Tissue P systems in the Euclidean space*

Alberto Leporati, Luca Manzoni, Giancarlo Mauri,
Antonio E. Porreca, and Claudio Zandron

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
{leporati,luca.manzoni,mauri,porreca,zandron}@disco.unimib.it

Abstract. Tissue P systems with cell division or cell separation have been proved able to solve **NP**-complete problems in polynomial time by trading space for time. We show that, when tissue P systems are embedded into the Euclidean space $\mathbb{R}^3$, the power of division and separation decreases due to the geometrical constraints of the space and, as a result, only problems in **P** can be solved in polynomial time.

1 Introduction

Tissue P systems are biologically inspired computing devices based on communication between cells of a tissue. From a complexity theoretic perspective, tissue P systems with cell division [11] or cell separation rules [10] are able to solve **NP**-complete problems in polynomial time by trading space for time, i.e., creating in polynomial time exponentially many cells working in parallel.

One of the assumptions underlying the normal definition of tissue P system, a feature that is critically exploited in the solution of **NP**-complete problems, is that each cell can in principle directly communicate with every other cell (having a different label). While this is reasonable for small systems, it is certainly unreasonable when the number of cells grows exponentially with the input size (for instance, the number of cells in a human body is about $3.72 \times 10^{13} \approx 1.06 \times 2^{45}$ [2], a number corresponding, e.g., to an instance of Boolean satisfiability with about 45 variables). Therefore, it is essential for large tissue P systems to take into account the placement of the cells in the three-dimensional Euclidean space.

As an abstraction, we model the geometric space where the tissue P systems are located as a graph, where vertices are possible locations for cells and edges are interpreted as proximity, that is, as the possibility of the cells located in the two extremes to communicate. This spatial graph model is a direct generalisation of standard tissue P systems and include them as a particular case.

In order to model systems that can actually be constructed, we characterise the spatial graphs that can be "embedded" in the three-dimensional Euclidean space. This embedding has only two restrictions, that can be informally stated

* This work was partially supported by Università degli Studi di Milano-Bicocca, Fondo d'Ateneo 2014: "Complessità computazionale nei sistemi a membrane".

as follows: cells that are considered "near" in the spatial graph cannot be placed too far in $\mathbb{R}^3$, and cells that are considered "distant" cannot be placed too close in $\mathbb{R}^3$. Practically speaking, the spatial graph must represent not-too-distorted real world distances.

The idea of giving a geometrical or topological dimension to models of P systems was previously investigated by Margenstern, with P systems in the hyperbolic space [7], by Barbuti et al., with spatial P systems [1], and by Csuhaj-Varjú et al., with P systems associated with a topology limiting the interactions between objects [3]. As described in Section 3, these models are however rather different in scope and features from what is proposed in this paper.

We show that these "realistic" tissue P systems are severely limited in their ability to increase their number of cells when working in polynomial time (their growth falls from exponential to at most polynomial – actually cubic – in $\mathbb{R}^3$). This, as a consequence, limits their computational ability to **P**. Therefore, we can claim that the current algorithms solving **NP**-complete problems by means of tissue P systems can only be used for small instances, where the assumption of direct communication among all the cells holds. For large instances the geometry of the Euclidean space cannot be ignored and, unfortunately, it does not seem to permit escaping from **P**.

2 Basic notions

We begin by recalling the definition of tissue P systems with division and separation rules; for a more detailed introduction on multiset processing and tissue P systems, we refer the reader to the original paper [8].

Definition 1. *A* tissue P system *is a structure* $\Pi = (\Gamma, E, w_1, \ldots, w_d, R)$*, where:*

- Γ *is an alphabet, i.e., a finite non-empty set of symbols, usually called* objects*;*
- $E \subseteq \Gamma$ *is the alphabet of objects initially located in the external environment, in* infinitely *many copies;*
- $d \geq 1$ *is the* degree *of the system, i.e., the initial number of cells;*
- $w_1, \ldots, w_d$ *are* finite *multisets over* Γ*, describing the initial contents of the* d *cells; here* $1, \ldots, d$ *are* labels *identifying the cells of the P system, and* 0 *is the label of the external environment;*
- R *is a finite set of rules.*

The rules of R are of the following types:

(a) *Communication rules*, denoted in this paper by $[u]_h \leftrightarrow [v]_k$ and in the literature by $(h, u/v, k)$, where h and k are distinct labels (including the environment), and u and v are multisets over Γ (at least one of them nonempty): these rules are applicable if there exist a region with label h containing u as a submultiset and a region k containing v as a submultiset; the effect of the rule is to exchange u and v between the two regions. If $h = 0$ (resp., $k = 0$) then u (resp., v) must contain at least one object from $\Gamma - E$, i.e.,

an object with finite multiplicity[1]. In this paper we consider a rule $[u]_h \leftrightarrow [v]_k$ and its syntactic reverse $[v]_k \leftrightarrow [u]_h$ to be the same rule.

(b) *Division rules*, of the form $[a]_h \rightarrow [b]_h\ [c]_h$, where $h \neq 0$ is a cell label and $a, b, c \in \Gamma$: these rules can be applied to a cell with label h containing at least one copy of a; the effect of the rule is to divide the cell into two cells, both with label h; the object a is replaced in the two cells by b and c, respectively, while the rest of the original multiset contained in h is replicated in both cells.

(c) *Separation rules*, of the form $[a]_h \rightarrow [\Gamma_1]_h\ [\Gamma_2]_h$, where $h \neq 0$ is a cell label, $a \in \Gamma$, and $\{\Gamma_1, \Gamma_2\}$ is a partition of Γ: these rules can be applied to a cell with label h containing at least one copy of a; the effect of the rule is to separate the cell into two cells, both with label h; the object a is consumed, while the objects from Γ_1 in the original multiset contained in h are placed inside one of the cells, and those from Γ_2 in the other. All separation rules in R must share the same partition $\{\Gamma_1, \Gamma_2\}$ of Γ.

A *tissue P system with cell division* only uses communication and division rules, while a *tissue P system with cell separation* only uses communication and separation rules.

A *configuration* $\mathcal{C}$ of a tissue P system consists of a multiset over $\Gamma - E$ describing the objects appearing with finite multiplicity in the environment, and a multiset of pairs (h, w), where h is a cell label and w a finite multiset over Γ, describing the cells. A *computation step* changes the current configuration according to the following set of principles:

- Each object can be subject to at most one rule, and each cell can be subject to *any number* of communication rules or, *alternatively*, a *single* division or separation rule.
- The application of rules is *maximally parallel*: each cell is subject to a maximal multiset of rules (i.e., no further rule can be applied).
- When several conflicting rules can be applied at the same time, a nondeterministic choice is performed; this implies that, in general, multiple possible configurations can be reached after a computation step.

A *computation* $\vec{\mathcal{C}} = (\mathcal{C}_0, \mathcal{C}_1, \ldots)$ of the tissue P system Π is a (possibly infinite) sequence of configurations where $\mathcal{C}_0$ is the *initial configuration*, and every configuration $\mathcal{C}_{i+1}$ is reachable from $\mathcal{C}_i$ via a single computation step. A finite computation $\vec{\mathcal{C}} = (\mathcal{C}_0, \ldots, \mathcal{C}_k)$ is *halting* if no rules are applicable in $\mathcal{C}_k$.

Tissue P systems can be used as language *recognisers* by employing two distinguished objects *yes* and *no*: we assume that all computations are halting, and that either object *yes* or object *no* (but not both) is released into the environment, and only in the last computation step, in order to signal acceptance or rejection, respectively. If all computations starting from the same initial

[1] Since communication rules are applied in a maximally parallel way, this restriction avoids the situation where infinitely many objects from the environment simultaneously enter a cell.

configuration are accepting, or all are rejecting, the tissue P system is said to be *confluent*.

In order to solve decision problems (i.e., decide languages), we use *families* of recogniser tissue P systems $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$. Each input x is associated with a tissue P system Π_x that decides the membership of x in the language $L \subseteq \Sigma^\star$ by accepting or rejecting. The mapping $x \mapsto \Pi_x$ must be efficiently computable for inputs of any length, as discussed in detail in [9].

Definition 2. *A family of tissue P systems $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ is said to be (polynomial-time) semi-uniform if the mapping $x \mapsto \Pi_x$ can be computed by a deterministic polynomial-time Turing machine.*

Any explicit encoding of Π_x is allowed as output of the construction, as long as the number of cells and objects represented by it does not exceed the length of the whole description, and the rules are listed one by one. This is also called a permissible encoding [9].

Semi-uniform families of confluent tissue P systems working in polynomial time are known to solve **NP**-complete problems when using cell division [11] or cell separation rules [10], while only problems in **P** are solvable when the tissue P systems are limited to communication rules [5].

In the rest of the paper, for brevity we use the term "tissue P system" to denote systems with communication and cell division or cell separation rules.

3 The geometry of tissue P systems

We introduce the notion of geometrical space in tissue P systems by means of an underlying undirected graph. This spatial graph, not to be confused with the graph implied by the communication rules of tissue P systems, is an abstract representation of the space in which tissue P systems are located; each vertex represents a possible location for a cell, while the edges represent spatial proximity. Only cells located at vertices connected by an edge can actually communicate (assuming a suitable communication rule actually exists); furthermore, the tissue P system can expand by division or separation only into adjacent empty vertices.

The spatial graph represents the geometrical space where a family of tissue P systems lives, and as such does not depend on the individual member of the family. Examples of geometrical spaces that we may want to model as spatial graphs are a test tube, a biochemistry laboratory, or the whole Euclidean space. Therefore, we do not require the spatial graph to be finite[2].

Definition 3. *Given a (possibly infinite) connected undirected graph $G = (V, E)$, a* tissue P system *Π* over *G is a tissue P system where the cells of each configuration are injectively mapped into the set of vertices V. The initial location of the cells is given as part of the initial configuration of Π, and the evolution works as for a standard tissue P system, with the following restrictions:*

[2] From a mathematical perspective, the infiniteness of the space allows us to process arbitrarily large inputs with tissue P system algorithms when the number of cells grows with the input size.

– *communication can only occur between cells located at adjacent vertices of the spatial graph;*
– *communication with the environment is always possible;*
– *cell division and cell separation can be applied to a cell located at vertex u only when an adjacent empty vertex v exists; the two resulting cells are nondeterministically placed at vertices u and v;*
– *except as an effect of cell division or cell separation, the cells keep their position on the spatial graph.*

In general, we deal with families *of tissue P systems $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ sharing the same spatial graph G.*

The ability of every cell to communicate directly with the environment corresponds to assuming the presence of an "interstitial fluid" surrounding each cell. Other models of communication with the environment are possible, such as the environment only being accessible to cells having an empty adjacent vertex; however, these restrictions can only decrease (if at all) the power of tissue P systems, and do not influence the results of this paper.

First of all, we remark that standard tissue P systems are a special case of tissue P systems over a spatial graph.

Theorem 1. *Standard tissue P systems are tissue P systems over the complete graph with a countably infinite number of vertices.*

Proof. The complete spatial graph poses no restriction to the communication between cells other than those given by the rules of the tissue P systems, and the initial position of the cells is immaterial. Since tissue P systems have a finite number of cells at each computation step, each cell always has free adjacent vertices; thus, cell division and separation are not limited by the shape of the space but, once again, only by the rules of the system. As a consequence, the evolution of the tissue P systems is the same as in the standard model with no underlying spatial graph. □

We now recall a notion of embedding for metric spaces [4]. Each connected undirected graph G is a metric space with distance $d_G(u, v)$ being the length of the shortest path between vertices u and v [4]. We then prove that graphs embedded into finite-dimensional Euclidean spaces (i.e., in a "realistic" setting) have growth bounded by a polynomial.

Definition 4. *Let (X, d_1) and (Y, d_2) be metric spaces. A map $\varphi\colon X \to Y$ is called a* bi-Lipschitz embedding of X into Y *if and only if there exist two positive real constants α and β such that, for all $x, y \in X$, the following inequalities hold:*

$$\alpha \times d_1(x, y) \leq d_2\big(\varphi(x), \varphi(y)\big) \leq \beta \times d_1(x, y).$$

These embeddings possess the two properties that we consider essential with respect to the geometry of tissue P systems: the centres of two cells have a minimum distance (as given, for instance, by the radius of the cells), and two *communicating* cells cannot be too far away. Bi-Lipschitz embeddings are always injective and continuous.

Example 1. The d-dimensional hypercubic grid over $V = \mathbb{Z}^d$ is bi-Lipschitz embeddable into the Euclidean space $\mathbb{R}^d$ via $\varphi(\vec{x}) = \vec{x}$. Indeed, the Euclidean distance between $\vec{x}$ and $\vec{y}$ is at most equal to the number of edges between the two vertices (when the two vertices lie on the same grid line) and at least $\frac{1}{\sqrt{d}} \times d_G(\vec{x}, \vec{y})$ (when they are opposite vertices of a hypercube). This is equal to the Manhattan (or taxicab) distance between $\vec{x}$ and $\vec{y}$ in $\mathbb{R}^d$ [4]. Hence, by choosing $\alpha = \frac{1}{\sqrt{d}}$ and $\beta = \sqrt{d}$ we have

$$\frac{1}{\sqrt{d}} \times d_G(\vec{x}, \vec{y}) \leq \|\vec{x} - \vec{y}\| \leq d_G(\vec{x}, \vec{y}) \leq \sqrt{d} \times d_G(\vec{x}, \vec{y})$$

as required.

Example 2. The complete graph with a countably infinite number of vertices cannot be bi-Lipschitz embedded into the Euclidean space $\mathbb{R}^d$ for any $d \in \mathbb{N}$. Indeed, assume the opposite and let $v \in V$. Since v has infinitely many adjacent vertices u (for which $d_G(u, v) = 1$) and

$$\alpha \leq \|\varphi(v) - \varphi(u)\| \leq \beta$$

there are infinitely many vertices in the ball of radius β centred around $\varphi(v)$, for each $\beta > 0$. But then there must exist arbitrarily close vertices, contradicting the leftmost inequality.

Unlike the geometric framework proposed here, in spatial P systems [1] membranes correspond to perimeters of polygons drawn over the bidimensional square grid, and multisets are located inside squares. This model (or rather its generalisation to higher-dimensional grids) is thus, in a way, more detailed than what we propose here, as it describes the shape of individual membranes as opposed to its spatial relationship with its neighbours. The model proposed by Margenstern [7] does also involve the embedding of membranes into a geometric space (namely, the hyperbolic plane, which is not considered as a "realistic" space as discussed in Section 4). On the other hand, the crucial issue we tackle here, the exponential proliferation of cells by division or separation, is not investigated in these two frameworks.

We also recall the notion of graph growth, that describes how many vertices are reachable at most by means of finite paths of increasing length [4].

Definition 5. *A graph $G = (V, E)$ has* growth $g \colon \mathbb{N} \to \mathbb{N}$ *if and only if*

$$g(n) = \sup_{v \in V} \big|\{u \in V : d_G(v, u) \leq n\}\big|.$$

Example 3. The d-dimensional hypercubic grid over $V = \mathbb{Z}^d$ has polynomial growth $O(n^d)$. The infinite b-ary tree has exponential growth $\Omega(b^n)$. The complete graph with a countably infinite number of vertices does not have finite growth, as there are already infinitely many vertices at distance 1.

In general, no graph with unbounded degree has finite growth: the definition of growth requires $g(1) > \deg v$ for all $v \in V$, which is impossible since there are vertices with arbitrarily large degree.

The geometry of finite-dimensional Euclidean spaces forces any embedded graph to have at most polynomial growth.

Lemma 1. *Let the graph $G = (V, E)$ be bi-Lipschitz embeddable into the Euclidean space $\mathbb{R}^d$. Then G has polynomial growth $O(n^d)$.*

Proof. Let $\varphi\colon V \to \mathbb{R}^d$ be a bi-Lipschitz embedding with constants α and β, and let $g(n)$ be the growth function of G. For all pairs $u, v \in V$ of adjacent vertices, we have

$$\alpha = \alpha \times d_G(u, v) \leq \|\varphi(u) - \varphi(v)\|.$$

As a consequence, each vertex is the centre of a d-dimensional ball of radius α whose *interior* does not contain any other vertex. By halving this radius, we can surround each vertex with a d-dimensional ball of radius $\frac{\alpha}{2}$ whose interior is disjoint from any other ball. Each of these balls has volume $c\big(\frac{\alpha}{2}\big)^d$, where c is a constant depending only on d [12]. The maximum distance of the images in $\mathbb{R}^d$ of two vertices $u, v \in V$ with $d_G(u, v) = n$ is

$$\|\varphi(u) - \varphi(v)\| \leq \beta \times d_G(u, v) = \beta n.$$

The images in $\mathbb{R}^d$ of vertices having graph distance at most n from a given vertex are thus all contained in a d-dimensional ball of radius βn; the radius becomes $\beta n + \frac{\alpha}{2}$ when also considering the surrounding balls of radius $\frac{\alpha}{2}$; its total volume is thus $c\big(\beta n + \frac{\alpha}{2}\big)^d$. Since the large ball entirely contains the small ones, the inequality

$$c\Big(\beta n + \frac{\alpha}{2}\Big)^d \geq g(n) \times c\Big(\frac{\alpha}{2}\Big)^d$$

must hold. This shows that $g(n) \in O(n^d)$. □

Notice that the upper bound of Lemma 1 is tight since, by Example 3, for all $d \in \mathbb{N}$ there exist graphs with growth $\Theta(n^d)$ which are embeddable in $\mathbb{R}^d$.

In particular, Lemma 1 shows that "realistic" graphs, that is, those that can be embedded into the Euclidean space $\mathbb{R}^3$, have at most cubic growth.

Example 4. The infinite b-ary tree has exponential growth $\Omega(b^n)$, and as a consequence it cannot be bi-Lipschitz embedded into any finite-dimensional Euclidean space, for any $b \geq 2$.

4 Tissue P systems over realistic spatial graphs

We want to define a notion of (semi-)uniformity for families $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ of tissue P systems over spatial graphs. If we only require the mapping $x \mapsto \Pi_x$ to

be computable in polynomial time, we might still be able to "cheat" by encoding the solution of the problem under consideration in the spatial graph itself. We thus limit spatial graphs to those with a local structure that is easy to compute.

Definition 6. *A graph $G = (V, E)$ is said to be* polynomial-time navigable *if and only if there exists a deterministic Turing machine M running in polynomial time such that, for all $v \in V$ and $k \in \mathbb{N}$, the output of $M(v, k)$ is the k-th vertex adjacent to v (under some fixed ordering of such set); the machine M rejects if v does not have a k-th adjacent vertex (i.e., if k is too large).*

Since a polynomial-time navigable graph is defined by the Turing machine M computing the adjacency lists of its vertices, formally these graphs are always defined over the set of vertices $V = \Sigma^\star$, where Σ is the input alphabet of M. However, we can always identify those strings with the elements of another countable set (such as $\mathbb{N}$ or $\mathbb{Z}^d$) by means of an encoding. We also assume that the input integer k is given in binary notation[3].

Example 5. The bidimensional square grid over $V = \mathbb{Z}^2$ is navigable in polynomial time via

$$M\big((i,j),k\big) = \begin{cases} (i, j+1) & \text{if } k = 0 \\ (i+1, j) & \text{if } k = 1 \\ (i, j-1) & \text{if } k = 2 \\ (i-1, j) & \text{if } k = 3 \\ \perp & \text{otherwise} \end{cases}$$

The d-dimensional hypercubic grid over $V = \mathbb{Z}^d$ can be analogously proved to be polynomial-time navigable for all $d \in \mathbb{N}$.

Example 6. Although non-embeddable in any finite-dimensional Euclidean space, the complete graph with a countably infinite number of vertices is also polynomial-time navigable: assuming $V = \mathbb{N}$, define $M(v, k) = k$. This shows that the ability of standard tissue P systems to solve intractable problems in polynomial time is not due to the complexity of navigating the space, but rather to its shape, which allows arbitrarily many cells to divide or separate at the same time while remaining adjacent in order to perform communication.

Example 7. The infinite binary tree over $V = \mathbb{N}$ with root 0 can be navigated in polynomial time by letting $M(v, 0) = 2v$, $M(v, 1) = 2v + 1$, $M(v, 2) = \left\lfloor \frac{v}{2} \right\rfloor$, and $M(v, k) = \perp$ otherwise. In general, the infinite b-ary tree is navigated in polynomial time by $M(v, k) = bv + k$ for $k < b$, $M(v, b) = \left\lfloor \frac{v}{b} \right\rfloor$, and $M(v, k) = \perp$ otherwise.

[3] Since all graphs embeddable in a finite-dimensional Euclidean space have bounded degree (namely, bounded by $g(1)$), there is a maximum value of k for which the machine M produces output, independent of the vertex under consideration. As a consequence, choosing a binary or unary encoding for k makes no difference for the results of this paper.

We now give a generalised definition of semi-uniformity for tissue P systems over a spatial graph.

Definition 7. *A family of tissue P systems $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ over a connected spatial graph $G = (V, E)$ is said to be (polynomial-time)* semi-uniform *if it is semi-uniform in the traditional sense, with the additional requirement that the deterministic Turing machine computing the mapping $x \mapsto \Pi_x$ (see Definition 2) must also output the vertices where the cells are located in the initial configuration; we also require G to be polynomial-time navigable.*

Our main technical result shows that polynomial-growth spatial graphs severely hamper the efficiency of tissue P systems as a function of their running time.

Theorem 2. *A semi-uniform family $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ of confluent tissue P systems over a polynomial-time navigable spatial graph $G = (V, E)$ with polynomial growth can be simulated by a deterministic Turing machine with a polynomial slowdown.*

Proof. Given an input string $x \in \Sigma^\star$, a deterministic Turing machine M can simulate the machine H providing the semi-uniformity condition and obtain, in polynomial time, the description of the tissue P system Π_x. Let $V_0 \subseteq V$ be the set of vertices where the cells of the initial configuration $\mathcal{C}_0$ of Π_x are located, and let $G_0 = (V_0, E_0)$ be the subgraph of G induced by V_0. Since G has polynomial growth $g(n)$, the set of edges E_0 can be computed in polynomial time by repeatedly querying the Turing machine navigating G on all pairs (v, k) with $v \in V_0$ and $0 \leq k < g(1)$.

Suppose t computation steps of the tissue P system have been simulated, and M stores an explicit description of the configuration $\mathcal{C}_t$ of Π_x, including the subgraph $G_t = (V_t, E_t)$ induced by the vertices V_t where the cells are located. The machine M can now iterate across all rules $r \in R$ of the tissue P system, in any given order, checking whether the rule is applicable and updating the configuration as needed. Recall that each cell can either apply communication rules (as many as possible), or a single cell division or cell separation rule.

If r is a communication rule $[w_1]_h \leftrightarrow [w_2]_k$, then M iterates across all pairs (h_i, k_j) of cells having label h and k, respectively[4]; for each such pair, M checks whether the corresponding vertices are connected by an edge in E_t and, if so, M applies the communication rules as many times as possible (depending on the multisets contained in h_i and k_j) updating the configuration accordingly.

If r is a communication rule $[w_1]_h \leftrightarrow w_2$ involving the environment, then M only needs to iterate over all cells h_i having label h, as the adjacency to the environment can always be assumed.

If r is a cell division rule $[a]_h \to [b]_h\ [c]_h$ or a separation rule $[a]_h \to [\Gamma_1]_h\ [\Gamma_2]_h$, then M iterates across all cells h_i having label h; for each such cell, M checks

[4] Here i and j represent identifiers used to distinguish cells having the same label. A suitable identifier schema has been proposed by Sosík and Cienciala [13].

whether the cell contains the required instance of a and, if so, whether there exists an empty vertex v adjacent to the vertex where h_i is located; these vertices can be enumerated in polynomial time, due to the polynomial growth of G. If an empty vertex is found, the graph G_t is updated by adding v to V_t and (u, v) to E_t for all vertices $u \in V_t$ such that $(u, v) \in E$; this establishes the possible adjacency of the newly occupied vertex to previous vertices in V_t. Another instance of cell h is then created at vertex v, and the contents of h_i are either replicated or separated, with any necessary update, according to the semantics of rule r.

After having iterated across all rules $r \in R$, the machine M has computed an updated configuration $\mathcal{C}_{t+1}$ of Π_x, including and updated graph G_{t+1}. The rules during the simulated computation step $\mathcal{C}_t \to \mathcal{C}_{t+1}$ have been applied in a maximally parallel way in one of the possible (but equally valid, due to confluence) nondeterministic ways. If the halting conditions are met, i.e., the object *yes* or *no* has been sent out to the environment, then the simulation can halt by accepting or rejecting accordingly; otherwise, another computation step of Π_x is simulated.

The initial number of cells of Π_x is bounded by a polynomial $p(n)$, and can only increase up to $p(n)g(t)$ after t computation steps due to the shape of the spatial graph. Indeed, division and separation can only happen when there is an empty vertex nearby; since the growth of the spatial graph is polynomial, only a polynomial number of empty vertices (with respect to t) are adjacent to cells at each time step. Hence, the total number of cells is polynomial with respect to time. The multisets of objects contained inside the cells can instead increase exponentially with time, but the multiplicities can be stored in binary, requiring a number of bits increasing polynomially with time, and only polynomial time to apply each communication rule in a maximally parallel way. As a consequence, the simulation is carried out with a polynomial slowdown. □

As a corollary, the sort of P systems that we may consider "realistic" cannot solve intractable problems in polynomial time.

Corollary 1. *The class of problems solved by semi-uniform families of confluent tissue P systems over spatial graphs embeddable in finite-dimensional Euclidean spaces is included in* **P**. □

5 Conclusions

We showed that the ability of standard tissue P systems with cell division or cell separation to solve **NP**-complete problems in polynomial time rests on the assumption that these systems can increase their volume exponentially in polynomial time while maintaining full communication capability. That growth is actually possible only if the space in which the P system is located can actually sustain it.

We have proved that, when embedding the tissue P systems into the Euclidean space, only polynomial growth is possible. As a consequence, tissue P systems over locally-easy-to-compute graphs which can be embedded in Euclidean spaces can solve at most problems in **P** when working in polynomial time. Under

the assumption that polynomial-time navigability and embeddability into the Euclidean space are necessary for a spatial graph to be realistic, we can conclude that realistic tissue P systems are actually no more efficient than deterministic Turing machines.

As a future research topic, it might be interesting to consider the computing power of families of tissue P systems over more general spatial graphs, dropping the polynomial-time navigability or the polynomial growth requirements. In particular, can a purposefully designed spatial graph allow tissue P systems to break the $\mathbf{P}^{\#\mathbf{P}}$ barrier of the standard model [6]?

References

1. Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Pardini, G.: Simulation of spatial P system models. Theoretical Computer Science 529, 11–45 (2014)
2. Bianconi, E., Piovesan, A., Facchin, F., Beraudi, A., Casadei, R., Frabetti, F., Vitale, L., Pelleri, M.C., Tassani, S., Piva, F., Perez-Amodio, S., Strippoli, P., Canaider, S.: An estimation of the number of cells in the human body. Annals of Human Biology 40(6), 463–471 (2013)
3. Csuhaj-Varjú, E., Gheorghe, M., Stannett, M., Vaszil, G.: Spatially localised membrane systems. Fundamenta Informaticae 128(1–2), 193–205 (2014)
4. Deza, M.M., Deza, E.: Encyclopedia of Distances, Third Edition. Springer (2014)
5. Díaz-Pernil, D., Pérez-Jiménez, M.J., Romero-Jiménez, A.: Efficient simulation of tissue-like P systems by transition cell-like P systems. Natural Computing 8(4), 797–806 (2009)
6. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Tissue P systems can be simulated efficiently with counting oracles. In: Proceedings of the 16th International Conference on Membrane Computing (CMC 2015) (2015), to appear
7. Margenstern, M.: Can hyperbolic geometry be of help for P systems? In: Martín-Vide, Carlos Mauri, G., Păn, Gh., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing, International Workshop, WMC 2003. Lecture Notes in Computer Science, vol. 2933, pp. 240–249 (2004)
8. Martín-Vide, C., Păun, Gh., Pazos, J., Rodríguez-Patón, A.: Tissue P systems. Theoretical Computer Science 296(2), 295–326 (2003)
9. Murphy, N., Woods, D.: The computational power of membrane systems under tight uniformity conditions. Natural Computing 10(1), 613–632 (2011)
10. Pan, L., Pérez-Jiménez, M.J.: Computational complexity of tissue-like P systems. Journal of Complexity 26(3), 296–315 (2010)
11. Păun, Gh., Pérez-Jiménez, M.J., Riscos Núñez, A.: Tissue P systems with cell division. International Journal of Computers, Communications & Control 3(3), 295–303 (2008)
12. Sommerville, D.M.Y.: An Introduction to the Geometry of n Dimensions. Methuen (1929)
13. Sosík, P., Cienciala, L.: A limitation of cell division in tissue P systems by PSPACE. Journal of Computer and System Sciences 81(2), 473–484 (2015)