shown an asymptotic time $t_{\square}(n) \approx O(e^{0.51n})$. This is a significant improvement with respect to the lower bound complexity $P_n \approx O(e^{0.46n})$.

On a qualitative perspective, the experiments reported in Section 5 allow arguing about several claims of [9]: First, the usage of local consistencies removes the necessity of symbolically inverting the map, which is critical for the efficiency of [9] but not always possible. Second, *initial-state* splitting strategy is not anymore a key ingredient for the efficiency, not even the best strategy in the CP framework. Finally, additional properties like the pre-computation of the non wandering part or some trapping region are not essential anymore: Local consistencies are able to efficiently remove boxes inconsistent with these additional properties using only the constraints $x_{(k+1) \bmod n} = f(x_k)$. In addition, the cost of their treatment may turn out to penalize the overall algorithm efficiency.

Future work shall tackle additional maps, including higher dimensional discrete time dynamical systems and ODE driven continuous time dynamical systems. One weakness of the approach, which is also pointed out in [9], is that the topological entropy approximation by counting the number of periodic orbits holds only for dynamical systems that satisfy the axiom A (although some exponential growth of the number of periodic orbits is a very strong hint of the presence of hyperbolic chaos in general). We shall investigate the possibility of providing some computer assisted proof of this property.

References

1. Arai, Z.: On hyperbolic plateaus of the Hénon map. *Journal Experimental Mathematics* 16(2), 181–188 (2007)
2. Araya, I., Trombettoni, G., Neveu, B.: Exploiting monotonicity in interval constraint propagation. In: *AAAI* (2010)

3. Banhelyi, B., Csendes, T., Garay, B., Hatvani, L.: A computer-assisted proof of σ_3 -chaos in the forced damped pendulum equation. *SIAM Journal on Applied Dynamical Systems* 7, 843–867 (2008)
4. Benhamou, F., Goualard, F., Granvilliers, L., Puget, J.F.: Revising hull and box consistency. In: *ICLP*. pp. 230–244 (1999)
5. Benhamou, F., McAllester, D., Van Hentenryck, P.: CLP(Intervals) revisited. In: *Procs. Intl. Symp. on Logic Prog.* pp. 124–138. The MIT Press (1994)
6. Blanchard, F., Glasner, E., Kolyada, S., Maass, A.: On Li-Yorke pairs. *Journal für die reine und angewandte Mathematik* 2002(547), 51–68 (2002)
7. Crofts, J.J., Davidchack, R.L.: Efficient detection of periodic orbits in chaotic systems by stabilizing transformations. *SIAM J. Sci. Comput.* 28(4), 1275–1288 (2006)
8. Davidchack, R.L., Lai, Y.C., Klebanoff, A., Bollt, E.M.: Towards complete detection of unstable periodic orbits in chaotic systems. *Physics Letters A* 287(12), 99 – 104 (2001)
9. Galias, Z.: Interval methods for rigorous investigations of periodic orbits. *International Journal of Bifurcation and Chaos* 11(09), 2427–2450 (2001)
10. Galias, Z.: Rigorous investigation of the Ikeda map by means of interval arithmetic. *Nonlinearity* 15(6), 1759 (2002)
11. Galias, Z.: Computational methods for rigorous analysis of chaotic systems. In: Kocarev, L., Galias, Z., Lian, S. (eds.) *Intelligent computing based on chaos*, *Studies in Computational Intelligence*, vol. 184, pp. 25–51. Springer-Verlag (2009)
12. Gao, F., Gao, H., Li, Z., Tong, H., Lee, J.J.: Detecting unstable periodic orbits of nonlinear mappings by a novel quantum-behaved particle swarm optimization non-Lyapunov way. *Chaos, Solitons & Fractals* 42(4), 2450 – 2463 (2009)
13. Goldsztejn, A., Hayes, W., Collins, P.: Tinkerbelle is chaotic. *SIAM Journal on Applied Dynamical Systems* 10(4), 1480–1501 (2011)
14. Goldsztejn, A., Jermann, C., Ruiz de Angulo, V., Torras, C.: Symmetry breaking in numeric constraint problems. In: *17th International Conference on Principles and Practice of Constraint Programming (CP'11)*. LNCS, vol. 6876, pp. 317–324 (2011)
15. Granvilliers, L., Benhamou, F.: Algorithm 852: Realpaver: an interval solver using constraint satisfaction techniques. *ACM Trans. Mathematical Software* 32(1), 138–156 (2006)
16. Hénon, M.: A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics* 50, 69–77 (1976)
17. Ikeda, K.: Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Opt Comm* pp. 257–261 (1979)
18. Ishii, D., Goldsztejn, A., Jermann, C.: Interval-based projection method for under-constrained numerical systems. *Constraints* 17(4), 432–460 (2012)
19. Kapela, T., Simó, C.: Computer assisted proofs for nonsymmetric planar choreographies and for stability of the Eight. *Nonlinearity* 20(5), 1241 (2007)
20. Katok, A., Hasselblatt, B.: *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press (1995)
21. Kolyada, S.F.: Li-Yorke sensitivity and other concepts of chaos. *Ukrainian Mathematical Journal* 56(8), 1242–1257 (2004)
22. Lhomme, O.: Consistency techniques for numeric CSPs. In: *IJCAI*. pp. 232–238 (1993)
23. May, R.M.: Simple mathematical models with very complicated dynamics. *Nature* 261, 459–467 (1976)
24. Moore, R.: *Interval Analysis*. Prentice-Hall (1966)
25. Neumaier, A.: *Interval Methods for Systems of Equations*. Cambridge University Press (1990)
26. Parsopoulos, K., Vrahatis, M.: Computing periodic orbits of nondifferentiable/discontinuous mappings through particle swarm optimization. In: *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*. pp. 34–41 (2003)

27. Sella, L., Collins, P.: Computation of symbolic dynamics for one-dimensional maps. *J. Comput. Appl. Math.* 234(2), 418–436 (2010)
28. Trombettoni, G., Chabert, G.: Constructive interval disjunction. In: CP. pp. 635–650 (2007)
29. Tucker, W.: A rigorous ODE solver and Smale’s 14th problem. *Found. Comput. Math.* 2, 53–117 (2002)
30. Wilczak, D., Żgliczynski, P.: Computer assisted proof of the existence of homoclinic tangency for the Hénon map and for the forced damped pendulum. *SIAM Journal on Applied Dynamical Systems* 8, 1632–1663 (2009)