Understanding Complex Systems

Founding Editor: S. Kelso

Future scientific and technological developments in many fields will necessarily depend upon coming to grips with complex systems. Such systems are complex in both their composition — typically many different kinds of components interacting simultaneously and nonlinearly with each other and their environments on multiple levels — and in the rich diversity of behavior of which they are capable.

The Springer Series in Understanding Complex Systems series (UCS) promotes new strategies and paradigms for understanding and realizing applications of complex systems research in a wide variety of fields and endeavors. UCS is explicitly transdisciplinary. It has three main goals: First, to elaborate the concepts, methods and tools of complex systems at all levels of description and in all scientific fields, especially newly emerging areas within the life, social, behavioral, economic, neuro- and cognitive sciences (and derivatives thereof); second, to encourage novel applications of these ideas in various fields of engineering and computation such as robotics, nano-technology, and informatics; third, to provide a single forum within which commonalities and differences in the workings of complex systems may be discerned, hence leading to deeper insight and understanding.

UCS will publish monographs, lecture notes, and selected edited contributions aimed at communicating new findings to a large multidisciplinary audience.

More information about this series at http://www.springer.com/series/5394
Springer Complexity

Springer Complexity is an interdisciplinary program publishing the best research and academic-level teaching on both fundamental and applied aspects of complex systems cutting across all traditional disciplines of the natural and life sciences, engineering, economics, medicine, neuroscience, social and computer science.

Complex Systems are systems that comprise many interacting parts with the ability to generate a new quality of macroscopic collective behavior the manifestations of which are the spontaneous formation of distinctive temporal, spatial or functional structures. Models of such systems can be successfully mapped onto quite diverse “real-life” situations like the climate, the coherent emission of light from lasers, chemical reaction-diffusion systems, biological cellular networks, the dynamics of stock markets and of the Internet, earthquake statistics and prediction, freeway traffic, the human brain, or the formation of opinions in social systems, to name just some of the popular applications.

Although their scope and methodologies overlap somewhat, one can distinguish the following main concepts and tools: self-organization, nonlinear dynamics, synergetics, turbulence, dynamical systems, catastrophes, instabilities, stochastic processes, chaos, graphs and networks, cellular automata, adaptive systems, genetic algorithms and computational intelligence.

The three major book publication platforms of the Springer Complexity program are the monograph series “Understanding Complex Systems” focusing on the various applications of complexity, the “Springer Series in Synergetics”, which is devoted to the quantitative theoretical and methodological foundations, and the “Springer Briefs in Complexity” which are concise and topical working reports, case studies, surveys, essays and lecture notes of relevance to the field. In addition to the books in these two core series, the program also incorporates individual titles ranging from textbooks to major reference works.

Editorial and Programme Advisory Board

Henry Abarbanel, Institute for Nonlinear Science, University of California, San Diego, USA
Dan Braha, New England Complex Systems Institute and University of Massachusetts Dartmouth, USA
Péter Erdős, Center for Complex Systems Studies, Kalamazoo College, USA and Hungarian Academy of Sciences, Budapest, Hungary
Karl Friston, Institute of Cognitive Neuroscience, University College London, London, UK
Hermann Haken, Center of Synergetics, University of Stuttgart, Stuttgart, Germany
Viktor Jirsa, Centre National de la Recherche Scientifique (CNRS), Université de la Méditerranée, Marseille, France
Janusz Kacprzyk, System Research, Polish Academy of Sciences, Warsaw, Poland
Kunihiko Kaneko, Research Center for Complex Systems Biology, The University of Tokyo, Tokyo, Japan
Scott Kelso, Center for Complex Systems and Brain Sciences, Florida Atlantic University, Boca Raton, USA
Markus Kirkilionis, Mathematics Institute and Centre for Complex Systems, University of Warwick, Coventry, UK
Jürgen Kurths, Nonlinear Dynamics Group, University of Potsdam, Potsdam, Germany
Andrzej Nowak, Department of Psychology, Warsaw University, Poland
Hassan Qudrat-Ullah, School of Administrative Studies, York University, Toronto, ON, Canada
Linda Reichl, Center for Complex Quantum Systems, University of Texas, Austin, USA
Peter Schuster, Theoretical Chemistry and Structural Biology, University of Vienna, Vienna, Austria
Frank Schweitzer, System Design, ETH Zürich, Zürich, Switzerland
Didier Sornette, Entrepreneurial Risk, ETH Zürich, Zürich, Switzerland
Stefan Thurner, Section for Science of Complex Systems, Medical University of Vienna, Vienna, Austria

xcfu@shu.edu.cn
# Contents

1. Discovering Cluster Dynamics Using Kernel Spectral Methods ........................................... 1  
   Rocco Langone, Raghvendra Mall, Joos Vandewalle  
   and Johan A.K. Suykens

2. Community Detection in Bipartite Networks: Algorithms and Case studies .......................... 25  
   Taher Alzahrani and K.J. Horadam

3. Epidemiological Modeling on Complex Networks .............................................................. 51  
   Zhen Jin, Shuping Li, Xiaoguang Zhang, Juping Zhang  
   and Xiao-Long Peng

4. Resilience of Spatial Networks .......................................................... 79  
   Daqing Li

5. Synchronization and Control of Hyper-Networks and Colored Networks .......................... 107  
   Xinchu Fu, Zhaoyan Wu and Guanrong Chen

6. New Nonlinear CPRNG Based on Tent and Logistic Maps .............................................. 131  
   Oleg Garasym, Ina Taralova and René Lozi

7. Distributed Finite-Time Cooperative Control of Multi-agent Systems .............................. 163  
   Yu Zhao, Guanghui Wen and Guanrong Chen

   Xiangyu Wang and Shihua Li
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Authors</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Application of Fractional-Order Calculus in a Class of Multi-agent Systems</td>
<td>Wenwu Yu, Guanghui Wen and Yang Li</td>
<td>229</td>
</tr>
<tr>
<td>10</td>
<td>Chaos Control and Anticontrol of Complex Systems via Parrondo’s Game</td>
<td>Marius-F. Danca</td>
<td>263</td>
</tr>
<tr>
<td>11</td>
<td>Collective Behavior Coordination with Predictive Mechanisms</td>
<td>Hai-Tao Zhang, Zhaomeng Cheng, Ming-Can Fan and Yue Wu</td>
<td>283</td>
</tr>
<tr>
<td>12</td>
<td>Convergence, Consensus and Synchronization of Complex Networks via Contraction Theory</td>
<td>Mario di Bernardo, Davide Fiore, Giovanni Russo and Francesco Scafuti</td>
<td>313</td>
</tr>
<tr>
<td>13</td>
<td>Towards Structural Controllability of Temporal Complex Networks</td>
<td>Xiang Li, Peng Yao and Yujian Pan</td>
<td>341</td>
</tr>
<tr>
<td>14</td>
<td>A General Model for Studying Time Evolution of Transition Networks</td>
<td>Choujun Zhan, Chi K. Tse and Michael Small</td>
<td>373</td>
</tr>
<tr>
<td>15</td>
<td>Deflection Routing in Complex Networks</td>
<td>Soroush Haeri and Ljiljana Trajkovic</td>
<td>395</td>
</tr>
<tr>
<td>16</td>
<td>Recommender Systems for Social Networks Analysis and Mining: Precision Versus Diversity</td>
<td>Amin Javari, Malihe Izadi and Mahdi Jalili</td>
<td>423</td>
</tr>
<tr>
<td>17</td>
<td>Strategy Selection in Networked Evolutionary Games: Structural Effect and the Evolution of Cooperation</td>
<td>Shaolin Tan and Jinhua Lü</td>
<td>439</td>
</tr>
<tr>
<td>18</td>
<td>Network Analysis, Integration and Methods in Computational Biology: A Brief Survey on Recent Advances</td>
<td>Shihua Zhang</td>
<td>459</td>
</tr>
</tbody>
</table>
Chapter 10
Chaos Control and Anticontrol of Complex Systems via Parrondo’s Game

Marius-F. Danca

Abstract In this chapter, we prove analytically and numerically aided by computer simulations, that the Parrondo game can be implemented numerically to control and anticontrol chaos of a large class of nonlinear continuous-time and discrete-time systems. The game states that alternating loosing gains of two games, one can actually obtain a winning game, i.e.: “losing + losing = winning” or, in other words: “two ugly parents can have beautiful children” (Zeilberger, on receiving the 1998 Leroy P. Steele Prize). For this purpose, the Parameter Switching (PS) algorithm is implemented. The PS algorithm switches the control parameter of the underlying system, within a set of values as the system evolves. The obtained attractor matches the attractor obtained by replacing the parameter with the average of switched values. The systems to which the PS algorithm based Parrondo’s game applies are continuous-time of integer or fractional order ones such as: Lorenz system, Chen system, Chua system, Rössler system, to name just a few, and also discrete-time systems and fractals. Compared with some other works on switch systems, the PS algorithm utilized in this chapter is a convergent algorithm which allows to approximate any desired dynamic to arbitrary accuracy.

10.1 Introduction

In [34, 36], Parrondo et al. showed that alternating the loosing gains of two games, one can actually obtain a winning strategy with a positive gain, i.e.

\[ \text{losing} + \text{losing} = \text{winning}. \] (10.1)
Since its discovery, this apparent contradiction has been known as Parrondo's paradox (or game, as we call in this work), becoming an active area of research for example in discrete-time ratchets [4], minimal Brownian ratchet [28], molecular transport [26], and so on. Parrondo's game is considered as game theory in the Blackwell sense [6] and in [2, 24] was extended from its original form to include player strategy. In [10, 11] a mechanism for pattern formation based on the alternation of two dynamics, is proposed. For a review of the history of Parrondo’s paradox, developments, and connections to related phenomena, see [1].

This kind of alternation between weakness and strength, order and chaos, or losing and winning, can be found or produced in physical, biological, quantum, mathematical systems and in control theory, or even fractals, where combining processes may lead to counterintuitive dynamics. The apparently trivial phenomenon seems to be typical not only for theoretical systems but also in nature, where there are many interactions due to some accidental or intentional parameter switches. Even more, there is a belief that this kind of mechanisms could be used as a possible explanation of the origin of life [18].

If we replace in Parrondo’s paradox the words “losing” with “chaos” and “winning” with “order” (as the opposite of chaos), then Parrondo’s game can be written in the following form:

\[ \text{chaos}_1 + \text{chaos}_2 = \text{order}, \]  

(10.2)

where \( \text{chaos}_{1,2} \) and \( \text{order} \) represent two chaotic dynamics and a regular dynamic respectively of a considered system. The form (10.2) of Parrondo’s game is exploited in e.g. [3], where it is used to study the effects of combining different dynamics of two real systems, and also in [39, 40] where alternations between two dynamics of quadratic maps are investigated. In [15, 17], the study was extended to complex systems (fractals).

Relation (10.2) can be considered as a new kind of chaos control in the sense that by alternating two chaotic dynamics, it is possible to obtain a regular dynamic. Similarly, one can imagine an anticontrol-like scheme as

\[ \text{order}_1 + \text{order}_2 = \text{chaos}. \]  

(10.3)

A natural question is if it is possible to generalize Parrondo’s game (10.2) in the sense that alternation between two dynamics in (10.2) is replaced with switches between \( N > 2 \) dynamics, i.e.

\[ \text{chaos}_1 + \text{chaos}_2 + \cdots + \text{chaos}_N = \text{order}, \]  

(10.4)

or

\[ \text{order}_1 + \text{order}_2 + \cdots + \text{order}_N = \text{chaos}. \]  

(10.5)

A positive answer is given in [39] for continuous time chaotic systems via the PS algorithm.
The goal of this chapter is to present a comprehensive account of the approaches used to define these chaos control-like and anticontrol-like algorithms, which are generalizations of Parrondo's paradox, via the PS algorithm.

10.2 Parameter Switching Algorithm

After presenting the general form of Parrondo's game, we describe the PS algorithm necessary to implement the Parrondo game. For this purpose, we have to choose a finite set of $N > 1$ parameters values, $\mathcal{P}_N = \{p_1, p_2, \ldots, p_N\}$, inside which the algorithm switches the control parameter $p$ as the considered continuous (discrete)-time system evolves. While for discrete-time systems, the algorithm simply switches $p$ every $m_i$ iterations, $i = 1, 2, \ldots, N$, for the continuous-time systems, the time interval where the system is defined $I = [0, T]$, for $T > 0$, is partitioned in short time subintervals $I_{i,j}$, for $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots$, each having length $m_i h$, $h$ being a small real value ($m_i$ being $p_i$ "weights"), such that $I = \bigcup_{j} \bigcup_{i=1}^{N} I_{i,j}$ (see the sketch in Fig. 10.1 for $N = 2$). While the underlying Initial Value Problem (IVP) is numerically integrated, the algorithm switches successively $p$ within $\mathcal{P}_N$ in the subintervals $I_{i,j}, i = 1, 2, \ldots, N, j = 1, 2, \ldots$, i.e. in $I_{1,1}, I_{2,1}, \ldots, I_{N,1}$, $I_{2,2}, \ldots, I_{2,N}$, $I_{3,1}, \ldots$ and so on, until the numerical integration ends.

For the sake of simplicity, hereafter the index $j$ will be dropped unless necessary.

For continuous-time systems, the resulted "switched" attractor approximates the "averaged" attractor which is obtained if the parameter $p$ is replaced with the average of the switched values, $p^*$ (see Fig. 10.1):

$$p^* := \frac{\sum_{i=1}^{N} m_i p_i}{\sum_{i=1}^{N} m_i}, \quad p_i \in \mathcal{P}_N. \quad (10.6)$$

Fig. 10.1 Time subintervals $I_{i,j}$ and the piece-wise constant function $p$, for the case $N = 2$ (sketch)
10.2.1 PS Algorithm Applied to Continuous-Time Systems

Consider a class of systems modeled by the following IVP:

\[ \dot{x}(t) = f(x(t)) + pA x(t), \quad t \in I = [0, T], \quad x(0) = x_0, \]  

(10.7)

for \( T > 0, x_0 \in \mathbb{R}^n, \ p \in \mathbb{R} \) the control parameter, \( A \in L(\mathbb{R}^n) \) and \( f : \mathbb{R}^n \to \mathbb{R}^n \) a nonlinear function.

The IVP (10.7) models a great majority of continuous nonlinear and autonomous dynamical systems depending on a single real control parameter \( p \) such as Lorenz system, Rössler system, Chen system, Lotka-Volterra system, Rabinovich-Fabrikant system, Hindmarsh-Rose system, Lü system, some classes of minimal networks, and many others. For example, for the Lorenz system

\[
\begin{align*}
\dot{x}_1 &= a(x_2 - x_1), \\
\dot{x}_2 &= x_1(p - x_3) - x_2, \\
\dot{x}_3 &= x_1x_2 - cx_3,
\end{align*}
\]

(10.8)

with \( a = 10, c = 8/3 \) and \( p \) the control parameter,\(^1\)

\[
f(x) = \begin{pmatrix} a(x_2 - x_1) \\ -x_1x_3 - x_2 \\ x_1x_2 - cx_3 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\]

Let \( p_h(t) = p(t) \) for any \( h > 0 \). Then, the “switching” equation (related to the PS algorithm) has the following form:

\[ \dot{x}(t) = f(x(t)) + p_h(t)A x(t), \quad t \in I = [0, T], \quad x(0) = x_0, \]

(10.9)

and the “average” equation, obtained for \( p \) replaced with \( p^* \) given by (10.6), is

\[ \dot{x}(t) = f(\bar{x}(t)) + p^*A \bar{x}(t), \quad t \in I = [0, T], \quad \bar{x}(0) = \bar{x}_0. \]

(10.10)

By applying the PS algorithm, the obtained switched solution of (10.9) will converge to the average solution of (10.10).

To approximate a desired solution, corresponding to some value \( p \), we have to replace \( p^* \) with \( p \) in (10.6) and choose a set \( P_N \) with the underlying weights \( m_i, \ i = 1, 2, ..., N \), such that (10.6) is verified. Next, by applying the PS algorithm with these ingredients, the obtained switched solution will approximate the searched (averaged) solution.

\(^1\)Also, \( a \) and \( c \) can be considered as control parameters to match to the form (10.7).
10.2.1.1 Convergence of the PS Algorithm

The following assumptions are made.

Assumption H1. $f$ satisfies the usual Lipschitz condition:

$$|f(y_1) - f(y_2)| \leq L |y_1 - y_2|, \quad \forall y_{1,2} \in \mathbb{R}^n,$$

(10.11)

for some $L > 0$.

Assumption H2. The initial conditions $x_0$ and $\bar{x}_0$ belong to the same basin of attraction $Y$ of the solution of (10.10).

Under the above assumptions, the convergence of the PS algorithm is given by the theorem

**Theorem 10.1** ([21]) Let $\| \cdot \|_0$ be the maximum norm on $C(I, \mathbb{R}^n)$. Under the above assumptions, it holds that

$$|x(t) - \bar{x}(t)| \leq (|x_0 - \bar{x}_0| + h\|A\|\|\bar{x}\|_0 K) \times e^{(L + \|P\|_0\|A\|)T},$$

(10.12)

for all $t \in [0, T]$, where

$$K := \max_{t \in [0, T]} \left| \int_0^t (P(s) - p^*) ds \right|.$$

**Sketch of the proof:**

From (10.9) and (10.10)

$$|x(t) - \bar{x}(t)| \leq |x_0 - \bar{x}_0| + L \int_0^t |x(s) - \bar{x}(s)| ds + \left| \int_0^t (p_h(s) - p^*) ds \right| \|A\|\|\bar{x}\|_0$$

$$+ \|P\|_0\|A\| \int_0^t |x(s) - \bar{x}(s)| ds = |x_0 - \bar{x}_0| + \|A\|\|\bar{x}\|_0 \left| \int_0^t (p_h(s) - p^*) ds \right|$$

$$+ (L + \|P\|_0\|A\|) \int_0^t |x(s) - \bar{x}(s)| ds$$

$$\leq |x_0 - \bar{x}_0| + h\|A\|\|\bar{x}\|_0 K + (L + \|P\|_0\|A\|) \int_0^t |x(s) - \bar{x}(s)| ds,$$

and by Gronwall inequality [25], one obtains (10.12). \(\square\)

Next, adopt the following reasonable assumption regarding the notion of the (numerical) attractor utilized in this paper, necessary to implement numerically the PS algorithm.

Assumption H3. To every $p$ value, for a given initial condition $x_0$, there corresponds a unique solution and, therefore, a single numerical attractor, denoted by $A_p$, considered as a numerically approximation of its $\omega$-limit set [22], after neglecting a sufficiently long transients.
The following theorem represents the main result concerning the PS algorithm for continuous-time systems.

**Theorem 10.2** Every attractor of the system (10.7) can be numerically approximated by the PS algorithm to arbitrary accuracy.

*Notation:* Denote by \( A^* \) the “synthesized attractor”, obtained with the PS algorithm, and by \( A_{p^*} \) the “averaged attractor”, obtained for \( p \) replaced with \( p^* \) given by (10.6).

To obtain a desired attractor \( A_p \) corresponding to some value \( p \), one has to replace in (10.6) \( p^* \) with \( p \) and choose an adequate set \( \mathcal{P}_N \) with underlying weights \( m_i \), \( i = 1, 2, ..., N \), such that (10.6) is verified. Next, by applying the PS algorithm, the obtained (switched) attractor \( A^* \) will approximate the searched (averaged) attractor \( A_p \).

**Remark 10.1** The relation (10.6) is convex: if one denotes \( \alpha_i = m_i / \sum_{k=1}^{N} m_k \), then \( \sum_{i=1}^{N} \alpha_i = 1 \), and \( p^* = \sum_{i=1}^{N} \alpha_i p_i \). Therefore, the only necessary condition to approximate some attractor \( A_p \) is to choose \( \mathcal{P}_N \) such that \( p \in (p_{\text{min}}, p_{\text{max}}) \), with \( p_{\text{min}} = \min \{ \mathcal{P}_N \} \) and \( p_{\text{max}} = \max \{ \mathcal{P}_N \} \). Moreover, the convexity implies a robustness-like property of the PS algorithm: for every set \( \mathcal{P}_N \), \( A^* \) will be situated “between” the attractors \( A_{p_{\text{min}}} \) and \( A_{p_{\text{max}}} \), with order being induced by the natural order of the real numbers in the parameter set \( \mathcal{P}_N \).

Theorem 10.2 means that by choosing some value \( p \), there always exists an attractor \( A_p \) (Remark 10.1) and a set of \( N > 1 \) parameters \( \mathcal{P}_N \), such that \( p^* = p \in (p_{\text{min}}, p_{\text{max}}) \) with the underlying weights \( m_i \), \( i = 1, 2, ..., N \), and \( p^* \) given by the relation (10.6).

Next, as stated by Theorem 10.2, \( A_{p^*} \) will be approximated by the attractor \( A^* \), generated by the PS algorithm.

### 10.2.1.2 Numerical Implementation of the PS Algorithm

In order to implement numerically the PS algorithm, a numerical method for ODEs is necessary (for example, the standard Runge-Kutta method) with a fixed step size \( h \). For the set \( \mathcal{P}_N \) with weights \( m_i \), \( i = 1, 2, ..., N \), and a fixed step-size \( h \), consider the PS algorithm in the following symbolic scheme:

\[
\begin{bmatrix}
m_1 p_1, m_2 p_2, ..., m_N p_N
\end{bmatrix}
\tag{10.13}
\]

For example, if one wants to apply the PS algorithm on the set \( \mathcal{P}_2 = \{ p_1, p_2 \} \) with weights \( m_1 = 2 \) and \( m_2 = 1 \), i.e. the scheme \( [2p_1, 1p_2] \) applied with step size \( h \), it means to do for the first two steps, \( 2h \), of integration of the underlying IVP, \( p = p_1 \), then for the next single step of size \( h \), \( p = p_2 \), then for the next two
Table 10.1 The pseudocode of the PS algorithm

<table>
<thead>
<tr>
<th>input: (x_0, T, N, h, P_t, m_1, \ldots, m_M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>output: (x)</td>
</tr>
<tr>
<td>(n := T/h), (j := 1)</td>
</tr>
<tr>
<td>while (j &lt; n)</td>
</tr>
<tr>
<td>for (i = 1 : N)</td>
</tr>
<tr>
<td>for (k = 1 : m_i)</td>
</tr>
<tr>
<td>(x_j \leftarrow \text{one step integration with } p = p_i)</td>
</tr>
<tr>
<td>(j = j + 1)</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

Fig. 10.2 Bifurcation diagram for the Lorenz system for \(p \in [90, 101]\)

steps, \(p = p_1\), and so on, until the entire integration interval \(I\) is covered (see the pseudocode in Table 10.1).

Let consider the Lorenz system. The obtained switched and the averaged attractors are overplotted in the phase space and in time series. Also, whenever necessary, the Poincaré section is utilized. The integration time is \(I = [0, 200]\) and \(h = 0.002\). For the stable cycles, the transients were removed.

To have a general view of the parameter space wherefrom we have to peak the \(p\) values, a bifurcation diagram is shown in Fig. 10.2.

1. Next, we present the way in which the PS algorithm can be used to obtain (approximately) stable or chaotic attractors of integer-order systems.
Fig. 10.3 Top Lorenz stable cycle corresponding to $p = 93$, obtained with the scheme $[1p_1, 1p_2]$ with $p_1 = 90$ and $p_2 = 96$ (Parrondo’s chaos control game: $\text{chaos}_1 + \text{chaos}_2 = \text{order}$); a Phase overplot of the attractors $A^*$ and $A_{p^*}$, b-c Underlying chaotic attractors $A_{90}$ and $A_{96}$, d-f Overplot of attractors $A^*$ and $A_{p^*}$ time series. The enlarged view in Figure e reveals the inherently numerical errors; Bottom g Lorenz stable cycle corresponding to $p = 93$, obtained with the scheme $[2p_1, 1p_2, 1p_3, 1p_4]$, with $p_1 = 90.4$, $p_2 = 91$, $p_3 = 95$, $p_4 = 98.2$ (Parrondo’s chaos control game: $\text{chaos}_1 + \text{chaos}_2 + \text{chaos}_3 + \text{chaos}_4 = \text{order}$)

a. Suppose one wants to approximate the attractor corresponding to $p = 93$ (chosen in a periodic window, Fig. 10.2), which is a stable cycle. To do that, one can choose, for example $\mathcal{P}_2 = \{90, 96\}$, whose values belong to different chaotic windows in the parameter space (Fig. 10.3b, c) with weights $m_1 = m_2 = 1$, which when replaced in (10.6) gives the desired (average) value $p^* = (1 \times 90 + 1 \times 96) / 2 = 93$. By applying the PS algorithm with the scheme $[1p_1, 1p_2]$, the obtained switched attractor $A^*$ approximates the
averaged attractor $A_{p^*}$ (Fig. 10.3a). A perfect match is also revealed by the
overplotted time series in Fig. 10.3d–f. Even there exists an apparently per-
fect superposition, in the detail in Fig. 10.3e, one can see a relatively small
difference between the two time series, due to the inherently numerical errors.
Since in this example the attractors, corresponding to $p_1$ and $p_2$, whose dyna-
ics have been switched, are chaotic and the switched attractor is a regular
motion, one can write in Parrondian words:

$$\text{chaos}_1 + \text{chaos}_2 = \text{order},$$

which represents Parrondo’s game applied as a chaos control-like result.

b. The same stable cycle can be obtained e.g. with the scheme $[2p_1, 1p_2, 1p_3,$
$1p_4]$, with $p_1 = 90.4, p_2 = 91, p_3 = 95, p_4 = 98.2$. Again, (10.6) gives
$p^* = 93$ and the switched attractor $A^*$ approximates the averaged attractor
$A_{p^*}$ (Fig. 10.3g). Since the attractors corresponding to $p_i, i = 1, 2, 3, 4,$ are
chaotic (Figs. 10.2 and 10.3h–i), the control-like Parrondo game is $\text{chaos}_1 +$
$\text{chaos}_2 + \text{chaos}_3 + \text{chaos}_4 = \text{order}$.

c. The PS algorithm can be utilized for anticontrol too. For example, using the
scheme $[1p_1, 1p_2]$ with $p_1 = 92$ and $p_2 = 100$ chosen in two periodic orbits
(see Fig. 10.2 and Fig. 10.4a, b), one obtains the chaotic attractor $A^*$ which
approximates the stable attractor $A_{p^*}$ with $p^* = 96$. Because one should use
an infinity time to generate the chaotic attractors, the inherently finite-time
approximation is less accurate than that for chaos control, as can be seen in
Fig. 10.4 c. However, the shapes of $A^*$ and $A_{p^*}$ look similar, as indicated also
by the Poincaré section with the plane $x_3 = 100$ (Fig. 10.4d). In Parrondian
words, the anticontrol result can be written $\text{order}_1 + \text{order}_2 = \text{chaos}$.

2. The PS algorithm applies also to fractional-order systems\textsuperscript{2}

Consider the Chen system of fractional-order [13, 29] in the following form:

\begin{align}
D_{x_1}^{0.92} x_1 &= p(x_2 - x_1), \\
D_{x_2}^{0.95} x_2 &= (3.65 - p)x_1 + 3.65x_2 - x_1x_3, \\
D_{x_3}^{0.90} x_3 &= x_1x_2 - 0.3x_3,
\end{align}

(10.14)

where $D^{\alpha}_x$ denotes the the Caputo differential operator of order $q$ (see e.g. [12, 33,
37]). The numerical method used here to integrate the system is the Grünwald-
Letnikov method for fractional differential equations (see e.g. [5, 30, 41]). For
$p_1 = 4.243$ and $p_2 = 4.302$, the system behaves chaotically (Fig. 10.5a, b) and
with the scheme $[3p_1, 1p_2]$ one obtains $p^* = 4.25775$ for which the system is
stable. By applying the PS algorithm, the switched attractor $A^*$ matches perfectly

\textsuperscript{2}There exists no convergence result so far. However, intensively numerical tests reveal, like in the
considered example, a good match between the switched attractor and the averaged attractor in the
case of fractional-order systems.
Fig. 10.4 Lorenz chaotic attractor corresponding to $p = 36$, obtained with the scheme $[1p_1, 1p_2]$ with $p_1 = 92$ and $p_2 = 100$ (Parondo's anticontrol game: order$_1 +$ order$_2 = chaos$). \textbf{a, b} Underlying stable cycles $A_{92}$ and $A_{100}$; \textbf{c} Phase overplot of the attractors $A^*$ and $A_{p^*}$; \textbf{d} Poincaré section with $x_3 = 100$ through the overplot attractors $A^*$ and $A_{p^*}$.

the averaged attractor $A_{p^*}$ (Fig. 10.5c), and this chaos control-like Parondo game reads $chaos_1 + chaos_2 = order$.

In the above examples, the scheme (10.13) is implemented periodically: the values of $p$ take successively the values $p_1$ for $m_1$ times, then $p_2$ for $m_2$ times, and so on until $p_N$ for $m_N$ times, after which it repeats. However, the order of $p_i$ with the underlying weight $m_i$ can be taken randomly by using, for example, some random uniformly distributed sequence$^3$ of values $p_i$. The averaged value, denoted $\overline{p}^*$, has to be determined now by the following relation:

$$\overline{p}^* = \frac{\sum_{i=1}^{N} m_i' p_i}{\sum_{i=1}^{N} m_i'}, \quad p_i \in \mathcal{P}_N,$$

(10.15)

$^3$E.g. the pseudorandom function, found in all dedicated software.
where, $m$ are the total number of switchings of $p_1$ when the integration ends. After a sufficiently large integration interval $I$, $\overline{p} \approx p^*$. However, in this case, for the considered example, supplementary precautions should be considered, such as the dispersal of $p$ values in the parameter space, which have to be close to $p^*$. Also, the integration interval has to be larger and the step-size $h$ smaller.

Consider the Chen system of integer order in the following form

$$
\begin{align*}
\dot{x}_1 &= 35(x_2 - x_1), \\
\dot{x}_2 &= (p - 35)x_1 - x_2x_3 + px_2, \\
\dot{x}_3 &= x_1x_2 - 3x_3,
\end{align*}
$$

and suppose one wants to obtain the stable cycle corresponding to $p = 26.08$ [16] by the scheme $[1p_1, 1p_2]$ with $p_1 = 26$ and $26.16$, generating chaotic attractors (Fig. 10.5d, e). With the step size $h = 0.0005$ and the integration interval
I = [0, 800], the PS algorithm approximates the stable cycle (Fig. 10.5f). Now, the relatively small differences between the two attractors are cleared.

10.2.2 PS Algorithm Applied to Maps

As proved analytically in Sect. 10.2.1.1 by applying Parrondo’s game to continuous-time systems, the switched solution obtained with the PS algorithm converges to the averaged solution. However, for the discrete systems, things are different.

Consider the following discrete variant of (10.7):

\[ x_{k+1} = f(x_k) + q_k A x_k, \]

where \( x_0 \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^n \) satisfies the Lipschitz condition, and \( \{q_k\}_{k \in \mathbb{N}}, q_k = p, \) for \( k \in [M_i - 1, M_i], M_0 = 0, M_i := \sum_{k=1}^i m_j, \) \( p_i \in \mathcal{P}_N, 1 \leq i \leq N, \) and \( T := M_N, \) is a \( T \)-periodic piecewise constant sequence. Then, there is no any relationship with the average equation

\[ x_{k+1} = f(x_k) + p^\ast A x_k, \]

where \( p^\ast \) is given by (10.6), such as for the case of continuous-time systems.

However, for the following discrete version of the PS algorithm:

\[ x_{k+1} = x_k + h(f(x_k) + q_k A x_k), \]

there exists an averaged form:

\[ \bar{x}_{k+1} = \bar{x}_k + h(f(\bar{x}_k) + p^\ast A \bar{x}_k), \]

where the averaging theory applies [14].

Even Theorem 10.1 does not apply to the most-known discrete-time systems (like the logistic map), or to complex systems (like fractals), but the PS algorithm still works as chaos-control like and anticontrol-like tasks, for which quite intriguing results as can be seen as in the next section.

10.2.2.1 PS Algorithm Applied to the Logistic Map

Apply the PS algorithm to the logistic map \( f : [0, 1] \to [0, 1], f(x) = px(1 - x), \)

\( p \in [0, 4], \) in the following simplest form:

\[ x_{k+1} = q_k x_k (1 - x_k), \quad k = 0, 1, \ldots \]  \hspace{1cm} (10.17)

\[ ^4 \text{In} \ [3, 23], \text{some particular forms of switches are used to study the behavior of alternated orbits for the more accessible quadratic (Mandelbrot) map } x_{k+1} = x_k^2 + p. \]
Fig. 10.6 Stable cycle obtained using the scheme \([5p_1, 1p_2, 3p_3, 1p_4, 4p_5]\) with \(\mathcal{D}_5 = \{3, 3.9, 3.61, 2.61, 3.4\}\) applied to the logistic map (Parrondo’s chaos-control game: \(\text{chaos}_1 + \text{chaos}_2 + \text{chaos}_3 + \text{chaos}_4 + \text{order}_1 = \text{order}\)). a Orbits of the underlying dynamics corresponding to \(p_i, i = 1, 2, \ldots, 5\). b Cobweb indicating the multiple periods of the stable cycle; c Time series of the controlled orbit; d First return map.

To analyze the numerical results, one can use time series, cobweb and first-return map, have been utilized with \(q_k\) defined as above: \(q_k = p_i\) for \(k \in [M_i - 1 + 1, M_i]\), \(1 \leq i \leq N\).

This time, with the PS algorithm one can obtain stable orbits which are different from those of the logistic map \([14]\). Therefore, the PS algorithm can be used to control chaos or obtain chaotization. By choosing empirically the weights \(m_i\) and \(\mathcal{D}_N\), it is possible to control the chaotic behavior of the logistic map. As verified numerically in \([14]\), there exists a positive probability to realize chaos control by using (the generalized) Parrondo’s game.

1. For example, choosing the scheme \([5p_1, 1p_2, 3p_3, 1p_4, 4p_5]\) with \(\mathcal{D}_5 = \{3, 3.9, 3.61, 2.61, 3.4\}\), one obtains the following Parrondo’s game for chaos control (Fig. 10.6): \(\text{order}_1 + \text{chaos}_2 + \text{chaos}_3 + \text{order}_2 + \text{order}_3 = \text{order}\). The dynamics corresponding to \(p_i, i = 1, 2, 3, 4, 5\), are plotted in Fig. 10.6 a. In this case, \(\text{order}\) represents a stable orbit, different from but similar to any of the possible orbits.
Fig. 10.7 Other chaos control and anticontrol of the logistic map; a Chaos control with \([1p_1, 1p_2, 1p_3, 1p_4, 1p_5, 1p_6]\) with \(p_1 = 3.6, p_2 = 3.7, p_3 = 3.75, p_4 = 3.8, p_5 = 3.86\) and \(p_6 = 0.9\) (Parrondo's chaos control game: \(\text{order}_1 + \text{order}_2 + \text{order}_3 + \text{order}_4 + \text{chaos}_1 + \text{chaos}_2 = \text{order}\)); b Chaos control with \([1p_1, 3p_2, 2p_3, 1p_4, 2p_5, 2p_6]\) with \(p_1 = 2.6, p_2 = 2.9, p_3 = 3.1, p_4 = 3.4, p_5 = 3.7, p_6 = 4\) (Parrondo's chaos control game: \(\text{order}_1 + \text{order}_2 + \text{order}_3 + \text{order}_4 + \text{order}_5 + \text{order}_6 = \text{order}\)); c Chaos control with \([10p_1, 1p_2, 5p_3, 10p_4, 1p_5, 1op_6]\) with \(p_1 = 3.4, p_2 = 2.35, p_3 = 3.5, p_4 = 2.8, p_5 = 1.9, p_6 = 0.85\) (Parrondo's chaos control game: \(\text{order}_1 + \text{order}_2 + \text{order}_3 + \text{order}_4 + \text{order}_5 + \text{order}_6 = \text{order}\)); d Anticontrol with \([1p_1, 1p_2]\) with \(p_1 = 3.738\) and \(p_2 = 3.84\) (Parrondo's anticontrol game: \(\text{chaos}_1 + \text{order}_1 = \text{chaos}_2\)).

of the logistic map, revealed by the cobweb, time series and first return map (Fig. 10.6, c and d, respectively).

2. By using the scheme \([1p_1, 1p_2, 1p_3, 1p_4, 1p_5, 1p_6]\) with \(p_1 = 3.6, p_2 = 3.7, p_3 = 3.75, p_4 = 3.8, p_5 = 3.86\) and \(p_6 = 0.9\), one obtains the stable orbit plotted in Fig. 10.7 a. In this case, chaos control is implemented by Parrondo's game: \(\text{chaos}_1 + \text{chaos}_2 + \text{chaos}_3 + \text{chaos}_4 + \text{order}_1 = \text{order}\).

3. The stable orbit plotted in Fig. 10.7 b is obtained by the scheme \([1p_1, 3p_2, 2p_3, 1p_4, 2p_5, 2p_6]\) with \(p_1 = 2.6, p_2 = 2.9, p_3 = 3.1, p_4 = 3.4, p_5 = 3.7, p_6 = 4\). In this case, the Parrondo game has the following form: \(\text{order}_1 + \text{order}_2 + \text{order}_3 + \text{order}_4 + \text{chaos}_1 + \text{chaos}_2 = \text{order}\).
4. The periodic bursts in Fig. 10.7 c [14] are obtained by the scheme $[10p_1, 1p_2, 5p_3, 10p_4, 1p_5, 10p_6]$, with $p_1 = 3.4$, $p_2 = 2.35$, $p_3 = 3.5$, $p_4 = 2.8$, $p_5 = 1.9$, $p_6 = 0.85$ and Parrondo’s game is: $order_1 + order_2 + order_3 + order_4 + order_5 + order_6 = order$.

5. If one uses the scheme $[1p_1, 1p_2]$ with $p_1 = 3.738$ and $p_2 = 3.84$, the PS algorithm simulates the Parrondo game to model the anticontrol of chaos: $chaos_1 + order_1 = chaos_2$ (Fig. 10.7 d).

10.2.2.2 PS Algorithm Applied to Fractals

In [17] the PS algorithm is used to alternate two different dynamics of the quadratic complex map $z_{n+1} = z_n^2 + c_i$ to prove that the obtained sets, called alternated Julia sets, can be connected, disconnected, or totally disconnected verifying the Fatou-Julia theorem [20, 27] in the case of polynomials of degree greater than two.

Because in this case one deals with a set of two values, $c_1$ and $c_2$, one operates with “alternations”, not switchings.

As is known, for a complex polynomial $P : \mathbb{C} \to \mathbb{C}$ of degree $d \geq 2$, infinity is a superattracting fixed point. If one denotes by $\mathcal{A}(\infty)$ the attraction basin of $\infty$ for the polynomial $P$, $A(\infty) = \{ z \in \mathbb{C} | P^n \to \infty \}$, then the filled Julia set of $P$ is the set $K = \mathbb{C} \setminus \mathcal{A}(\infty)$. The boundary of the filled Julia set is called the Julia set, where chaotic dynamics occur.

The connectivity properties of the Julia set are in a relationship with the dynamical properties about its finite critical points (Fatou-Julia Theorem [20, 27]): The Julia set is connected if and only if all the critical orbits are bounded; and the set is totally disconnected, a Cantor set, if (but not only if) all the critical orbits are unbounded. In [9, 38], the theorem was completed as follows: For a polynomial with at least one critical orbit unbounded, the Julia set is totally disconnected if and only if all the bounded critical orbits are aperiodic.

The alternated Julia sets $K_{c_1,c_2}$ are the set of points in the complex plane with bounded orbits when one iterates the alternated system

$$
P_{c_1,c_2} : \ z_{n+1} = \begin{cases} 
   z_n^2 + c_1, & n \text{ even}, \\
   z_n^2 + c_2, & n \text{ odd}.
\end{cases}
$$

The generated orbit is

$$
z_0, \\
z_1 = z_0^2 + c_1, \\
z_2 = (z_0^2 + c_1)^2 + c_2, \\
z_3 = ((z_0^2 + c_1)^2 + c_2)^2 + c_1, \\
...$$
In a similar way one can be define the alternated filled Julia set \( K_{c_2,c_1} \), which has the same shape with the alternated filled Julia set \( K_{c_1,c_2} \), as being the set of points in the complex plane with bounded orbits when one iterates the alternated system

\[
P_{c_2,c_1}: z_{n+1} = \begin{cases} 
z_n^2 + c_2, & n \text{ even}, \\
z_n^2 + c_1, & n \text{ odd}. 
\end{cases}
\]

In [17], it is proved that the alternated Julia sets verify the Fatou-Julia theorem in the case of complex polynomials of degree greater than two.

As known, the Julia set is totally disconnected if \( c \) does not belong to the Mandelbrot set [7, 8, 32]. However, in [17] it is proved that the alternated Julia sets can be connected, disconnected or totally disconnected. Because the totally disconnected sets, disconnectedness sets and connected form a four-dimensional body (it depends on four real variables: \( Re(c_{1,2}) \) and \( Im(c_{1,2}) \)), to study computationally the connectivity problem, one has to fix some of these variables, and scroll the others within some domain. In other words, to obtain a three-dimensional views (of the four existing objects), one has to slice the four-dimensional body with one of the four planes \( Re(c_{1,2}) = ct, Im(c_{1,2}) = ct \) (see the volume rendering [19, 31] in Fig. 10.8, where a three-dimensional view is obtained by sectioning the body with the plane \( Im(c_1) = 1 \)). To obtain two dimensional views, two planes sections (slices) are necessary.

**Fig. 10.8** Three-dimensional view of the connectivity body of the alternated Julia sets, obtained with the section with \( Im(c_1) = 1 \). The white region (body's exterior) indicates the points for which the alternated Julia sets are totally disconnected, the blue regions indicate the disconnectedness while the red regions the connectedness.

For example, if one considers the planar section with \( c_2 = -0.1562 + 1.0320i \) and \( c_1 \in [-0.176, -0.136] \times [1.012, 1.052] \) (Fig. 10.9 a), the filled Julia set correspond-
Fig. 10.9  a Section through the three-dimensional body, obtained by alternating Julia sets with $c_2 = -0.1562 + 1.0320i$ and $c_1 \in [-0.176, -0.136] \times [1.012, 1.052]$; b Totally disconnected filled Julia set corresponding to $c = -0.1562 + 1.0320i$; c Connected alternated filled Julia set corresponding to the point $A$; d Disconnected alternated Julia set corresponding to the point $B$; e Totally disconnected alternated Julia set corresponding to the point $C$.

ing to $c_2 = -0.1562 + 1.0320i$ is a totally disconnected set (Fig. 10.9 b), while the alternated Julia sets for $c_2 = -0.1562 + 1.0320i$ and $c_1$ considered in the connected region (point $A$) is a connected set (Fig. 10.9 c). For $c_2 = -0.1562 + 1.0320i$ and $c_1$ considered in the disconnected region (point $B$) is a disconnected set (Fig. 10.9 d), and for $c_2 = -0.1562 + 1.0320i$ and $c_1$ considered in the white region (point $C$) is a totally connected set (Fig. 10.9 e).

Remark 10.2  Representing graphically the three-dimensional connectivity bodies, a remarkable property was revealed in [17]: as known, the Mandelbrot set is the set of all $c$ values for which each (classical) Julia set is connected. However, the "ends" of the three-dimensional body shown in Fig. 10.8, indicate a new and intriguing property: it is the set of all parameter values, for which each corresponding alternated Julia set is disconnected form Mandelbrot sets (the blue points in Fig. 10.9 a). By using special algorithms to draw fractals, one can prove that the apparently separated parts (dots) of connectivity and disconnectivity are in reality connected to their body [35, Chap. 4].

10.3 Conclusion

In this chapter, we have presented the approach of a generalization of Parrondo's game, implemented for both continuous-time and discrete-time systems, via the PS algorithm. Thus, by applying the PS algorithm, the forms of Parrondo's para-
dox game read $\text{chaos}_1 + \text{chaos}_2 + \ldots + \text{chaos}_N = \text{order}$, for $N \geq 2$, or $\text{order}_1 + \text{order}_2 + \ldots + \text{order}_N = \text{chaos}$, acting like chaos-control like or anticontrol-like behaviors. Also, combinations of ordered and chaotic motions can lead to chaos-control like and anticontrol-like results. These generalizations of Parrondo’s game, applied as chaos control or anticontrol schemes have been used here to Lorenz system, Chen systems of integer and fractional order, the logistic map, and also fractals (alternated Julia sets). While for the continuous-time systems, the convergence of the PS algorithm has been proved analytically, but for the fractional-order systems, the convergence has been verified only numerically. Also, for the logistic map, the PS algorithm generates different orbits from the existing orbits, Parrondo’s paradox has been implemented to realize chaos control and anticontrol. The apparently paradoxical result obtained with the PS algorithm applied to continuous systems, resides in the linearity dependence on the parameter $p$ in the underlying IVP. Although this particularity seems to be restrictive, it characterizes most-known continuous systems. One of the most interesting new property, revealed by the PS algorithm, is the fact that the Mandelbrot set seems to be not only the set of all complex points for which the Julia sets are connected, but also the set of all complex points for which the alternated Julia sets are disconnected. With the PS algorithm, every attractor of a considered system can be generated (approximated), but due to some objective reasons, one cannot set some parameter values. The PS algorithm can be used as a possible explanation of the strange dynamics of some systems where switchings between the underlying dynamics occur, either periodically or randomly.

References