Journal of Membrane Computing (2023) 5:193–204
https://doi.org/10.1007/s41965-023-00132-x

RESEARCH PAPER

A topology for P-systems with active membranes

Alberto Dennunzio[1] · Enrico Formenti[2] · Luca Manzoni[3] · Luciano Margara[4] · Giuliamaria Menara[3]

Received: 19 August 2023 / Accepted: 29 November 2023 / Published online: 20 December 2023

Abstract

This paper proposes a study of deterministic P systems with active membranes in the context of discrete time dynamical systems. First of all, we prove that, for a fixed set of objects and labels, the set of all P system configuration is countable and that the dynamical behaviors defining a chaotic system are not possible. Then, we define a notion of distance between membrane configurations encoding the intuitive concept of "dissimilarity" between configurations. We prove that all functions defined by evolution, communication, and division rules are continuous under that distance and that the resulting topological space is discrete but not complete. Furthermore, we adapt in a natural way the classical notions of sensitivity to initial conditions and topological transitivity to P systems, and we show that P systems exhibiting those new properties exist. Finally, we prove that the proposed distance is efficiently computable, i.e., its computation only requires polynomial time with respect to the size of the input configurations.

Keywords Membrane computing · P systems · Discrete time dynamical systems · Topological spaces

1 Introduction

P systems are computational models inspired by the mechanisms and structures of the living cell. Introduced by Gh. Păun [1], they were immediately used to solve intractable problems efficiently [2, 3] due to their massively parallel nature. The main idea behind the definition of P systems is that the space is divided into regions, i.e., membranes, containing chemical substances, i.e., objects, that evolve in time (mimicking the chemical reactions that can happen inside cells) and move between regions. Thus, P systems provide a computational model that is both distributed (due to the subdivision of the space in regions) and massively parallel (due to the parallel evolution and movement of chemical objects). In addition, P systems with active membranes provide a hierarchical subdivision of space and the ability of membranes to duplicate (mimicking the process of mitosis), thus making P systems able to simulate the non-deterministic computations of a Turing machines "in parallel" with an interesting trade-off between the space used by the computation and the time need to perform it. Another prominent kind of P systems are tissue P systems [4–7], where the regions (cells) have a "flat" structure and objects are exchanged between pairs of cells or between cells and an external environment. In fact, during the years, multiple variants of P systems were introduced, as Enzymatic Numerical P Systems [8] and Spiking Neural P Systems [9], just to give some examples. Furthermore, the investigation of P systems and their computational power in terms of space and time resources were studied in depth [10–14], giving rise to an active field of research.

✉ Alberto Dennunzio
alberto.dennunzio@unimib.it

Enrico Formenti
enrico.formenti@univ-cotedazur.fr

Luca Manzoni
lmanzoni@units.it

Luciano Margara
luciano.margara@unibo.it

Giuliamaria Menara
giuliamaria.menara@phd.units.it

[1] Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milan, Italy

[2] Université Côte d'Azur, CNR,S I3S, Nice, France

[3] Dipartimento di Matematica, Informatica e Geoscienze, Università degli Studi di Trieste, Via Alfonso Valerio 12/1, 34127 Trieste, Italy

[4] Department of Computer Science and Engineering, University of Bologna, Cesena Campus, Via Sacchi 3, Cesena, Italy

In addition to be seen as modeling tools or as computational devices, *deterministic* P systems can also be considered as discrete time dynamical systems and their dynamics can then be studied. Indeed, given any deterministic P system Π, each configuration $\mathcal{C}$ of Π is a point of the related state space and the next state is given by the next configuration in the computation, which is unique under the assumption of determinism. Equipping the set of all configurations with a distance (and with the induced topology) allows one to study dynamical properties [15], as sensitivity to initial conditions, equicontinuity, chaotic behavior, and so on. Those dynamical properties describe which kind of behavior a system exhibits and they can have significant repercussions in the computational power of a P system. For example, suppose that the dynamics starting from a configuration $\mathcal{C}$ can never lead to configurations at distance from $\mathcal{C}$ greater than a certain threshold. Hence, if no accepting configurations are present at a distance from $\mathcal{C}$ below that threshold, then surely any computation containing $\mathcal{C}$ can never be an accepting one.

The introduction of a distance between membranes has already been a topic of some interest; see, for example, the work by López and Sempere [16]. However, the investigations were generally focused on other areas and not on topological dynamics.

In this study, we explore the possible dynamics that can be exhibited by P systems viewed as discrete time dynamical systems and we introduce a distance based on the intuitive notion of “dissimilarity” between membranes. Namely:

- We prove that configuration space is countable and, as a consequence, the dynamical behaviors as sensitivity to initial condition and topological transitivity defining Devaney’s notion of chaos cannot be exhibited.
- We show that with the proposed distance, any function defined by a set of evolution, communication, and division rules is continuous. The resulting topological space is discrete but not complete, with its completion requiring the addition of configurations where the nesting depth is infinite.
- We adapt, in a natural way, the notions of sensitivity to initial conditions and of topological transitivity to P systems and we show that P systems exhibiting the adapted properties actually exist.
- Finally, by exploiting a variant of the Bottleneck Bipartite Matching Problem, we prove that the proposed distance is efficiently computable, i.e., its computation only requires polynomial time with respect to the size of the input configurations.

We believe that the study of a distance for the explicit study of dynamical properties will encourage cross-fertilization between the P systems community and those studying other bio-inspired models, as cellular automata, where the topics of topological dynamics are well established, see, e.g., [17, 18] and, in particular, [19–25] for the case in which the configuration space is equipped with a certain algebraic structure (for example, a group structure).

The rest of the paper is structured as follows: in Sect. 2, we recall the necessary notions regarding P systems and discrete dynamical systems. There, we prove some general results on spaces with a countable number of states. In Sect. 3, we propose a distance between configurations and we illustrate the properties that it shows regarding the dynamics of P systems. We stress that one of the main interests of the introduction of such a distance is that it is not too rough and it can be computed in polynomial time. Indeed, its computability is the main matter of Sect. 4. In the last section, we draw our conclusions and propose some promising research directions.

2 Basic notions

In this section, we briefly recall the notion of P systems with active membranes without charges (for a comprehensive survey, the reader is referred to [26, 27]). Then, we define the configuration space and the global rule of a given P system. Finally, we review the basics on discrete time dynamical systems.

Before proceeding, let us recall the following fundamental notion. A **multiset** over a finite set A is a mapping $u : A \to \mathbb{N}$, such that for any $a \in A$, $u(a)$ is the number of elements of type a in the multiset represented by u. Setting an ordering on A, a multiset can also be conveniently and (uniquely up to the ordering) represented by a string $a_1^{u(a_1)} \dots a_k^{u(a_k)}$ with $a_i \in A$.

2.1 P systems with active membranes

Formally, a **P system** Π **with active membranes without charges** and of initial degree $d \geq 1$ is a tuple $\Pi = (\Gamma, \Lambda, \mu, w_{h_1}, \dots, w_{h_d}, R)$ where

- Γ is a finite set of symbols, i.e., an alphabet, where all the elements are usually called *objects*;
- Λ is a set of membrane labels, i.e., another alphabet disjoint from Γ;
- μ is a rooted unordered tree representing the membrane structure of Π in which each of the d nodes of the tree is mapped one-to-one to a label of Λ;
- $w_{h_1}, \dots, w_{h_d}$ with $h_1, \dots, h_d \in \Lambda$ are multisets of objects representing the initial content of the membranes;
- R is a finite set of rules of the following forms:

1. *Evolution rules* of the form $[a \to w]_h$ with $h \in \Lambda$, $a \in \Gamma$ and $w \in \Gamma^\star$ is a multiset over Γ.

Each of these rules can be applied when there is an object of type a inside a membrane with label h. When applied, that copy of a is removed and the multiset w is added to the content of the membrane.

2. *Send-out rules* of the form $[a]_h \to []_h\, b$, with $h \in \Lambda$ and $a, b \in \Gamma$.

   Each of these rules can be applied in a membrane having label h and containing an object of type a. When applied, an instance of a is removed from the content of membrane h and an instance of b is added to the enclosing membrane.
3. *Send-in rules* of the form $a\,[]_h \to [b]_h$, with $h \in \Lambda$ and $a, b \in \Gamma$.

   Each of these rules can be applied when in the membrane containing a membrane with label h, there is an instance of an object of type a. When applied, that instance is removed, an instance of b is added in membrane h.
4. *Division rules* of the form $[a]_h \to [b]_h\,[c]_h$, with $h \in \Lambda$ and $a, b, c \in \Gamma$.

   Each of these rules can be applied when there is an object of type a inside a membrane with label h. When applied, membrane h is duplicated including the entire membrane substructure of which it is the root along all its content. In both copies, an instance of a is removed and replaced with b in one copy and c in the other one.

Rules directly modifying the membrane structure (i.e., division rules) or allowing communication between membranes (i.e., send-in and send-out rules) are called *blocking rules*. From now on, when not specified otherwise, we will use the term P systems to denote P systems with active membranes and without charges.

An *instantaneous configuration* of a single membrane is denoted as $[w]_h$, where h is the membrane label and w is the multiset over Γ contained in the membrane h. A **configuration** $\mathcal{C}$ of a P system Π is a function which associates each node of the membrane structure with the instantaneous configuration of the corresponding membrane.

A computation step changes the configuration $\mathcal{C}$ in the following way:

- Each object or membrane can be subjected to at most one rule in each computation step. Regarding the evolution rules, the membranes are not directly involved, so multiple evolution rules can take place at the same time inside the same membrane.
- The application of rules is maximally parallel. This means that every object and membrane appearing on the left-hand side of a rule that can be applied must be subject to at least one rule. That is, an object or membrane remains idle only if no rule can be applied to it.
- When multiple conflicting rules can be applied at the same time, the choice of which rule will be applied is performed in a non-deterministic way.
- The outermost membrane—usually called the *skin* membrane—cannot be divided, and any object sent out from it cannot re-enter the system again. In this case, we say that an object has been sent out into the *environment*.
- Finally, when the rules to be applied have been selected, they are applied atomically and in parallel to the entire configuration $\mathcal{C}$.

Notice that, under these conditions, there might not be a *unique* configuration starting from $\mathcal{C}$ due to non-determinism in the conflict resolution. From now on, we require all computation steps to be deterministic. While this is a strong restriction, it can be obtained by adding a total ordering to the rules to establish a priority of application.

Since we are interested in the space of all possible configurations, we define $X_{\Gamma,\Lambda,\mu}$ as the set of all possible configurations over a given set of object types Γ and of membrane labels Λ. Notice that all P systems Π with the same Γ and Λ will give rise to the same set $X_{\Gamma,\Lambda,\mu}$ of configurations, without taking into account the fact that not all of them will be reachable from the initial membrane structure of Π, irrespective of the initial content of the membranes. For example, $X_{\Gamma,\Lambda,\mu}$ will contain configurations of any depth (maximum number of nesting), but the membrane structure μ of Π can never change in depth.[1]

The set of rules R induces a **global rule** $G_R : X_{\Gamma,\Lambda,\mu} \to X_{\Gamma,\Lambda,\mu}$ which describes the evolution of the system from any time step to the next one. Namely, given any configuration $\mathcal{C}$, $G_R(\mathcal{C})$ is the unique configuration obtained from performing one computation step starting from $\mathcal{C}$. In the following, G_R^t will indicate the t-fold application of G_R with itself (with the convention that G_R^0 is the identity map).

This particular definition of G_R is matter of choice. We defined it in that way to ensure that G_R is a function. Of course, ours is not the only way to obtain a function from the rule set R. However, in the present paper, the only property required for G_R is continuity and this property is independent from the choice we made.

2.2 Basics on discrete time dynamical systems

Formally, a **discrete (time) dynamical system** (DDS) is a structure $\langle X, f\rangle$, where X is a topological space, called the **space of states**, and f is a continuous function from X to itself. An **orbit** $O_f(x)$ of f with the initial condition

[1] Even when dissolution rules are considered, they only allow the membrane structure to decrease in depth.

$x \in X$ is the sequence $\{x, f(x), f^2(x), \dots, f^n(x), \dots\}$ where f^n is the n-fold composition of f with itself and f^0 is the identity function. An orbit $O_f(x)$ is **periodic** if there exists an integer p, such that $f^p(x) = x$. The least integer p with the previous property is the **period** of $O_f(x)$ and all elements of $O_f(x)$ are **periodic points**.

Studying a DDS means to investigate the properties of its orbits. In this paper, we will focus on two properties, namely, (topological) transitivity and sensitivity to initial conditions and we will assume that X is a metric space (so to be able to define sensitivity to initial conditions). Moreover, we denote $B_\varepsilon(x)$ the open ball of radius $\varepsilon > 0$ centered in $x \in X$.

Definition 1 (*Topological transitivity*) A DDS $\langle X, f \rangle$ is **(topologically) transitive** iff for any pair of non-empty open sets $U, V \subseteq X$ there exists $n \in \mathbb{N}$, such that $f^n(U) \cap V \neq \emptyset$.

Transitivity is a property of non-decomposability of the system. Indeed, a transitive system cannot be decomposed into two independent sub-systems. Transitivity is also a property often used to detect some form of chaotic behavior in the system.

Definition 2 (*Sensitivity*) A DDS $\langle X, f \rangle$ is **sensitive to initial conditions** if there exists $\epsilon > 0$ (the **sensitivity constant**), such that for every $x \in X$ and every $\delta > 0$, there exist $y \in B_\delta(x) \backslash \{x\}$ and $n > 0$, such that $d(f^n(y), f^n(x)) \geq \epsilon$.

The notion of sensitivity to initial conditions is concerned with the concept of instability of a system and it is often popularized under *the butterfly effect*: small perturbations in the initial conditions may originate huge differences upon iterations of the system.

Definition 3 (*DPO*) A DDS $\langle X, f \rangle$ has **dense periodic orbits (DPO)** if the set of periodic points is dense in X.

Denseness of periodic orbits is often seen as an element of *regularity* of a system and together with sensitivity and transitivity it composes the definition of **chaotic system** popularized by Robert Luke Devaney [28].

It has been proved that transitivity and DPO imply sensitivity to initial conditions when the state space X has infinite cardinality [29]. In the case of cellular automata, the situation is even simpler, namely, transitivity alone implies sensitivity to initial conditions [30]. Criticism and alternative proposals to Devaney's definition of chaos can be found in [31, 32] and the follow-up papers. In this paper, we consider Devaney's definition.

It is important to underline that when the cardinality of the space is countable, the dynamical properties that we have reviewed so far become meaningless, since no system can satisfy them. Before proving these facts, we need some preliminary results from topology.

Proposition 1 *A countably infinite complete metric space has an isolated point.*

Proof In [33, Th. 2.43 pag. 38], it is proved that perfect subsets of $\mathbb{R}^k$ are uncountable. However, in the proof of that result, the only properties of $\mathbb{R}^k$ that are used are the fact that it is a metric space (with the usual metric induced by the 2-norm) and that it is complete (this is because the Cantor nested closed set theorem is also needed in the proof). We can therefore turn the statement into "A complete and perfect metric space is uncountable". □

The previous proposition has strong consequences both on the topological structure of the space and on the possible dynamics on such a space as proved by the following corollaries

Corollary 1 *If X is a countably infinite metric space, then it is not compact.*

Proof By contradiction, assume X compact. Then, X is complete. By Proposition 1, we deduce that X has an isolated point x_1. Now, choose $\varepsilon_1 > 0$, such that $B_{\varepsilon_1}(x_1) = \{x_1\}$. Considering $V_1 = X \backslash B_{\varepsilon_1}(x_1)$, one finds that V_1 is closed and hence compact. Using Proposition 1 again, we find that there exists an isolated point x_2 in V_1. By iterating the same argument as before, we can build an infinite sequence $\{x_i\}_{i \in \mathbb{N}}$, such that

- $\forall i, j \in \mathbb{N},\ x_i \neq x_j$;
- $\forall i \in \mathbb{N},\ \exists \varepsilon_i$ s.t. $B_{\varepsilon_i}(x_i) = \{x_i\}$.

Let $W = \bigcup_{i \in \mathbb{N}} B_{\varepsilon_i}(x_i)$. W is an open set, since it is a union of open sets. Hence, $X \setminus W$ is compact, since it is a closed subset of the compact set X. Let U be an open cover of $X \setminus W$. Then, $U, B_{\varepsilon_1}(x_1), B_{\varepsilon_2}(x_2), \dots, B_{\varepsilon_n}(x_n), \dots$ is an open cover of X which does not admit any finite open subcover. □

Corollary 2 *If X is a complete countably infinite metric space, then no DDS having X as state space is sensitive to initial conditions.*

Proof The statement follows from the definition of sensitivity to initial conditions. Indeed, assume that $\langle X, f \rangle$ is sensitive to initial conditions and that ε is the sensitivity constant. By Proposition 1, let x be an isolated point of X and let $\xi > 0$ be such that $B_\xi(x) = \{x\}$ and $\xi < \varepsilon$. Then, for all $\delta > 0$, for all $y \in B_\xi(x)$ and for all $t \in \mathbb{N}$, one finds $d(f^t(x), f^t(y)) = d(f^t(x), f^t(x)) = 0 < \varepsilon$. □

Corollary 3 *If X is a complete countably infinite metric space, then X has a countably infinite number of isolated points.*

Proof Let us prove the statement by induction on $n > 0$. At stage $n = 1$, Proposition 1 provides the statement. Assume the statement true up to stage $n > 1$. Then, let $W_n = \bigcup_{i=1}^{n} B_{\varepsilon_i}(x_i)$, where the x_i are distinct isolated points produced at the previous stages and for all $1 \leq i \leq n$ it holds that $B_{\varepsilon_i}(x_i) = \left\{x_i\right\}$. Thus, W_n is an open set. As a consequence, $X \setminus W_n$ is a closed subset of a complete space, and hence, it is complete. By Proposition 1, $X \setminus W_n$ admits an isolated point x_{n+1} which is, obviously, distinct from those contained in W_n. □

Proposition 2 *If X is a complete countably infinite metric space, then no DDS $\langle X,f \rangle$ is (topologically) transitive.*

Proof By contradiction, assume that there exists a transitive DDS $\langle X,f \rangle$. Let $x_1, x_2, \dots, x_n, \dots$ be the countably infinite set of distinct isolated points from Corollary 3. For $i,j \in \mathbb{N}$ with $i \neq j$, consider x_i and x_j and let ε_i, (*resp.*, ε_j) be such that $B_{\varepsilon_i}(x_i) = \left\{x_i\right\}$ (resp., $B_{\varepsilon_j}(x_j) = \left\{x_j\right\}$). By transitivity, there exists $n_{i,j} \in \mathbb{N}$, such that $f^{n_{i,j}}(B_{\varepsilon_i}(x_i)) \cap B_{\varepsilon_j}(x_j) \neq \emptyset$, and hence, $f^{n_{i,j}}(x_i) = x_j$. Similarly, by exchanging the role of x_i and x_j, we have that there exists $n_{j,i} \in \mathbb{N}$, such that $f^{n_{j,i}}(x_j) = x_i$. Therefore, we conclude that all the isolated points must be in the same periodic orbit which is, of course, impossible unless X is finite. □

Using a very similar proof technique as in the previous proofs, one can prove the following.

Proposition 3 *If X is a metric space with an isolated point, then no DDS $\langle X,f \rangle$ can be sensitive to initial conditions.*

Proposition 4 *If X is a metric space with a countably infinite number of isolated points, then no DDS $\langle X,f \rangle$ can be transitive.*

A special situation that needs to be mentioned is when X is a finite set. In this case, no matter the metric with which X is equipped, all points of X are isolated and, hence, there are no sensitive systems on X. However, there might exist transitive systems. In this last case, the dynamics of such systems consists in a single periodic orbit.

Concerning DPO, it is clear that the structure of the state space X has a great impact on the dynamical systems acting on it as it is illustrated by the following result.

Proposition 5 *If X is a countable metric space and the DDS $\mathcal{A} = \langle X,f \rangle$ has DPO, then all isolated points of X are periodic points of $\mathcal{A}$. In particular, if X is finite, then all points are periodic points for $\mathcal{A}$ and f is a bijection.*

Proof If the DDS $\langle X,f \rangle$ has DPO, then any open set contains a periodic point. Hence, if $x \in X$ is an isolated point, consider $\varepsilon > 0$, such that $B_\varepsilon(x) \setminus \{x\} = \emptyset$ (this $\varepsilon > 0$ always exists, since x is isolated). Hence, the open set $B_\varepsilon(x) = \{x\}$ must contain a periodic point. We conclude that x must be a periodic point. If X is finite, then all points are isolated and, by the first part of the proof, we deduce that all points are periodic points. If all points are periodic, then f is clearly a bijection. Indeed, f is surjective, because, given a state $x \in X$ of period $p \geq 1$, $f^{p-1}(x)$ is a preimage of x. For the injectivity, take $x, y \in X$ with $x \neq y$. If $y \in O_f(x)$, then $f(x) \neq f(y)$, because $x \neq y$. If $y \notin O_f(x)$, then, by contradiction, assume $z = f(x) = f(y)$. Let p be the period of z. We have $f^2(x) = f^2(y)$, since $f(x) = f(y)$ and f is a function. And, more generally, we have $f^n(x) = f^n(y)$ for all integers $n > 0$. In particular, for $n = p$, one finds $x = f^p(x) = f^p(y) = y$ contradicting the initial hypothesis. □

Summing-up, we may conclude that in a complete countable metric space, no DDS can be chaotic according to Devaney's definition.

3 A distance on P systems and its induced topology

This section proposes a new distance suitable for the study of P systems with active membranes. One principle behind its definition is that two configurations differing in the skin membrane should be farther away than two configurations where the difference is deeper in the membrane structure. The reason for this is that, when used as computing devices, the output (acceptance or rejection) is signaled by an object being send out from the skin membrane into the environment. Hence, to obtain two different outputs, any difference between two configurations must "bubble up" in the membrane structure to reach the skin membrane. In this sense, two configurations with only one object of difference in the skin membrane—which may determine the output of the computation—should be maximally apart. Similarly, if a difference between two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$ remains deep in the membrane structure, then the output of the two computations starting at $\mathcal{C}_1$ and $\mathcal{C}_2$ will be the same and the two dynamics will stay "near" each other.

The first step allowing us to introduce the new distance is to *pair* the membranes of two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$ of $X_{\Gamma,\Lambda,\mu}$. Let $\hat{\mu}_1$ and $\hat{\mu}_2$ be the corresponding membrane structures in which each node $a \in \hat{\mu}_1 \cup \hat{\mu}_2$ is labeled with

$\ell(a) = hw \in \Lambda \times \Gamma^*$, i.e., its membrane label and the multiset of its contents, represented here as a string.[2] We are now able to define the notion of *pairing relation*. A set $M \subseteq \hat{\mu}_1 \times \hat{\mu}_2$ is a pair relation if, for any $(a, b) \in M$ with $a \in \hat{\mu}_1$ and $b \in \hat{\mu}_2$, it holds that

1. if a and b are the roots of the membrane structures then $\ell(a) = \ell(b)$, i.e., they have the same content and membrane label.
2. if a and b are not the root of the membrane structure and they have parent membranes a' and b', respectively, then $\ell(a) = \ell(b)$, $(a', b') \in M$, and for all $a' \in \hat{\mu}_1$ and $b' \in \hat{\mu}_1$ with $a' \neq a$ and $b' \neq b$, it holds that $(a, b') \notin M$ and $(a', b) \notin M$. In other terms, a and b have equal labels and content, and the same holds for all pairs of membranes in the path from them to the root of the membrane structure. Furthermore, a membrane can be paired only with one other membrane.

Notice that, as a consequence, a membrane will be paired only with one at the same depth in the membrane structure. Moreover, there might be multiple relations respecting the conditions above (trivially, the empty relation satisfies them).

We are now able to define a partial ordering across all possible relations on the basis of the depth of the first node a which is not "paired". Formally, given two pairing relations M and M', we say that $M \leq M'$ if there exists a depth $k \in \mathbb{N}$ and a node $a \in \hat{\mu}_1 \cup \hat{\mu}_2$ at depth k, such that for all $(a', b') \in M$, with $a' \neq a$ and $b' \neq b$, and for all $b \in \hat{\mu}_1 \cup \hat{\mu}_2$ at depth k, there exists $(a'', b'') \in M'$, such that $b = a''$ or $b = b''$. Intuitively, M has a node at depth k with no pairing, and M' has a pairing for all nodes at depth k. We call a relation M that is maximal with respect to this order a *perfect pairing*. Clearly, all perfect pairings between distinct configurations have the same depth k with an "unpaired" node (but not necessarily the same one in every relation).

Definition 4 (*Distance for P systems*) The function $d : X_{\Gamma,\Lambda,\mu} \to \mathbb{R}^+$ as follows. Consider two configurations $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$

$$d(\mathcal{C}_1, \mathcal{C}_2) = \begin{cases} 2^{-k} & \text{if } \mathcal{C}_1 \neq \mathcal{C}_2 \\ 0 & \text{otherwise,} \end{cases}$$

where k is the depth at which any perfect pairing between $\mathcal{C}_1$ and $\mathcal{C}_2$ has an unpaired node.

Intuitively, the value of the distance is obtained by pairing the membranes of the two configurations starting from the root "at best as possible" and then proceed until there is a level for which a pairing is not possible. At first glance the computation of the distance between two configurations seems to be expensive, since it requires the computation of a relation that is maximal across all possible pairings of nodes. However, we will provide an efficiently computable method that computes the distance (see Algorithm 1).

We now prove that d is, in fact, a distance.

Proposition 6 *The function d is a distance over $X_{\Gamma,\Lambda,\mu}$.*

Proof Consider any two configurations $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$. It is clear that $d(\mathcal{C}_1, \mathcal{C}_2) = 0$ iff $\mathcal{C}_1 = \mathcal{C}_2$ and that $d(\mathcal{C}_1, \mathcal{C}_2) = d(\mathcal{C}_2, \mathcal{C}_1)$. Let us prove that the triangular inequality holds. Consider a third configuration $\mathcal{C}_3 \in X_{\Gamma,\Lambda,\mu}$. Assume $d(\mathcal{C}_1, \mathcal{C}_3) = 2^{-k}$ and $d(\mathcal{C}_2, \mathcal{C}_3) = 2^{-l}$ with $k > 0$ and $l > 0$. Then, $\mathcal{C}_2$ coincides with $\mathcal{C}_3$ up to level l and, similarly, $\mathcal{C}_1$ coincides with $\mathcal{C}_3$ up to level k. Therefore, $\mathcal{C}_1$ and $\mathcal{C}_2$ must coincide at least up to level $\min(k, l)$. And, hence, the triangular inequality holds, since $2^{-k} + 2^{-l} > 2^{-\min(k,l)}$. □

3.1 Continuity and limit dynamics

As a first investigation, we show how the space $X_{\Gamma,\Lambda,\mu}$ and the distance d influence the properties exhibited by any global rule G_R.

One interesting observation is that, when considering the limit dynamics of a configuration, the space $X_{\Gamma,\Lambda,\mu}$ does not contain *all* limit configurations. This is due to the existence of membrane division rules. Let us consider the following configuration $\mathcal{C}$:

$$[[a]_{h_1}]_{h_0}$$

of a P system with only one rule: $[a]_{h_1} \to [a]_{h_1}\ [a]_{h_1}$. Clearly, for every $t \in \mathbb{N}$, the orbit with the initial condition $\mathcal{C}$ has the following configuration as element at time t:

$$[\underbrace{[a]_{h_1}\ [a]_{h_1}\ \cdots\ [a]_{h_1}}_{2^t \text{ copies}}]_{h_0},$$

which, at the limit, will result in a membrane structure with infinitely many membranes at one level. Denote $\bar{X}_{\Gamma,\Lambda,\mu}$, the set $X_{\Gamma,\Lambda,\mu}$ augmented with configurations admitting membrane structures with nodes that possibly have a countable number of children. The definition of the distance d can be extended to $\bar{X}_{\Gamma,\Lambda,\mu}$ without any change, since it is based on the depth of the difference in a pairing between two configurations. However, it is still important to keep this distinction and not use always $\bar{X}_{\Gamma,\Lambda,\mu}$, since over such a set, the distance

[2] To avoid the problem of multiple representations for the same multiset, we consider as a representation the smaller string in lexicographic order.

will not even be computable due to the existence of infinitely large configurations.

To explore the properties of $X_{\Gamma,\Lambda,\mu}$ and $\bar{X}_{\Gamma,\Lambda,\mu}$, we need to prove that $X_{\Gamma,\Lambda,\mu}$ (which is clearly infinite) is a countable set:

Proposition 7 *The set $X_{\Gamma,\Lambda,\mu}$ is countable.*

Proof First of all, notice that for a given membrane structure with $k \in \mathbb{N}$ membranes, the set of all possible configurations having that membrane structure is countable, since all the content of the membranes can be represented in $\mathbb{N}^{k|\Gamma|}$, i.e., as a multiset in which we count for each object a and membrane h how many instances of a are in membrane h.

For any given number k of membranes the set of membranes structures having k membranes is finite, and hence, for each given number of membranes k, the set of all possible configurations is still countable. Let us denote it by $X_{\Gamma,\Lambda,\mu}(k)$.

Since $X_{\Gamma,\Lambda,\mu} = \bigcup_{k\in\mathbb{N}} X_{\Gamma,\Lambda,\mu}(k)$ and the union of any countable sequence of countable set is countable, we get the desired result. □

On the other hand, all configurations of finite depth but with possibly infinitely many membranes are uncountable in number.

Proposition 8 *The set $\bar{X}_{\Gamma,\Lambda,\mu}$ is uncountable.*

Proof Assume that $\Gamma = \{a, b\}$. For any function $f : \mathbb{N} \to \{0, 1\}$, build a configuration $\mathcal{C}_f \in \bar{X}_{\Gamma,\Lambda,\mu}$, such that the associated membrane structure μ_f has a root with an infinite number of children, say, $N_1, N_2, \ldots$, all with the same label. For $n \in \mathbb{N}$, let the content of N_n be $a^n b$ if $f(n) = 1$ or a^n if $f(n) = 0$. Thus, for each function $f : \mathbb{N} \to \{0, 1\}$, it is possible to define an (infinite) configuration encoding it. Since the set $\{0, 1\}^{\mathbb{N}}$ of all functions from $\mathbb{N}$ to $\{0, 1\}$ is uncountable, $X_{\Gamma,\Lambda,\mu}$ is also uncountable. □

Denote T the topology induced by the metric d on $X_{\Gamma,\Lambda,\mu}$ and $\bar{T}$ the induced topology when d is extended to $\bar{X}_{\Gamma,\Lambda,\mu}$. The following propositions provide important characterizations of T and $\bar{T}$.

Proposition 9 *Both T and $\bar{T}$ are discrete (but not uniformly).*

Proof Consider a configuration $\mathcal{C}_1$ and let $\hat{\mu}_1$ be the corresponding membrane structure. Then, it is not difficult to see that $B_{1/2^{k+1}}(\mathcal{C}_1) \setminus \{\mathcal{C}_1\} = \emptyset$, where k is the height of $\hat{\mu}_1$. □

By Propositions 6, 8, and 9, we get that T is second-countable (i.e., there exists a countable base for open sets), but $\bar{T}$ is only first-countable (i.e., any point admits a countable local base for open sets) where first-countability is entailed by the fact that $\bar{T}$ is a metric space.

Proposition 10 *Neither T nor $\bar{T}$ is complete (and hence not compact).*

Proof For any $i \in \mathbb{N}$, let the configuration $\mathcal{C}_i$ contain a membrane structure $\hat{\mu}_i$ having height i and in which each node has exactly one child. Then, for $m > n$, $d(\mathcal{C}_n, \mathcal{C}_m) = 2^{-n}$ and, hence, $(\mathcal{C}_i)_{i\in\mathbb{N}}$ is a Cauchy sequence. However, $(\mathcal{C}_i)_{i\in\mathbb{N}}$ does not converge to a point in $X_{\Gamma,\Lambda,\mu}$ or $\bar{X}_{\Gamma,\Lambda,\mu}$. □

To turn T or $\bar{T}$ into a complete space, configurations with trees of infinite height should be included in these sets. Leaving aside the questions about the biological justification of allowing membrane structures of infinite height, it is not difficult to see that, once such configurations are included in $X_{\Gamma,\Lambda,\mu}$, the distance d is not longer computable, since in such configurations, one can encode any function from $\mathbb{N}$ to $\{0, 1\}$ and, hence, the halting problem, too. Note that completing the space and capturing all limit dynamics give rise to infinity in two different ways, i.e., in the depth or in the width of the membrane structure.

While it is true that, by the previous propositions, all global rules G_R must be continuous functions, the following property also holds regarding how much the distance between two configuration can grow with a single application of G_R:

Proposition 11 *For every global rules G_R and configurations $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$, if $d(\mathcal{C}_1, \mathcal{C}_2) = 2^{-k}$ then $d(G_R(\mathcal{C}_1), G_R(\mathcal{C}_2)) \leq 2^{-k+1}$.*

Proof If $d(\mathcal{C}_1, \mathcal{C}_2) = 2^{-k}$, it means that the two configurations are equal from the root to depth $k - 1$. Since we assume determinism, the only possibility for two configurations to differ after one computation step is for some of the rules applied at depth k to produce a difference at a level $h < k$. Among all possible rules, only send-out rules can produce a change in one of the membranes higher in the membrane structure and in particular only at the parent membrane. Hence, if the first difference between the two configurations is at depth k, then after one computation step it can be at most at depth $k - 1$. □

Concerning the study of the dynamical aspects of P systems, Proposition 9 has also another important consequence, namely, that any P system is continuous in both T and $\bar{T}$. However, the discrete topology imposes strong constraints on the dynamics of P systems, since all systems turn out to be equicontinuous, i.e., strongly stable. However, in the next section, we will see that some form of sensitivity to initial conditions is still possible.

3.2 Sensitivity to initial conditions and transitivity

We have seen that $\langle X, f\rangle$ is never sensitive if X is a countably infinite metric space, as the configuration set $X_{\Gamma,\Lambda,\mu}$ of any P system is once equipped with any distance. However, one can provide a weaker form of sensitivity that can still turn to be useful in practice. The one we propose here is specially crafted for P systems.

First of all, for each configuration $\mathcal{C} \in X_{\Gamma,\Lambda,\mu}$, let $\xi_{\mathcal{C}} \in \mathbb{R}$ be the minimum positive number, such that for all $r > \xi_{\mathcal{C}}$, it holds that $B_r(\mathcal{C}) \neq \{\mathcal{C}\}$, i.e., an open ball of radius $r > \xi_{\mathcal{C}}$ centered in $\mathcal{C}$ will contain at least another configuration.

Definition 5 (*Sensitivity to initial conditions for P systems*) A global rule G_R of any P system is *P-sensitive to initial conditions* if there exists a constant $\epsilon > 0$, such that for every configuration $\mathcal{C}_1$ and for every $\delta > \xi_{\mathcal{C}_1}$, there exist a configuration $\mathcal{C}_2 \in B_\delta(\mathcal{C}_1)$ with $\mathcal{C}_2 \neq \mathcal{C}_1$ and a natural $n > 0$ such that $d(G^n_R(\mathcal{C}_1), G^n_R(\mathcal{C}_2)) \geq \epsilon$.

Notice that the difference with respect to the standard definition of sensitivity is that here condition $\delta > 0$ is replaced by $\delta > \xi_{\mathcal{C}_1}$ to avoid the ball centered in $\mathcal{C}_1$ to be the singleton $\{\mathcal{C}_1\}$.

With respect to this revised definition, there exist global rules which are P-sensitive to initial conditions as the one in the following.

Example 1 Assume $\Gamma = \{a\}$ and let $\Lambda = \{h\}$. Let the set of rules only consist of the following rule: $[a]_h \to []_h\, a$. The effect of this rule is that all instances of the object a will be sent toward the root of the membrane structure.

Let $\mathcal{C}_1$ be any configuration in $X_{\Gamma,\Lambda,\mu}$ and let $\mathcal{C}_2$ be any configuration with the same membrane structure of $\mathcal{C}_1$ except that one of the membranes at the maximal depth in $\mathcal{C}_1$ has an additional child membrane in $\mathcal{C}_2$, containing exactly one instance of a. If $\mathcal{C}_1$ has depth k, then $\mathcal{C}_2$ will have depth $k+1$. By construction, the two configurations will coincide up to depth k, while they will differ at depth $k+1$, thus $d(\mathcal{C}_1, \mathcal{C}_2) = 2^{k+1}$, which is also the minimal distance possible between $\mathcal{C}_1$ and any other configuration. Hence, $\mathcal{C}_2 \in B_r(\mathcal{C}_1)$ for any $r > \xi_{\mathcal{C}_1} = 2^{k+1}$.

By the presence of a single send-out rule "moving" instances of a toward the root of the membrane structure, after a number $t \in \mathbb{N}$ of time steps (that will depend on the specific membrane structure and number of instances of a in it), $G_R(\mathcal{C}_1)$ and $G_R(\mathcal{C}_2)$ will differ in the number of instances of a in the root of the membrane structures, i.e., at depth 0, so that $d(G^t_R(\mathcal{C}_1), G^t_R(\mathcal{C}_2)) = 1$.

Since this holds for any configuration $\mathcal{C}_1 \in X_{\Gamma,\Lambda,\mu}$, we get that the system is P-sensitive to initial conditions with sensitivity constant $\varepsilon = 1$.

In addition to sensitivity, by Proposition 2, not even transitivity as originally formulated can be exhibited by DDS $\langle X_{\Gamma,\Lambda,\mu}, G_R\rangle$. It is possible to "patch" the standard definition of transitivity as done with sensitivity, but with some significant restrictions.

To follow the same idea as sensitivity, suppose that we only allow open balls of radius at least r for some real constant $r > 0$ as possible open sets appearing in the definition of transitivity. Then, there are three possible cases:

- If $r = 1$, then for all $\mathcal{C} \in X_{\Gamma,\Lambda,\mu}$, it trivially holds that $B_1(\mathcal{C}) = X_{\Gamma,\Lambda,\mu}$, and hence, the transitivity condition is satisfied.
- If $r \leq \frac{1}{2}$, then let $\mathcal{C}$ be a configuration having depth 0 (i.e., there is only the root in the membrane structure). In this case, $B_r(\mathcal{C}) = \{\mathcal{C}\}$ and, in order that the intersection condition inside the definition of transitivity is satisfied, the orbit starting at $\mathcal{C}$ would include all elements of $X_{\Gamma,\Lambda,\mu}$. Since $X_{\Gamma,\Lambda,\mu}$ includes configurations of any depth and G_R preserves the depth of the configuration on which it acts, this can never happen.
- If $\frac{1}{2} < r < 1$, then for all $\mathcal{C} \in X_{\Gamma,\Lambda,\mu}$, $B_r(\mathcal{C})$ is neither a singleton nor the entire space (and it actually does not change when varying r). This is the case we will consider in the sequel.

Hence, the only way of adapting transitivity to DDS $\langle X_{\Gamma,\Lambda,\mu}, G_R\rangle$ is allowing only open sets of radius between $\frac{1}{2}$ and 1 in the definition.

Definition 6 (*Transitivity for P systems*) A DDS $\langle X_{\Gamma,\Lambda,\mu}, G_R\rangle$ is *P-transitive* iff for any pair of configurations $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$, there exists $n \in \mathbb{N}$, such that $G^t_R(B_r(\mathcal{C}_1)) \cap B_r(\mathcal{C}_2) \neq \emptyset$ for any $\frac{1}{2} < r < 1$.

We now show that there exist P-transitive DDS $\langle X_{\Gamma,\Lambda,\mu}, G_R\rangle$.

Example 2 As in the previous example, let $\Gamma = \{a\}$, $\Lambda = \{h\}$, and let the set of rules contain only $[a]_h \to []_h\, a$. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be any two configurations, having m_1 and m_2 instances of the object a in the root of the membrane structure. Thus, for any $r \in \left(\frac{1}{2}, 1\right)$, the open ball $B_r(\mathcal{C}_1)$ (resp., $B_r(\mathcal{C}_2)$) is the set of all configurations in which the root of the membrane contains exactly m_1 (resp., m_2) instances of a in the outermost membrane (which has necessarily label h). Hence, to ensure that $G^t_R(B_r(\mathcal{C}_1)) \cap B_r(\mathcal{C}_2) \neq \emptyset$ for some $t \in \mathbb{N}$, it is necessary (and sufficient) to exhibit a configuration in $B_r(\mathcal{C}_1)$, whose orbit contains a configuration with exactly m_2 instances of a in the outermost membrane.

Let $\mathcal{C}'_1$ be the following configuration:

$$[a^{m_1}\overbrace{\underbrace{[[\cdots[a]_h]_h]_h}_{depth\ m_1+1}\cdots[[\cdots[a]_h]_h]_h}^{m_2\ copies}]_h.$$

Roughly speaking, the outer membrane of $\mathcal{C}_1'$ contains m_1 copies of the same linear membrane structure of depth $m_1 + 1$, each of them containing one copy of object a in the deepest membrane.

The dynamics of $\mathcal{C}_1'$ is as follows: in the first m_1 steps, all copies of object a in the outermost membrane will be set out one-by-one, "emptying" the membrane. At the same time, m_2 copies of a will have entered m_2 distinct membranes at depth 1 by m_1 consecutive applications of the send-out rule for each instance. At the next time step, all of them will be sent out in the outermost membrane, thus providing m_2 instances of a in the outermost membrane. Hence, we get $G_R^{m_1+1}(\mathcal{C}_1') \in B_r(\mathcal{C}_2)$, as required to prove P-transitivity of G_R.

While still restrictive, the adapted definition of transitivity allows capturing the idea that the iterated image of an open set intersects any other open set in the space.

4 Computability of the distance *d*

One of the motivations of introducing a distance in the context of P systems was to cope with practical applications or with simulations. Indeed, in these cases, one needs an effective algorithm to compute the distance.

Theorem 1 *For any $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$, it holds that $d(\mathcal{C}_1, \mathcal{C}_2)$ can be computed in polynomial time in the size of $\mathcal{C}_1$ and $\mathcal{C}_2$.*

The proof of Theorem 1 essentially exploits a connection between the computation of the distance and a variant of the Bottleneck Bipartite Matching Problem (BBMP) which can be solved in time $O(n\sqrt{nm})$ for a graph having n vertices and m edges [34]. We recall that BBMP is the problem that, given a bipartite undirected graph G in which any edge is associated with a weight, asks to find a maximal matching M, such that the largest weight in M is as small as possible.

The proof of Theorem 1 will essentially consist in an algorithm for computing the distance but before providing such an algorithm let us illustrate the data structures that it uses. We associate any pair of configurations $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$ with three matrices $T(\mathcal{C}_1)$, $T(\mathcal{C}_2)$, and $M(\mathcal{C}_1, \mathcal{C}_2)$. The first two matrices will be used to keep track of the tree-structure of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and of their membrane contents. The third one is used to account for the distances between nodes (sub-trees) in the two configurations. Given a configuration $\mathcal{C} \in X_{\Gamma,\Lambda,\mu}$, the matrix $T(\mathcal{C})$ has size $h \times n$, where h is the height of the tree-structure in $\mathcal{C}$ and n is the number of nodes of the level with the largest number of nodes. Therefore, $T(\mathcal{C})_{lev,pos}$ uniquely determines the node at level lev and position pos, where the position is computed from left to right. The content of an entry of $T(\mathcal{C})$ is the data structure described in Algorithm 1. If an entry of $T(\mathcal{C})$ does not represent any node, then it is left empty.

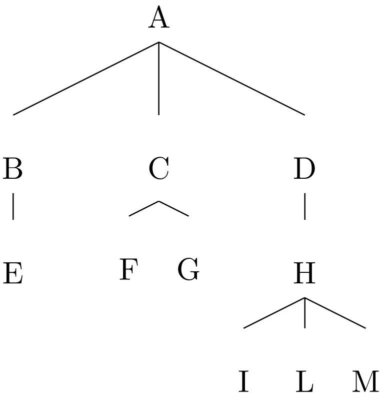

Fig. 1 An example of membrane structure

```
struct {
    string label // the name of the membrane and a string
        representing its content
    integer childnum // the number of children
    intpair childlist // (first,last) children of the current node in
        the list of nodes of the next level
    string content // the molecules contained in the membrane
}
```

Data Structure 1: The data structure for the entries of the matrix T

For example, assume that $\mathcal{C}$ has the membrane structure given in Fig. 1.

Then, $T(\mathcal{C})_{2,2}$.label = $\mathcal{C}$, $T(\mathcal{C})_{2,2}$.childnum = 2 , and $T(\mathcal{C})_{2,2}$.childlist = (2, 3). We also use two functions to have informations on the tree-like structure, namely, `height(`$\mathcal{C}$`)` which returns the height of the tree (in our previous example `height(`$\mathcal{C}$`)=4`) and `levelsize(`$\mathcal{C}$`,k)` which returns the total number of nodes of the level k in $\mathcal{C}$ (in our example `levelsize(`$\mathcal{C}$`,3)=4`). The matrix $M(\mathcal{C}_1, \mathcal{C}_2)$ is a three-dimensional matrix of size $k \times n \times m$, where k is the maximum between the heights of $\mathcal{C}_1$ and $\mathcal{C}_2$, while n (resp., m) is the number of columns of $T(\mathcal{C}_1)$ (resp. $T(\mathcal{C}_2)$).

Algorithm 1 The polynomial-time algorithm for computing the distance d between two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$.

```
Function d($T(\mathcal{C}_1), T(\mathcal{C}_2)$)
begin
    for $k \longleftarrow \max(height(\mathcal{C}_1), height(\mathcal{C}_2))$ down-to $1$ do
        for $i \longleftarrow 1$ to $levelsize(\mathcal{C}_1, \min(k, height(\mathcal{C}_1)))$ do
            for $j \longleftarrow 1$ to $levelsize(\mathcal{C}_2, \min(k, height(\mathcal{C}_2)))$ do
                if $T(\mathcal{C}_1)_{k,i}$ and $T(\mathcal{C}_2)_{k,j}$ are undefined then continue
                if $T(\mathcal{C}_1)_{k,i}.label \neq T(\mathcal{C}_2)_{k,j}.label$ then
                    $M(\mathcal{C}_1, \mathcal{C}_2)_{k,i,j} \longleftarrow 1$
                    continue
                end
                if $T(\mathcal{C}_1)_{k,i}.childnum \neq T(\mathcal{C}_2)_{k,j}.childnum$ then
                    $M(\mathcal{C}_1, \mathcal{C}_2)_{k,i,j} \longleftarrow 1/2$
                    continue
                end
                if $T(\mathcal{C}_1)_{k,i}.content \neq T(\mathcal{C}_2)_{k,j}.content$ then
                    $M(\mathcal{C}_1, C_2)_{k,i,j} \longleftarrow 1/2$
                    continue
                end
                // Build the complete bipartite graph $G$
                // as follows
                $(l_1, r_1) \longleftarrow T(\mathcal{C}_1)_{k,i}$
                $(l_2, r_2) \longleftarrow T(\mathcal{C}_2)_{k,j}$
                $L \longleftarrow \{T(\mathcal{C}_1)_{k+1,l_1}, T(\mathcal{C}_1)_{k+1,l_1+1}, \dots, T(\mathcal{C}_1)_{k+1,r_1}\}$
                $R \longleftarrow \{T(\mathcal{C}_2)_{k+1,l_2}, T(\mathcal{C}_2)_{k+1,l_2+1}, \dots, T(\mathcal{C}_2)_{k+1,r_2}\}$
                // Build the set of edges E and weights W
                for $a \longleftarrow l_1$ to $r_1$ do
                    for $b \longleftarrow l_2$ to $r_2$ do
                        $e \longleftarrow (T(\mathcal{C}_1)_{k+1,a}, T(\mathcal{C}_2)_{k+1,b})$
                        $E \longleftarrow E \cup \{e\}$
                        $W[e] \longleftarrow M(\mathcal{C}_1, \mathcal{C}_2)_{k+1,a,b}$
                    end
                end
                $G \longleftarrow \langle L \cup R, E \rangle$
                $res \longleftarrow$ AdHocBBPM($G, W$)
                if $res = 0$ then $M(\mathcal{C}_1, \mathcal{C}_2)_{k,i,j} \longleftarrow 0$
                else $M(\mathcal{C}_1, \mathcal{C}_2)_{k,i,j} \longleftarrow res/2$
            end
        end
    end
    return $M(\mathcal{C}_1, \mathcal{C}_2)_{1,1,1}$
end
```

Algorithm 2 The *ad hoc* version of the Bottleneck Bipartite Perfect Matching Problem used for the computation of the distance *d*.

```
Function AdHocBBPM($G, W$)
// $G$ is complete bipartite graph
// $W$ is a dictionary of weights, one entry per edge
begin
    $weights \longleftarrow$ getset($W$)
    $weights \longleftarrow$ sort($weights$)
    // addzero() adds a 0 value as first element of the array
       while all other elements are shifted to the right
    if $weight[1] \neq 0$ then $weights \longleftarrow$ addzero($weights$)
    for $i \longleftarrow 1$ to $weights.size$ do
        if BBPM($weigths[i]$) then return $weights[i]$
    end
end
```

Proof of Theorem 1 Given two configurations $\mathcal{C}_1, \mathcal{C}_2 \in X_{\Gamma,\Lambda,\mu}$, the Listing 1 provides a function which computes the distance between them. The algorithm uses a dynamic programming technique to fill bottom–up (and left to right) the matrix $M(\mathcal{C}_1, \mathcal{C}_2)$. When comparing two sub-trees (one from $\mathcal{C}_1$ and one from $\mathcal{C}_2$), three cases can occur:

1. the two roots have distinct labels, and hence, independently from the rest, the distance is 1 (as checked by the first `if`-block);
2. the two roots have the same label but distinct numbers of children and, hence, independently from the rest, the distance is 1/2 (as checked by the second `if`-block);
3. finally, it holds that the two roots have the same label and the same number of children. At this point, it is clear that we cannot compare all the possible combinations of children of one root with those of the other one without paying an exponential complexity cost. Therefore, we use a different strategy. Indeed, we use an *ad hoc* version of the bottleneck perfect matching problem. The idea is to build a complete bipartite graph G in which the children of the first root are the **left nodes** L and those of the second one are the **right nodes** R. Each left node is connected with a right node by an edge labeled with a weight given by their distance (that have been already computed at the previous step). A perfect matching in G using weights of value at most w provides a way to pair the children between the two sub-trees in which the worst distance between such two children is w. Therefore, `AdHocBBPM()` (see Listing 2) is invoked to find such a bottleneck perfect matching by starting with the lowest possible weight (i.e., 0) and progressively increasing it (this is the reason for the functions `getset()` which returns an array of unique values from W and `sort()` which sorts in ascending order the array taken in input) until a perfect matching is found. `AdHocBBPM()` returns the maximum weight in the matching found. We underline that a perfect matching always exists, since L and R have the same size and G is complete.

To conclude, the complexity of the algorithm in Listing 1 is given by the function `AdHocBBPM()`, the latter being called a number of times which is linear in the size of the largest matrix between $T(\mathcal{C}_1)$ and $T(\mathcal{C}_2)$. In its turn `AdHocBBPM()` invokes `BBPM()` a number of times which is quadratic in the size of $T(\mathcal{C}_1)$ or $T(\mathcal{C}_2)$ (since there is one weight per pair of nodes). By the main result in [34], we have the thesis. □

We now explain how Algorithms 1 and 2 work by the following illustrative example.

Example 3 Let us consider the following two configurations:

$$\mathcal{C}_1 = [[[]_{h_4}]_{h_2}\ []_{h_3}]_{h_1} \qquad \mathcal{C}_2 = [[[]_{h_6}]_{h_2}\ []_{h_5}]_{h_1}.$$

For a sake of simplicity, throughout this example, we will denote by $h_{i,j}$ the entire subtree rooted at membrane h_i in configuration $\mathcal{C}_j$.

Algorithm 1 is going to compute the distance between $h_{i,1}$ and $h_{j,2}$ for all i and j at the same depth in the two configurations, starting from the leaves and moving toward the outermost membrane, while saving the results in matrix $M(\mathcal{C}_1, \mathcal{C}_2)$.

Consider the computation of the distance between $h_{1,1}$ and $h_{1,2}$. A graph with nodes $h_{2,1}, h_{3,1}, h_{2,2}$, and $h_{5,2}$ (i.e., all children membranes of $h_{1,1}$ and $h_{1,2}$) is built. The edges are all the pairs with a node from $\mathcal{C}_1$ and the other one from $\mathcal{C}_2$, i.e., the pairs $(h_{i,1}, h_{j,2})$. In this way, a bipartite graph G is then obtained. The weight of each edge is the distance between the its two extremities, where the value of such a distance has already been computed and saved in matrix $M(\mathcal{C}_1, \mathcal{C}_2)$. At this point, the graph G and the weights of all edges are passed as arguments to the function `AdHocBBPM()` of Algorithm 2. Finally, a subroutine solving the BBPM problem is invoked to find the matching in the bipartite graph with the smallest maximum weight, i.e., the distance between $h_{1,1}$ and $h_{1,2}$.

5 Conclusions

In this paper, we provided a distance between configurations of P systems with active membranes and without charges with the explicit intent to study the dynamics generated by the transition function defined by a set of rules. To reach this objective, we provided a characterization of the dynamics in countable spaces, we defined a polynomial-time computable distance, and we adapted the common notions of sensitivity to initial conditions and transitivity to the case of P systems.

There are a large number of interesting questions for future research. First, it calls for a suitable definition of chaotic behavior. Indeed, we have seen that in a countable metric space, no chaotic behavior is possible. We proposed weak forms of sensitivity to initial conditions and transitivity, but this aspect should be investigated further. A connected question concerns the computability of the dynamical behavior. For instance, given a description of a P system in input, is it decidable to establish if the system is sensitive to initial conditions (in the weak sense, of course)?

In the study of discrete dynamical systems, a central issue consists in characterizing the *limit set*, i.e., the set of all the configurations which can appear after an arbitrary long number of iterations. Can our distance be of some help in figuring out the properties of the limit set?

Additionally, there are a number of questions regarding the interaction between a set of rules and possible dynamical behaviors. For example, what kind of additional dynamics are possible when membrane dissolution and creation are allowed?

Finally, in the definition of the global rule G_R, we required determinism. How can non-determinism be added? Moreover, have some common notions—like confluence—a topological characterization?

Acknowledgements This work was partially supported by PRIN PNRR P2022MPFRT CASCA ("Cellular Automata Synthesis for Cryptography Applications") and by MUR under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy.

References

1. Păun, Gh. (2000). Computing with membranes. *Journal of Computer and System Sciences, 61*(1), 108–143.
2. Păun, Gh. (2001). P systems with active membranes: Attacking NP-complete problems. *Journal of Automata, Languages and Combinatorics, 6*(1), 75–90.
3. Sosík, P. (2019). P systems attacking hard problems beyond NP: A survey. *Journal of Membrane Computing, 1*, 198–208.
4. Martín-Vide, C., Păun, Gh., Pazos, J., & Rodríguez-Patón, A. (2003). Tissue P systems. *Theoretical Computer Science, 296*(2), 295–326.
5. Song, B., Li, K., Orellana-Martín, D., Zeng, X., & Pérez-Jiménez, M. J. (2023). Tissue P systems with states in cells. *IEEE Transactions on Computers, 72*(9), 2561–2570.
6. Song, B., Li, K., & Zeng, X. (2022). Monodirectional evolutional symport tissue P systems with promoters and cell division. *IEEE Transactions on Parallel Distributed Systems, 33*(2), 332–342.
7. Song, B., Zeng, X., Jiang, M., & Pérez-Jiménez, M. J. (2021). Monodirectional tissue P systems with promoters. *IEEE Transactions on Cybernetics, 51*(1), 438–450.
8. Pavel, A. B., Arsene, O., & Buiu, C. (2010). Enzymatic numerical P systems—A new class of membrane computing systems. In Li, K., Tang, Z., Li, R., Nagar, A. K., & Thamburaj, R. (eds.) Proceedings 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2010), pp. 1331–1336.
9. Ionescu, M., Păun, Gh., & Yokomori, T. (2006). Spiking neural P systems. *Fundamenta Informaticae, 71*(2–3), 279–308.
10. 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.
11. Leporati, A., Manzoni, L., Mauri, G., Porreca, A. E., & Zandron, C. (2019). A gentle introduction to membrane systems and their computational properties. In T. Song, P. Zheng, M. L. D. Wong, & X. Wang (Eds.), *Bio-inspired computing models and algorithms* (pp. 1–32). World Scientific.
12. Leporati, A., Manzoni, L., Mauri, G., Porreca, A. E., & Zandron, C. (2018). A survey on space complexity of P systems with active membranes. *International Journal of Advances in Engineering Sciences and Applied Mathematics, 10*(3), 221–229.
13. Pan, L., & Pérez-Jiménez, M. J. (2010). Computational complexity of tissue-like P systems. *Journal of Complexity, 26*(3), 296–315.
14. Pérez-Jiménez, M. J. (2010). A computational complexity theory in membrane computing. In Păun, Gh., Pérez-Jiménez, M. J., Riscos-Núñez, A., Rozenberg, G., & Salomaa, A. (eds.)

Membrane Computing, 10th International Workshop, WMC 2009. Lecture Notes in Computer Science, vol. 5957, pp. 125–148. Springer.

15. Kůrka, P. (2003). *Topological and symbolic dynamics*. Paris: Société Mathématique de France.
16. López, D., & Sempere, J. M. (2005). Editing distances between membrane structures. In International Workshop on Membrane Computing, pp. 326–341. Springer.
17. Kari, J. (2005). Theory of cellular automata: A survey. *Theoretical Computer Science, 334*(1–3), 3–33.
18. Formenti, E., & Kůrka, P. (2009). Dynamics of cellular automata in non-compact spaces. In R. Meyer (Ed.), *Encyclopedia of complexity and systems science* (pp. 2232–2242). New York, NY: Springer.
19. Dennunzio, A., Formenti, E., Manzoni, L., Margara, L., & Porreca, A. E. (2019). On the dynamical behaviour of linear higher-order cellular automata and its decidability. *Information Sciences, 486*, 73–87.
20. Dennunzio, A., Formenti, E., & Margara, L. (2023). An easy to check characterization of positive expansivity for additive cellular automata over a finite abelian group. *IEEE Access, 11*, 121246–121255.
21. Dennunzio, A., Formenti, E., Grinberg, D., & Margara, L. (2021). Decidable characterizations of dynamical properties for additive cellular automata over a finite abelian group with applications to data encryption. *Information Sciences, 563*, 183–195.
22. Dennunzio, A., Formenti, E., Grinberg, D., & Margara, L. (2020). Chaos and ergodicity are decidable for linear cellular automata over (z/mz)n. *Information Sciences, 539*, 136–144.
23. Dennunzio, A., Formenti, E., Grinberg, D., & Margara, L. (2021). An efficiently computable characterization of stability and instability for linear cellular automata. *Journal of Computer and System Sciences, 122*, 63–71.
24. Béaur, P., & Kari, J. (2020). Decidability in group shifts and group cellular automata. In Esparza, J., & Král', D. (eds.) 45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic. LIPIcs, vol. 170, (pp. 12–11213). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
25. Dennunzio, A., Formenti, E., & Margara, L. (2024). An efficient algorithm deciding chaos for linear cellular automata over $(\mathbb{Z}/m\mathbb{Z})^n$ with applications to data encryption. *Inf Sci, 657*, 119942. https://doi.org/10.1016/j.ins.2023.119942.
26. Song, B., Li, K., Orellana-Martín, D., Pérez-Jiménez, M. J., & Pérez-Hurtado, I. (2022). A survey of nature-inspired computing: Membrane computing. *ACM Computing Surveys, 54*(1), 22–12231.
27. Păun, Gh., Rozenberg, G., & Salomaa, A. (Eds.). (2010). *The Oxford handbook of membrane computing*. Oxford University Press.
28. Devaney, R. L. (1989). *An introduction to chaotic dynamical systems*. Addison-Wesley advanced book program: Addison-Wesley.
29. Banks, J., Brooks, J., Cairns, G., Davis, G., & Stacey, P. (1992). On Devaney's definition of chaos. *American Mathematical Monthly, 99*, 332–334.
30. Codenotti, B., & Margara, L. (1996). Transitive cellular automata are sensitive. *The American Mathematical Monthly, 103*(1), 58–62.
31. Cattaneo, G., Formenti, E., Margara, L., & Mauri, G. (1999). On the dynamical behavior of chaotic cellular automata. *Theoretical Computer Science, 217*(1), 31–51.
32. Cattaneo, G., Formenti, E., Margara, L., & Mazoyer, J. (1997). A shift-invariant metric on s^{zz} inducing a non-trivial tolology. In Prívara, I., & Ruzicka, P. (eds.) Mathematical Foundations of Computer Science 1997, 22nd International Symposium, MFCS'97, Bratislava, Slovakia, August 25-29, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1295, (pp. 179–188). Springer.
33. Rudyn, W. (1995). *Principes D'analyse Mathématique*. Ediscience International.
34. Punnen, A. P., & Nair, K. P. K. (1994). Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem. *Discrete Applied Mathematics, 55*(1), 91–93.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Alberto Dennunzio received the M.Sc. and Ph.D. degrees in computer science from the University of Milano, in 1999 and 2004, respectively. He is currently an Associate Professor with the Informatics, System and Communication Department, University of Milano-Bicocca, Italy. His research interests include complex systems, cellular automata, including the classes of linear, higher-order, non-uniform, and asynchronous CA, along with their long-term behavior which is understood through the investigation of properties, such as stability, instability, chaos, periodic behaviors, reachability, and reversibility. In particular, formal models for describing and simulating complex systems are considered and studied.

Enrico Formenti received the Ph.D. degree from Ecole Normale Supérieure de Lyon, in 1998. He is currently a Professor in computer science with Université Côte d'Azur, France. Since 2003, he has been a Full Professor with Université Nice Sophia Antipolis. His main research interests include discrete dynamical systems, chaos, tilings, and complex systems in general, but he is also interested in computational complexity, computability, and unconventional models of computation.

Luca Manzoni is an associate professor at the University of Trieste, Italy. He obtained his Ph.D. in Computer Science at the University of Milano-Bicocca in 2013. In 2012, he obtained a JSPS postdoctoral fellowship. In 2017, he obtained an award as the best young postdoc in Computer Science and Mathematics at the University of Milano-Bicocca. He has published more than 100 papers in international journal, conferences, and workshops. His interests are in the areas of natural computing models, like P systems, reactions systems, and cellular automata, and in the area of evolutionary computation, and genetic programming in particular.

Luciano Margara received the Laurea degree in scienze dell'informazione and the Dottorato di Ricerca in Informatica (Ph.D.) degree in computer science from the University of Pisa, in 1991 and 1995, respectively. He is currently a Professor in computer science with the University of Bologna, Italy. He joined the University of Bologna, in 1995 (a Research Associate, from 1995 to 2000, and an Associate Professor, from 2000 to 2005). His research interests include discrete time dynamical systems, optical networks, computational complexity, and bioinformatics.

Giuliamaria Menara is currently a Ph.D. student at the University of Trieste (Italy). She received her Bachelor's and Master's in Pure Mathematics from the University of Trieste and she wrote her Master's Thesis in Applied Topology. In 2021 she received her Master's in Applied Mathematics from the University of Delaware. Her research interests include Algebraic Topology, its applications to Neural Networks and Neurosciences, Topological Data Analysis, Evolutionary Algorithms, and Topological Dynamics.