Seeking computational efficiency boundaries: the Păun's conjecture

David Orellana-Martín[1] · **Agustín Riscos-Núñez**[1]

Abstract

In 2005, Gh. Păun raised an interesting question concerning the role of electrical charges in P systems with active membranes from a complexity point of view. Specifically, he formulated a question about the computational efficiency of polarization-less P systems with dissolution rules and division rules only for elementary membranes. Several approaches have been carried out, and some partial answers have been given. This is probably the most important open problem in computational complexity theory in the framework of Membrane Computing. The study of the efficiency of membrane systems has been a very fruitful area, providing not only the above-stated partial answers, but also several frontiers of efficiency to tackle the **P** vs **NP** problem. In this work, a survey on classical and current results on complexity aspects is given, emphasizing on the frontiers of efficiency and the ingredients taken into account for each of them.

Keywords Păun's conjecture · Membrane computing · Computational complexity theory · P systems

Mathematics Subject Classification MSC 68Q10 · MSC 68Q15

1 Introduction

Given an abstract problem, a solution that can be structured through a sequence of computational tasks whose execution provides the correct answer is called a *mechanical solution* or *algorithmic solution* of the problem. *Computability theory* aims to define the concept of mechanical solvability in a mathematical context (providing *computing models*) allowing to classify abstract problems according to whether they are mechanically solvable (*decidable*) or not (*undecidable*). Nevertheless, it is interesting to distinguish between *solvability in principle*, with which computability theory deals, and *solvability in practice*, which is a matter of obtaining a mechanical procedure that can be executed in a computational device using space and time resources likely to be available. *Computational complexity theory* deals with the *solvability in practice* of abstract problems; that is, by analyzing the amount of resources needed to execute a mechanical solution to a problem. This theory aims to classify abstract problems according to whether they are algorithmically solvable in a *reasonable* way (*tractable*) or not (*intractable*). The term *reasonable algorithm* or *feasible algorithm* refers to a mechanical procedure requiring computational resources (time or space) bounded above by a polynomial function on the size of its inputs in a universal computing model (usually, Deterministic Turing Machines).

A *computing paradigm* is a mathematical theory that allows to describe formal mechanical procedures, according to certain syntactic and semantic requirements of the theory. A *computing model* consists of a mathematical definition of the concept of mechanical procedure in such a way that it captures the intuitive and informal idea of it; that is, a mechanical procedure is a specific case of a computing model. From there, it is naturally defined what solving an abstract problem in the computer model through mechanical procedures, means. Therefore, computing paradigms consist of a formal framework providing computing models. Following [11], a computing model is said to be *efficient* (respectively, *presumably efficient*) if it has the ability to provide polynomial-time solutions for intractable problems (resp., **NP**-complete problems). The term *presumably*

✉ David Orellana-Martín
dorellana@us.es

Agustín Riscos-Núñez
ariscosn@us.es

[1] Research Group on Natural Computing Dept. of Computer Science and Artificial Intelligence, Universidad de Sevilla E.T.S.I. Informática, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

efficient refers to the fact that, as generally believed, if $\mathbf{P} \neq \mathbf{NP}$ then each **NP**-complete problem is an intractable one; consequently, under this hypothesis, any presumably efficient computing model would be efficient.

Usually, computational complexity theory in Membrane Computing deals with *decision problems*; that is, problems that require a "`yes`" or "`no`" answer. Formally, a *decision problem*, X, is an ordered pair (I_X, θ_X) such that I_X is a language over a finite alphabet whose elements are called *instances* of the problem, and θ_X is a total Boolean function over I_X. The solvability of decision problems in a computing model is defined through the recognition of the languages associated with them (the set of instances whose answer is "`yes`") by means of mechanical procedures.

Membrane computing is a computing paradigm inspired from the structure and the functioning of living cells, as well as from the organization of cells in tissues, organs, and other higher-order structures. The computing models in this paradigm are generically called *membrane systems* or *P systems*. The main syntactical ingredients of a membrane system are: (a) a finite alphabet, the *working* alphabet, whose elements, called *objects*, are abstractions of chemical substances; (b) a finite set of processor units delimiting *compartments*, called *membranes*, *cells* or *neurons*, interconnected by a hierarchical or a graph structure, in such a manner that initially each compartment contains a multiset of objects; (c) a finite set of *rewriting rules*, abstractions of chemical reactions, that provide the dynamics of the system; and (d) an *environment* playing different roles depending on the system you are working with. According with the type of structure underlying the systems and the semantics, there are basically three types of membrane systems: *cell-like* membrane systems where the compartments are arranged in a labelled rooted tree, explicitely given, like in a living cell [28]; *tissue-like* membrane systems with a directed graph structure associated, implicitly given from the set of rules of the system, inspired from the living tissues where cells bump into each other and communicate through pores or other membrane mechanisms [26, 31, 32]; and *neural-like* membrane systems with a directed graph structure associated, explicitely given, aiming to abstract the way the neurons communicate with each other processing short electrical impulses, in a complex network established by synapses [7].

It is worth noting that the environment plays a singular role in some membrane systems where the communication rules are of the type symport/antiport, observing the *conservation law*, in the sense that they compute by changing the places of objects with respect to the membranes, and not by changing the objects themselves. Unlike in other membrane systems, in these, the environment plays an active role in the sense that the system not only sends objects to the environment, but also brings objects from the environment. In these membrane systems, there exists a special alphabet, associated with the environment, whose elements appear at the initial configuration of the system in an arbitrary large number of copies.

The paper is organized as follows: in Sect. 2 the definition of solving decision problems by means of membrane systems is recalled. Section 3, some results on cell-like membrane systems are recalled, as well as their relation with the Păun's conjecture. In Sect. 4, results concerning complexity of tissue P systems are recalled. The paper ends with some conclusions of the work and interesting open research lines in the field of computational complexity theory in Membrane Computing.

2 Solving decision problems by means of membrane systems

To solve decision problems in the framework of Membrane Computing, language recognition devices must be defined. In this context, the concept of *recognizer membrane system* has been introduced in [38].

A membrane system is said to be a *recognizer system* if it has the following syntactic and semantic peculiarities: (a) the working alphabet has two distinguished objects (`yes` and `no`); (b) there exist an input alphabet strictly contained in the working alphabet and an *input compartment*; (c) the initial content of each compartment is a multiset of objects from the working alphabet not belonging to the input alphabet; (d) all the computations of the system halt; and (e) for each computation either object `yes` or object `no` (but not both) must have been released to the environment and only at its last step. Recognizer membrane systems have the ability to accept or reject multisets over the input alphabet. Specifically, given a recognizer membrane system Π, for each multiset m over the input alphabet, a new initial configuration is obtained by adding the multiset m to the content of the input compartment at the initial configuration of Π (the system Π with this new initial configuration associated with m is denoted by $\Pi + m$). Then, we say that system Π *accepts* (respectively, *rejects*) the input multiset m if and only if all the computations of the system $\Pi + m$ answer `yes` (resp. `no`). That is, the system $\Pi + m$ must be confluent (i.e., all the possible computations starting from the initial configuration yield to the same result). Let us recall that, while deterministic systems are always confluent, the notion of confluence is of particular interest in the case of nondeterministic systems, since they are not always confluent.

Following [39], we say that a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$, being n the size of the instance, of recognizer membrane systems solves a decision problem X in *polynomial time and uniform way* [1] if the family $\mathbf{\Pi}$ can be generated by a deterministic Turing machine working in polynomial time, and there exists a pair (cod, s) of polynomial-time computable functions with regards to n (over the set of instances of X) such that the family $\mathbf{\Pi}$ is *polynomially bounded*, *sound* and *complete* with regard to (X, cod, s) (see [39] for details). The function s is such that $s^{-1}(n)$ is a finite set of instances of X and cod is the encoding function such that for each instance u of the problem X for which $s(u) = n$, it returns a multiset of objects that will be the input of the P system $\Pi(n)$. Given a computing model $\mathcal{R}$ of recognizer membrane systems, $\mathbf{PMC}_{\mathcal{R}}$ denotes the set of decision problems solvable by families from $\mathcal{R}$ in polynomial time and uniform way. The class $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and under polynomial-time reduction [39]. Thus, if X is a complete problem for a complexity class $\mathcal{K}$ and $X \in \mathbf{PMC}_{\mathcal{R}}$ then we deduce that $\mathcal{K} \cup \mathbf{co}\text{-}\mathcal{K} \subseteq \mathbf{PMC}_{\mathcal{R}}$.

The computing model $\mathcal{T}$ of recognizer *basic transition P systems* consists of a set of cell-like membrane systems whose membrane structure does not grow, that is, there are no rules producing new membranes in the system. It is well known that by using families of these membrane systems only problems from class $\mathbf{P}$ of computationally tractable problems, can be solved in a uniform way and polynomial time (see [6] for details). Therefore, the computing model $\mathcal{T}$ of recognizer basic transition P systems is non-efficient, that is, $\mathbf{PMC}_{\mathcal{T}} = \mathbf{P}$. Consequently, assuming that $\mathbf{P} \neq \mathbf{NP}$, the ability of a membrane system to construct an exponential workspace (in terms of the number of objects) in polynomial time is not enough to provide polynomial-time and uniform solutions to $\mathbf{NP}$–complete problems. This ability of creating 2^n objects can be reached by means of evolution rules of the type $[\, a_i \to a_{i+1}^2 \,]_h$, where i will go from 0 to $n - 1$, and it will create two objects a_{i+1} from a single object a_i at each step.

In membrane computing, there are, basically, two mechanisms to produce an exponential workspace (in terms of the number of objects and compartments) in polynomial time. These mechanisms are inspired by two relevant processes: *mitosis* and *membrane fission*. The first one is a process of nuclear division in eukaryotic cells during which one cell gives place to two genetically identical children cells. *Membrane fission* occurs when a membrane gives place to two separated membranes, that is, whenever a vesicle is produced or a larger subcellular compartment is divided into smaller discrete units. Specifically, inspired by the mitosis process, *membrane division* rules were defined in the framework of cell-like P systems providing computing devices called *cell-like P systems with active membranes* [29]. These rules are defined as follows: $[\, a \,]_h \to [\, b \,]_h [\, c \,]_h$. For this rule to apply, an object a must be present in a membrane labelled by h, and through the application of the rule, the membrane is divided into two new ones with the same contents than the original, but in the first one, changing the object a by an object b and in the second one, changing the object a by an object c. With respect to the membrane fission process, *cell-like P systems with membrane separation* were introduced in [20]. The definition is similar to division rules: $[\, a \,]_h \to [\, \Gamma_0 \,]_h [\, \Gamma_1 \,]_h$. The applicability requisite is the same as in division rules, and it creates two new membranes; but objects present in the membrane that belong to Γ_0 will go to the first new membrane while objects that belong to Γ_1 to the second new membrane, being Γ_0 and Γ_1 predefined, where $\Gamma_0 \cap \Gamma_1 = \emptyset$ and $\Gamma_0 \cup \Gamma_1 = \Gamma$. These concepts were also considered in the framework of tissue-like P systems: *tissue P systems with cell division* [34] and *tissue P systems with cell separation* [21].

3 Recognizer cell-like membrane systems

In this section, the capability of the computing model of recognizer membrane systems working in a cell-like manner, is recalled from a complexity point of view.

3.1 P systems with active membranes

The computing model of P systems with active membranes was introduced in [29]. These membrane systems have some important features: (a) they use three electrical charges; (b) the polarization of a membrane but not the label, can be modified by the application of a rule; (c) they use division rules as a mechanism to produce an exponential workspace, expressed in terms of the number of membranes and objects, in polynomial time. Consequently, unlike basic transition P systems, the membrane structure of these systems can grow in an exponential way. This fact becomes relevant in order to provide polynomial time and uniform solutions to computationally hard problems by means of families of such kind of recognizer P systems, even using only division for elementary membranes and without using dissolution rules (e.g. `Subset Sum` [36], `Knapsack` [35], `Partition` [4], etc.). The computing model of recognizer P systems with active membranes using division rules (respectively, which do not make use of division rules) is denoted by $\mathcal{AM}$ (resp., $\mathcal{NAM}$). Similarly, $\mathcal{AM}(\alpha, \beta)$, where $\alpha \in \{-d, +d\}$, $\beta \in \{-ne, +ne\}$, denotes the computing model of recognizer P systems with active membranes such that: (a) if $\alpha = +d$ (resp., $\alpha = -d$) then dissolution rules are permitted (resp., forbidden); and (b) if $\beta = +ne$ (resp., $\beta = -ne$) then division

[1] Terms uniform and semi-uniform are used similarly to how they are used in circuit complexity [50].

Table 1 Frontiers of efficiency in P systems with active membranes

Non-efficient	Presumably efficient	Type of frontier
$\mathbf{PMC}_{\mathcal{NAM}}$	$\mathbf{PMC}_{\mathcal{AM}(-d,-ne)}$	Division rules

rules for non–elementary membranes are permitted (resp., forbidden).

From the proof of the Milano theorem [51] (*each deterministic P system with active membranes but without membrane division can be simulated by a deterministic Turing machine with a polynomial slowdown*) and from a proof given by A.E. Porreca [44] (*each tractable problem can be solved in polynomial time by families of recognizer P systems with active membranes and without input*), we have $\mathbf{PMC}_{\mathcal{NAM}} = \mathbf{P}$. Therefore, assuming that $\mathbf{P} \neq \mathbf{NP}$, the computing model $\mathcal{NAM}$ is non-efficient, so in the framework of recognizer P systems with active membranes and electrical charges, division rules (only for elementary membranes) provide a borderline between tractability and intractability, that can be observed in Table 1. Nevertheless, P systems with active membranes seem to be too powerful from a complexity point of view. In fact, all decision problems from the class **PP**, for which there is a non-deterministic Turing machine, M, accepting an instance u if and only if more than half computations of $M(u)$ are accepting ones, is solvable in polynomial time and uniform way by a family from $\mathcal{AM}(-d,-ne)$ [42]. If dissolution rules are allowed then the time complexity class $\mathbf{PMC}_{\mathcal{AM}(+d,-ne)}$ equals to the complexity class of all decision problems recognizable in polynomial time by deterministic Turing machines with oracles for #**P** (see [23] for details), that is, $\mathbf{PMC}_{\mathcal{AM}(+d,-ne)} = \mathbf{P}^{\#\mathbf{P}}$ [8, 45]. As it is known, $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{P}^{\#\mathbf{P}}$, therefore, a frontier of efficiency is found in the use of division rules in P systems with active membranes. If division for non-elementary membranes is also allowed then the complexity class $\mathbf{PMC}_{\mathcal{AM}(+d,+ne)}$ characterizes **PSPACE**, that is, $\mathbf{PSPACE} = \mathbf{PMC}_{\mathcal{AM}(+d,+ne)}$ [46]. Therefore, it would be interesting to remove some ingredients from P systems with active membranes in order to obtain new frontiers of the efficiency. Each of such boundaries will provide tools that hopefully could lead to the separation of well known complexity classes, thus solving long-standing open problems.

3.2 Polarizationless P systems with active membranes

First, we analyze the possible substitution of electrical charges for catalysts, resembling their behavior in chemical reactions, acting as "enablers", in the sense that they provoke the reaction but do not change, and the reduction of the number of possible polarizations from three to two.

In this context, two interesting cases were considered by replacing electrical charges, first with the so called *bi-stable catalysts* [41]; and second, with only two polarizations [1]. Note that, in contrast to [25], the bi-stable catalysts are not always flip-floping from non-barred to barred versions and back, but also rules of the form $ca \rightarrow cw$ and $\bar{c}a \rightarrow \bar{c}w$ are allowed, where a is a single object, and w is a multiset. On one hand, the computing model of recognizer polarizationless P systems with active membranes without dissolution rules but allowing division rules only for non-elementary membranes and also allowing the use of bi-stable catalysts, are presumably efficient (see [41] for details). On the other hand, if only two electrical charges are considered then computational hard problems can be solved in polynomial time and in a uniform way by families of these systems [1]. Let these classes of P systems be denoted by $\mathcal{AM}^c$ and $\mathcal{AM}^+$, respectively.

Therefore, it seems like it might be interesting to analyze the computational efficiency of P systems with active membranes in the case that only one (or equivalently, without) electrical charge is considered. In this context, at the beginning of 2005, Gh. Păun wrote: *"My favorite question (related to complexity aspects in P systems with active membranes and with electrical charges) is that about the number of polarizations. Can the polarizations be completely avoided? The feeling is that this is not possible—and such a result would be rather sound: passing from no polarization to two polarizations amounts to passing from non-efficiency to efficiency"* (problem **F** from [30]). This is the so-called *Păun's conjecture* that can be expressed as follows: the computing model of polarizationless P systems with active membranes which make use of division rules only for elementary membranes, is non-efficient. In some sense, the Păun's conjecture attempts to provide a boundary between the tractability and the presumed intractability of problems. This is of a great relevance since each of these boundaries will, in turn, provide a tool for attacking the **P** *versus* **NP** problem, which is undoubtedly one of the most important problems in Computer Science. In this paper, we emphasize the duality between the formulation of the Păun's conjecture and the search for the boundaries mentioned above.

In what follows, the following notations will be used: $\mathcal{DAM}^0(\alpha,\beta)$ (resp., $\mathcal{SAM}^0(\alpha,\beta)$) denotes the computing model of recognizer polarizationless P systems with active membranes and membrane division (resp. membrane separation) rules. We recall that: (a) if $\alpha = +d$ (resp., $\alpha = -d$) then dissolution rules are permitted (resp., forbidden); and (b) if $\beta = +ne$ (resp., $\beta = -ne$) then division rules for non-elementary membranes are permitted (resp., forbidden). In this context, the Păun's conjecture can be formalized in terms of time complexity classes of recognizer membrane systems, as follows: $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(+d,-ne)}$. Thus, an *affirmative answer* to the conjecture would indicate that the ability to create

Table 2 Frontiers of efficiency in P systems with active membranes with less than three polarizations

Non-efficient	Presumably efficient	Type of frontier
$\mathbf{PMC}_{\mathcal{DAM}^0(-d,-ne)}$	$\mathbf{PMC}_{\mathcal{AM}^c(-d,-ne)}$	Catalysts
$\mathbf{PMC}_{\mathcal{DAM}^0(-d,-ne)}$	$\mathbf{PMC}_{\mathcal{AM}^+(-d,-ne)}$	Number of polarizations
$\mathbf{PMC}_{\mathcal{DAM}^0(-d,+ne)}$	$\mathbf{PMC}_{\mathcal{DAM}^0(+d,+ne)}$	Dissolution rules

an exponential amount of workspace (expressed in terms of the number of membranes and objects) in polynomial time, is not enough in order to solve computationally hard problems efficiently. Conversely, a *negative answer* to the conjecture would provide a borderline between tractability and intractability (assuming that $\mathbf{P} \neq \mathbf{NP}$): division rules for non-elementary membranes. In [5], the non-efficiency of P systems from $\mathcal{AM}^0(-d, +ne)$ was demonstrated, giving a partial positive answer to the conjecture. In [2], a partial negative answer was given by means of a polynomial-time solution with a family of P systems from $\mathcal{AM}^0(+d, +ne)$. A particular case of the conjecture, where systems are mono-directional and deterministic was studied in [9], where they were demonstrated to be non-efficient systems; that is, only problems from class $\mathbf{P}$ can be solved efficiently by means of this kind of membrane systems (Table 2).

3.3 P systems with symport/antiport rules

A kind of cell-like P systems that use communication rules capturing the biological phenomenon of trans-membrane transports of several chemical substances was introduced in [27]. Specifically, two processes were considered. The first one (*symport* process) allows a multiset of chemical substances to pass through a membrane in the same direction. In the second one (*antiport* process), two multisets of chemical substances (located in two adjacent biological membranes) only pass with the help of each other (an *exchange* of objects between both membranes). Division rules and separation rules are defined in a similar manner as in P systems with active membranes. The environment of a P system with symport/antiport rules can be active or passive. If the environment is active (P systems *with environment*), it can receive and send objects, interacting with the skin membrane, with a predefined set of objects which will be located in the environment with an infinite multiplicity. If the environment is passive (P systems *without environment*) can only receive objects from the system, but cannot send them back into the system again. Besides, the environment alphabet is empty. Let us denote by $\mathcal{CC}$ the computing model of recognizer basic cell-like P systems with only communication (symport/antiport) rules. For each natural number $k \geq 1$, $\mathcal{CDC}(k)$ (resp., $\mathcal{CSC}(k)$) denotes the computing model of recognizer cell-like P systems with membrane division (resp., membrane separation) and with communication (symport/antiport) rules of length at most k (the length of a communication rule is the total number of objects involved in it). The corresponding computing models associated with cell-like P systems *without environment* will be denoted by $\widehat{\mathcal{CDC}}(k)$ and $\widehat{\mathcal{CSC}}(k)$, respectively.

Table 3 Frontiers of efficiency in P systems with symport/antiport rules

Non-efficient	Presumably efficient	Type of frontier
$\mathbf{PMC}_{\mathcal{CC}}$	$\mathbf{PMC}_{\mathcal{CDC}}$	Division rules
$\mathbf{PMC}_{\mathcal{CC}}$	$\mathbf{PMC}_{\mathcal{CSC}}$	Separation rules
$\mathbf{PMC}_{\mathcal{CDC}(1)}$	$\mathbf{PMC}_{\mathcal{CDC}(2)}$	Length of rules
$\mathbf{PMC}_{\mathcal{CSC}(2)}$	$\mathbf{PMC}_{\mathcal{CSC}(3)}$	Length of rules
$\mathbf{PMC}_{\widehat{\mathcal{CSC}}(k)}, k \geq 2$	$\mathbf{PMC}_{\widehat{\mathcal{CDC}}(k)}, k \geq 2$	Type of rules
$\mathbf{PMC}_{\widehat{\mathcal{CSC}}(k)}, k \geq 3$	$\mathbf{PMC}_{\mathcal{CSC}(k)}, k \geq 3$	Environment

On one hand, as a particular case of computing model of recognizer membrane systems whose membrane structure does not grow, the non-efficiency of the computing model $\mathcal{CC}$, associated with recognizer basic cell-like P systems with only symport/antiport rules, is obtained. On the other hand, the non-efficiency of the computing model $\mathcal{CDC}(1)$ and $\mathcal{CSC}(1)$, associated with recognizer P systems with membrane division or membrane separation, using only symport rules with length one, was established in [13]. Moreover, the non-efficiency of the computing model $\mathcal{CSC}(2)$ was established in [49]. With respect to the presumed efficiency of these computing models, in the context of membrane separation rules, it suffices to consider symport/antiport rules with length at most three [12], whereas using membrane division rules it is enough to consider symport/antiport rules with length at most two [19, 48]. Therefore, the computing models $\mathcal{CDC}(2)$ and $\mathcal{CSC}(3)$ are presumably efficient.

At this point, it seems interesting to analyze the role that the environment plays from a complexity point of view. First, with respect to the computing models associated with recognizer cell-like P systems with membrane division, a surprising result was obtained [17]: for each natural number $k \geq 1$, we have $\mathbf{PMC}_{\mathcal{CDC}(k)} = \mathbf{PMC}_{\widehat{\mathcal{CDC}}(k)}$, that is, to obtain the time complexity class associated with the computing model $\mathcal{CDC}(k)$, the role of the environment is irrelevant. Nevertheless, in the case of P systems with membrane separation, the situation is completely different: for each natural number $k \geq 1$ we have $\mathbf{PMC}_{\widehat{\mathcal{CSC}}(k)} = \mathbf{P}$ [11]. The frontiers of efficiency obtained in this framework have been summarized in Table 3.

4 Recognizer tissue-like membrane systems

In this section, the capability of the computing model of recognizer membrane systems working in a tissue-like manner, is analyzed from a complexity point of view.

Networks of membranes, which compute by communication only in the form of symport/antiport rules, were considered in [33]. In this computing paradigm, cells are connected through channels, where chemical substances go from one cell to another one across channels, in the same or opposite directions. Such rules are used both for communication with the environment and for direct communication between different membranes. It is worth noting that in such a system the environment plays an active role, because not only objects can be sent outside the system, but also objects can be brought into the system from the environment.

With respect to the tissue-like computation models, from the seminal definitions of tissue P systems [14, 15], one of the most interesting variants of tissue P systems was presented in [34]. In that paper, the definition of tissue P systems with symport/antiport rules is combined with the one of P systems with active membranes, yielding *tissue P systems with cell division*. Membrane fission was introduced into tissue P systems with symport/antiport rules through *cell separation* rules yielding *tissue P systems with cell separation* [21].

4.1 Tissue P systems with symport/antiport rules

Let $\mathcal{TC}$ be the computing model of recognizer basic tissue-like P systems with only communication (symport/antiport) rules. For each natural number $k \geq 1$, $\mathcal{TDC}(k)$ (resp., $\mathcal{TSC}(k)$) denotes the computing model of recognizer tissue-like P systems with membrane division (resp., membrane separation) and with symport/antiport rules of length at most k. The corresponding classes associated with tissue-like P systems *without environment* will be denoted by $\widehat{\mathcal{TDC}(k)}$ and $\widehat{\mathcal{TSC}(k)}$, respectively.

On one hand, as a particular case of computing model of recognizer membrane systems whose membrane structure does not grow, the non-efficiency of the computing model $\mathcal{TC}$, associated with recognizer basic tissue-like P systems with only symport/antiport rules, is obtained. On the other hand, the non-efficiency of the computing model $\mathcal{TDC}(1)$ and $\mathcal{TSC}(1)$, associated with recognizer tissue P systems with division or separation rules, using communication (symport/antiport) rules with length one, was established in [3]. Moreover, the non-efficiency of the computing model $\mathcal{TSC}(2)$ was established in [22]. With respect to the presumed efficiency of these computing models, in the context of membrane separation rules it suffices to consider symport/antiport rules with length at most three [40], whereas using membrane division rules it is enough to consider symport/antiport rules with length at most two [43]. Therefore, the computing models $\mathcal{TDC}(2)$ and $\mathcal{TSC}(3)$ are presumably efficient.

At this point, it seems interesting to analyze the role that the environment plays from a complexity point of view. First, with respect to the computing models associated with recognizer tissue P systems with membrane division, a similar surprising result was obtained for tissue P systems [37]: for each natural number $k \geq 1$ we have $\mathbf{PMC}_{\mathcal{TDC}(k)} = \mathbf{PMC}_{\widehat{\mathcal{TDC}(k)}}$, that is, to obtain the time complexity class associated with the computing model $\mathcal{TDC}(k)$, the role of the environment is irrelevant. Nevertheless, in the case of tissue with membrane separation, the situation is completely different: for each natural number $k \geq 1$, we have $\mathbf{PMC}_{\widehat{\mathcal{TSC}(k)}} = \mathbf{P}$ [10].

Table 4 Frontiers of efficiency in tissue P systems with symport/antiport rules

Non-efficient	Presumably efficient	Type of frontier
$\mathbf{PMC}_{\mathcal{TC}}$	$\mathbf{PMC}_{\mathcal{TDC}}$	Division rules
$\mathbf{PMC}_{\mathcal{TC}}$	$\mathbf{PMC}_{\mathcal{TSC}}$	Separation rules
$\mathbf{PMC}_{\mathcal{TDC}(1)}$	$\mathbf{PMC}_{\mathcal{TDC}(2)}$	Length of rules
$\mathbf{PMC}_{\mathcal{TSC}(2)}$	$\mathbf{PMC}_{\mathcal{TSC}(3)}$	Length of rules
$\mathbf{PMC}_{\widehat{\mathcal{TSC}(k)}}, k \geq 2$	$\mathbf{PMC}_{\widehat{\mathcal{TDC}(k)}}, k \geq 2$	Type of rules
$\mathbf{PMC}_{\widehat{\mathcal{TSC}(k)}}, k \geq 3$	$\mathbf{PMC}_{\mathcal{TSC}(k)}, k \geq 3$	Environment

It is worth pointing out that with respect to the analysis of the efficiency or presumed efficiency of computing models of recognizer membrane systems, the rooted tree structure, associated with cell-like membrane systems, or the directed graph associated with tissue-like membrane systems, do not play a relevant role. Therefore, the underlying structure of a membrane system does not matter from a complexity point of view. This can be observed in Table 4. As it can be seen in Table 3, the results are surprisingly the same, therefore the underlying structure of the system does not seem to affect the computational power of the systems.

4.2 Evolutional symport/antiport rules

Based on the communication of cells within a living tissue, where objects can evolve when rules are applied, a new variant of tissue P systems has been introduced in [47]. The concepts of *symport* and *antiport* are still present in this framework, in the sense that if objects are transported in one direction, it is called a symport rule, denoted by $[\,u\,]_i[\quad]_j \to [\quad]_i[\,u'\,]_j$, while if objects are interchanged in two directions it is called antiport rules, denoted by $[\,u\,]_i[\,v\,]_j \to [\,v'\,]_i[\,u'\,]_j$. Let us denote by $\mathcal{TEC}$ the computing model of recognizer basic tissue P systems with evolutional communication (symport/antiport) rules. For each natural number $k \geq 1$, $\mathcal{TDEC}(k)$ (respectively, $\mathcal{TSEC}(k)$) denotes the computing model of recognizer tissue P systems with cell division (resp., cell separation) and with evolutional communication rules such that the total number of objects implied in these rules is at most k. For each pair of natural numbers $k_1, k_2 \geq 1$, $\mathcal{TDEC}(k_1, k_2)$ (respectively, $\mathcal{TSEC}(k_1, k_2)$) denotes the computing model

Table 5 Frontiers of efficiency in tissue P systems with evolutional symport/antiport rules

Non-efficient	Presumably efficient	Type of frontier
$\mathbf{PMC}_{\mathcal{TEC}}$	$\mathbf{PMC}_{\mathcal{TDEC}}$	Division rules
$\mathbf{PMC}_{\mathcal{TEC}}$	$\mathbf{PMC}_{\mathcal{TSEC}}$	Separation rules
$\mathbf{PMC}_{\mathcal{TDEC}(1)}$	$\mathbf{PMC}_{\mathcal{TDEC}(2)}$	Length of rules
$\mathbf{PMC}_{\mathcal{TSEC}(2)}$	$\mathbf{PMC}_{\mathcal{TSEC}(4)}$	Length of rules
$\mathbf{PMC}_{\mathcal{TDEC}(1,k)}, k \geq 1$	$\mathbf{PMC}_{\mathcal{TDEC}(2,k)}, k \geq 1$	Length of rules (RHS)
$\mathbf{PMC}_{\mathcal{TSEC}(k,1)}, k \geq 2$	$\mathbf{PMC}_{\mathcal{TSEC}(k,2)}, k \geq 2$	Length of rules (LHS)
$\mathbf{PMC}_{\mathcal{TSEC}(1,k)}, k \geq 2$	$\mathbf{PMC}_{\mathcal{TSEC}(2,k)}, k \geq 2$	Length of rules (RHS)
$\mathbf{PMC}_{\mathcal{TSEC}(k,1)}, k \geq 2$	$\mathbf{PMC}_{\mathcal{TDEC}(k,1)}, k \geq 2$	Type of rules

of evolutional communication rules with length at most (k_1, k_2), being the length a pair of natural numbers defined by the number of objects in the left-hand side of the rule and in the right-hand side of the rule, respectively, and cell division (resp., cell separation).

The non-efficiency of membrane systems from $\mathcal{TEC}$ is proven in a similar way to their non-evolutional counterparts. In [47], the computational power of this kind of P systems where cell division is allowed is studied. Moreover, the non-efficiency of P systems where the lengths of the rules is at most 2 is demonstrated; that is, the *non-efficiency* of $\mathcal{TDEC}(2)$ is demonstrated. In the same work, an efficient solution to the SAT problem is given by a family of membrane systems from $\mathcal{TDEC}(4)$. Moreover, observing the solution, these systems can be considered to be from $\mathcal{TDEC}(3, 2)$, since there are rules which left-hand side has length 3, and there are other rules whose right-hand side has length 2. However, there are not any rules whose left-hand side length is 3 and whose right-hand side length is 2 at the same time, as indicated with $k = 4$.

In [24], separation rules were studied in this computing model, providing a proof of the non-efficiency of the classes $\mathcal{TSEC}(1, n)$ and $\mathcal{TSEC}(n, 1)$ (for each $n \geq 1$), as well the presumed efficiency of the class $\mathcal{TSEC}(3, 2)$ by providing an efficient solution to SAT by a family of these systems. This solution is, in fact, considered to be in $\mathcal{TSEC}(4)$. In [16], a better solution to the problem SAT in terms of the length of the rules is given. More precisely, an efficient solution is given by a family of tissue P systems from $\mathcal{TSEC}(2, 2)$, reducing the length of the left-hand side of the rules to 2.

Thinner frontiers were given in [18], where the classes $\mathcal{TDEC}(1, n)$ for $n \geq 1$ are proven to be non-efficient, and the classes $\mathcal{TDEC}(2, 1)$ and $\mathcal{TSEC}(2, 2)$ are proven to be presumably efficient. These results give new and thinner frontiers of efficiency in the framework of tissue P systems with evolutional communication rules. These frontiers have been summarized in Table 5.

5 Conclusions and future work

The study of computational complexity theory is one of the most active research lines in the framework of Membrane Computing. Several results are continuously emerging, each of them possibly providing another tool to tackle the **P** vs **NP** problem. One of the most important open problems in this area is the so-called Păun conjecture, that states $\mathbf{P} = \mathbf{PMC}_{\mathcal{AM}^0(+d,-ne)}$. Different approaches to this problem have been carried out, none of them giving a total answer. However, each of the partial solutions provide new developments in this field.

It seems interesting to think about all the possibilities of membrane systems as problem solvers, given their intrinsic massive parallelism. In this sense, providing new solutions with less ingredients would be ideal to get them implemented in highly parallel platforms. An interesting research line is to search for classes of P systems similar to $\mathcal{AM}^0(+d, -ne)$, to check which problems can be solved by means of families of P systems from these classes. Another breaking research line would be to find a novel technique to prove properties of P systems from $\mathcal{AM}^0(+d, -ne)$.

Finding new frontiers of efficiency is crucial for the search of new tools to attack the $\mathbf{P} \neq \mathbf{NP}$ conjecture. For instance, in the computing paradigm of tissue P systems with evolutional communication rules, is $\mathcal{TSEC}(3)$ an efficient computing model? Another interesting idea would be to understand the role of the environment in this kind of systems, and to translate the results of tissue P systems with evolutional communication rules to cell-like membrane systems so we could compare their relation with their non-evolutional counterparts.

Acknowledgements This work was supported in part by the research project TIN2017-89842-P, cofinanced by Ministerio de Economía, Industria y Competitividad (MINECO) of Spain, through the Agencia Estatal de Investigación (AEI), and by Fondo Europeo de Desarrollo Regional (FEDER) of the European Union.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Alhazov, A., Freund, R., Păun, Gh. (2004) P systems with active membranes and two polarizations. In Gh. Păun, A. Riscos-Núñez, Á. Romero-Jiménez, F. Sancho-Caparrini (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Sevilla, 2-7 February 2004, Research Group on Natural Computing, TR 01/2004, University of Seville, pp. 20-36.

2. Alhazov, A., & Pérez-Jiménez, M.J. (2007). Uniform solution of QSAT using polarizationless active membranes. In J. Durand-Lose and M. Margenstern (eds.) *Machines, Computations, and Universality. Lecture Notes in Computer Science*, **4664**, 122-133.
3. Gutiérrez-Escudero, R., Pérez-Jiménez, M. J., & Rius-Font, M. (2010). Characterizing tractability by tissue-like P systems. *Lecture Notes in Computer Science*, *5957*(5957), 289–300.
4. Gutiérrez-Naranjo, M. Á., Pérez-Jiménez, M. J., & Riscos-Núñez, A. (2005). A fast P system for finding a balanced 2-partition. *Soft Computing*, *9*(9), 673–678.
5. Gutiérrez, M.Á., Pérez-Jiménez, M.J., Riscos-Núñez, A., & Romero-Campero, F.J. (2006). On the power of dissolution in P systems with active membranes. In R. Freund, Gh. Paun, Gr. Rozenberg, A. Salomaa (eds.) *Membrane Computing, 6th International Workshop, WMC 2005*, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers. *Lecture Notes in Computer Science*, **3850**, pp. 224-240.
6. Gutiérrez-Naranjo, M. Á., Pérez-Jiménez, M. J., Riscos-Núñez, A., Romero-Campero, F. J., & Romero-Jiménez, Á. (2006). Characterizing tractability by cell-like membrane systems. In K. G. Subramanian, K. Rangarajan, & M. Mukund (Eds.), *Formal models, languages and applications* (pp. 137–154). Singapore: World Scientific.
7. Ionescu, M., Păun, Gh, & Yokomori, T. (2006). Spiking neural P systems. *Fundamenta Informaticae*, *71*(2–3), 279–308.
8. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., & Zandron, C. (2014). Simulating elementary active membranes. In M. Gheorghe et al. (eds.) *Membrane Computing, 15th International Conference, CMC 2014, Revised Selected Papers, Lecture Notes in Computer Science*, **8961**, pp. 284-299.
9. Leporati, A., Manzoni, L., Mauri, G., Porreca, A. E., & Zandron, C. (2018). Solving a special case of the P conjecture using dependency graphs with dissolution. In M. Gheorghe, et al. (Eds.), *Membrane Computing, 18th International Conference* (pp. 196–213). CMC 2017, Springer.
10. Macías-Ramos, L.F., Pérez-Jiménez, M.J., Riscos, A., Rius, M., & Valencia-Cabrera, L. (2013). The efficiency of tissue P systems with cell separation relies on the environment. In E. Csuhaj-Varjú, M. Gheorghe, G. Rozenberg, A. Salomaa, G. Vaszil (eds.) *Membrane Computing- 13th International Conference CMC 2012* Budapest, Hungary, August 28-31, 2012, Revised Selected Papers. *Lecture Notes in Computer Science*, 7762, pp. 243-256.
11. Macías-Ramos, L. F., Pérez-Jiménez, M. J., Riscos-Núñez, A., & Valencia-Cabrera, L. (2015). Membrane fission versus cell division: when membrane proliferation is not enough. *Theoretical Computer Science*, *608*, 57–65.
12. Macías-Ramos, L. F., Song, B., Valencia-Cabrera, L., Pan, L., & Pérez-Jiménez, M. J. (2016). Membrane fission: a computational complexity perspective. *Complexity*, *21*(6), 321–334.
13. Macías-Ramos, L.F., Song, B., Song, T., Pan, L., & Pérez-Jiménez, M.J. (2017). Limits on efficient computation in P systems with symport/antiport. In C. Graciani, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (eds.) Proceedings of the Fifteenth Brainstorming Week on Membrane Computing, January 31, February 3, Sevilla, Spain, pp. 147-160.
14. Martín Vide, C., Pazos, J., Păun, Gh., & Rodríguez Patón, A. (2002). A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. *Lecture Notes in Computer Science* **2387**, 290-299.
15. Martín Vide, C. Pazos, J., Păun, Gh. & Rodríguez Patón, A. (2003). Tissue P systems. *Theoretical Computer Science*, **296**, 295-326.
16. Orellana-Martín, D., Valencia-Cabrera, L., Song, B., Pan, L., & Pérez-Jiménez, M.J. (2018). Narrowing Frontiers with Evolutional Communication rules and Cell Separation. In D. Orellana, Gh. Păun, A. Riscos, L. Valencia (eds.), *Proceedings of the Sixteenth Brainstorming Week on Membrane Computing*, Sevilla, Spain, January 30 - February 2, pp. 123-162.
17. Orellana-Martín, D., Valencia-Cabrera, L., Song, B., Pan, L., & Pérez-Jiménez, M. J. (2020). P systems with symport/antiport rules: When do the surroundings matter? *Theoretical Computer Science*, *805*, 206–217.
18. Orellana-Martín, D., Valencia-Cabrera, L., Song, B., Pan, L., & Pérez-Jiménez, M.J. Tuning frontiers of efficiency in tissue P systems with evolutional communication rules. *Complexity*, to be published.
19. Orellana-Martín, D., Valencia-Cabrera, L., Riscos-Núñez, A., & Pérez-Jiménez, M. J. (2019). A path to computational efficiency through membrane computing. *Theoretical Computer Science*, *777*, 443–453.
20. Pan, L., & Ishdorj, T.-O. (2004). P systems with active membranes and separation rules. *Journal of Universal Computer Science*, *10*(5), 630–649.
21. Pan, L., & Pérez-Jiménez, M. J. (2010). Computational complexity of tissue-like P systems. *Journal of Complexity*, *26*(3), 296–315.
22. Pan, L., Pérez-Jiménez, M.J., Riscos, A., & Rius, M. (2012). New frontiers of the efficiency in tissue P systems. L. Pan, Gh. Paun, T. Song (eds.) *Pre-proceedings of Asian Conference on Membrane Computing (ACMC 2012)*, Huazhong University of Science and Technology, Wuhan, China, October 15-18, pp. 61-73.
23. Papadimitriou, C. H. (1994). *Computational complexity*. USA: Addison-Wesley Publishing Company.
24. Pan, L., Song, B., Valencia-Cabrera, L., & Pérez-Jiménez, M.J. (2018). The Computational Complexity of Tissue P Systems with Evolutional Symport/Antiport Rules. *Complexity*, *2018*(3745210), 21 pages, (https://doi.org/10.1155/2018/3745210).
25. Păun, Gh. (2002). *Membrane computing. an introduction*. Berlin: Springer-Verlag.
26. Păun, A., & Păun (2001). On the Membrane Computing based on splicing. In C. Martín-Vide, V. Mitrana (eds.) *Where Mathematics, Computer Science, Linguistics and Biology Meet: Essays in honour of Gheorghe Păun*, **54**, 409-422
27. Păun, A., & Păun, Gh. (2002). The power of communication: P systems with symport/antiport. *New Generation Computing*, *20*(3), 295–305.
28. Păun, Gh. (2000). Computing with membranes. *Journal of Computer and Systems Science*, *61*(1), 108–143.
29. Păun, Gh. (2001). P systems with active membranes: attacking NP-complete problems. *Journal of Automata, Languages and Combinatorics*, *6*, 75–90.
30. Păun, Gh. (2005). Further twenty six open problems in membrane computing. In M.Á. Gutiérrez-Naranjo, A. Riscos-Núñez, F.J. Romero-Campero, D. Sburlan (eds.) *Proceedings of the Third Brainstorming Week on Membrane Computing*, Fénix Editora, Sevilla, pp. 249-262.
31. Păun, Gh, Sakakibara, Y., & Yokomori, T. (2002). P systems on graphs of restricted forms. *Publicationes Mathematicae*, *60*, 635–660.
32. Păun, Gh., Yokomori, T. (1999). Membrane Computing based on splicing. In E. Winfree, D.K. Guidford (eds.) *DNA based Computers V. DIMACS, Series Discrete Mathematics Theoretical Computer Science*, **54**, 217-231.
33. Păun, A., Păun, Gh, & Rozenberg, G. (2002). Computing by communication in networks of membranes. *International Journal of Foundations of Computer Science*, *13*(6), 779–798.
34. Păun, Gh., Pérez-Jiménez, M.J., & Riscos-Núñez, A. (2008). Tissue P system with cell division. *International Journal of Computers, Communications & Control*, **Vol. III**, 3, 295-303.
35. Pérez-Jiménez, M.J., & Riscos-Núñez, A. (2004). A linear-time solution to the Knapsack problem using P systems with active membranes. In C. Martín-Vide, Gh. Păun, G. Rozenberg, A.

Salomaa (eds.) *Membrane Computing International Workshop, WMC 2003, Tarragona, Spain, July 17-22, 2003, Revised Papers. Lecture Notes in Computer Science*, **2933**, 250-268.

36. Pérez-Jiménez, M. J., & Riscos-Núñez, A. (2005). Solving the Subset-Sum problem by active membranes. *New Generation Computing*, *23*(4), 367–384.
37. Pérez-Jiménez, M. J., Riscos-Núñez, A., Rius-Font, M., & Romero-Campero, F. J. (2013). A polynomial alternative to unbounded environment for tissue P systems with cell division. *International Journal of Computer Mathematics*, *90*(4), 760–775.
38. Pérez-Jiménez, M.J., Romero-Jiménez, Á., & Sancho-Caparrini, F. (2002). Decision P systems and the **P** ≠ **NP** conjecture. *Lecture Notes in Computer Science*, **2597** (2003), 388-399. A preliminary version in Gh. Păun, C. Zandron (eds.) *Pre-proceedings of Workshop on Membrane Computing 2002, MolCoNet project-IST-2001-32008, Publication No. 1*, Curtea de Arges, Romanian, August 19-23, pp. 345-354.
39. Pérez-Jiménez, M. J., Romero-Jiménez, Á., & Sancho-Caparrini, F. (2003). Complexity classes in models of cellular computing with membranes. *Natural Computing*, *2*(3), 265–285.
40. Pérez-Jiménez, M. J., & Sosík, P. (2015). An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Separation. *Fundamenta Informaticae*, *138*(1–2), 45–60.
41. Pérez-Jiménez, M.J., Romero-Campero, F.J. (2005). Trading polarizations for bi-stable catalysts in P systems with active membranes. In G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, Gr. Rozenberg, A. Salomaa (eds.) *Membrane Computing, 5th International Workshop, WMC5*, Revised Selected and Invited Papers. *Lecture Notes in Computer Science*, 3365, 373-388.
42. Porreca, Antonio E., Leporati, Alberto, Mauri, Giancarlo, Zandron. P., Claudio (2011). systems with elementary active membranes: Beyond NP and coNP. In Marian Gheorghe, Thomas Hinze, Gheorghe Păun, Grzegorz Rozenberg,Arto Salomaa (eds.), *Membrane Computing, 11th International Conference, CMC 2010,Lecture Notes in Computer Science*, **6501**, pages 338-347.
43. Porreca, A.E., Murphy, N., & Pérez-Jiménez, M.J. (2012). An optimal frontier of the efficiency of tissue P systems with cell division. In M. García-Quismondo, L.F. Macías-Ramos, Gh. Paun, I. Pérez Hurtado, L. Valencia-Cabrera (eds.) *Proceedings of the Tenth Brainstorming Week on Membrane Computing, Volume II*, Seville, Spain, January 30- February 3, 2012, Report RGNC 01/2012, Fénix Editora, pp. 141-166.
44. Porreca, A.E. (2008). *Computational Complexity Classes for Membrane Systems*, Master Degree Thesis, Università di Milano-Bicocca, Italy.
45. Porreca, A.E., Leporati, A., Mauri, G., & Zandron, C. (2012). P systems simulating oracle computations. In M. Gheorghe, Gh. Păun, G. Rozenberg, A. Salomaa, S. Verlan (eds.) *Membrane Computing 12th International Conference, CMC 2011, Fontainebleau, France, August 23-26, 2011, Revised Selected Papers. Lecture Notes in Computer Science*, **7184**, 346-358.
46. Sosík, P., & Rodríguez-Patón, A. (2007). Membrane computing and complexity theory: a characterization of PSPACE. *Journal of Computer and System Sciences*, *73*(1), 137–152.
47. Song, B., Zhang, C., Pan, L. (2017). Tissue-like P systems with evolutional symport/antiport rules. *Information Sciences*, **378**, 177-193 (https://doi.org/10.1016/j.ins.2016.10.046).
48. Valencia-Cabrera, L., Song, B., Macías-Ramos, L.F., Pan, L., Riscos-Núñez, A., Pérez-Jiménez, M.J. (2015). Computational efficiency of P systems with symport/antiport rules and membrane separation. In L.F. Macías, Gh. Păun, A. Riscos, L. Valencia (eds) *Proceedings of the Thirteenth Brainstorming Week on Membrane Computing*, 2-6 February, Sevilla, Spain, pp. 325-370.
49. Valencia-Cabrera, L., Song, B., Macías-Ramos, L.F., Pan, L., Riscos-Núñez, A., Pérez-Jiménez, M.J. (2015). Minimal cooperation in P systems with symport/antiport: A complexity approach. In L.F. Macías, Gh. Păun, A. Riscos, L. Valencia (eds) *Proceedings of the Thirteenth Brainstorming Week on Membrane Computing*, 2-6 February, Sevilla, Spain, pp. 301-323
50. Vollmer, H. (1999). Introduction to circuit complexity—a uniform approach. *Springer*,. https://doi.org/10.1007/978-3-662-03927-4.
51. Zandron, C., Ferretti, C., & Mauri, G. (2000). Solving NP-complete problems using P systems with active membranes. In I. Antoniou, et al. (Eds.), *Unconventional Models of Computation* (pp. 289–301). Berlin: Springer-Verlag.