Geometrical Membrane Systems

authors

No Institute Given

Abstract. Geometrical P systems ...

1 Introduction

SOLID INTRO HERE...

2 Geometric P systems

The motivation for introducing geometrical P systems is twofold. Firstly, we believe that the potential of the idea of membrane computing remains unexploited to a large extent when restricted to amorphous cells/regions counting atomic (amorphous) objects. While its simplicity is attractive from the mathematical point of view, it can be also limiting, both when treated as a bio-inspired computing model, and when used to model biological processes. Note that the style of computing of most variants of P systems seems to be restricted to a counter machine capable of addition and subtraction. Note also that geometrical arrangements and self-assembly processes are believed to be at the very core of evolution of life on the Earth, and that certain internal organization is essential for each living cell.

Secondly, we also aim towards a model extending our understanding of computation beyond the scope of traditional computer science. Besides being able to compute in the Turing sense, the model should, e.g., intensively interact with and "sense" its physical environment, to be capable of self-modification and to evolve unenthropically, i.e., to increase its fitness (however defined) in the environment it is embedded within. Membrane systems seem to be a good candidate, but a sufficient level of self-modification and evolution of new features is hardly possible on the amorphous level.

The geometrical P system is freely inspired by P systems with symport/antiport [2] and P systems with proteins on membranes [3, 4], but it is extended with explicit geometrical features and self-assembly capabilities. Recall that a *P system with proteins on membranes* consists of a structure of (possibly nested) *membranes* which delimit *regions* precisely identified by the membranes. In addition, the system contains:

– *Proteins* which are placed on membranes. They act as protein channels, allowing some objects pass through the membrane, or catalyzing reactions of objects. Proteins can never change their place. Generally, they can change

their type during their interactions with other objects, but in this paper we assume they never change.
- *Objects* placed inside membranes or in the surrounding environment. They can pass through protein channels and participate in reactions catalyzed by proteins. Objects do not bear any information about their location and/or shape. Several copies of the same object can be present in a region, so we work with *multisets* of objects.

The geometrical P system extends the above concept as follows:

- The whole system is embedded in an n-dimensional Euclidean space.
- Objects of the type described above are referred to as *floating objects*.
- In addition, there are *fixed objects* with pre-defined shape and with fixed positions in space, possibly forming interconnected structures during the system evolution. Examples of fixed objects can be *tiles* and *tubules*.
- Fixed objects can mutually attach due to a given *glue relation*, as in tiling systems.
- Proteins are placed on fixed objects and, apart from acting as protein channels, they can also play the role of glues connecting fixed objects.
- Fixed objects can be created or destroyed in a reaction with given (multisets of) floating objects.
- Fixed objects thus define an *interconnection graph*, each vertex being labelled by one fixed object in place, and each arc labelled by a protein bond connecting two objects;
- A membrane is not just an abstract container of objects, but it is formed of fixed objects (tiles) and, therefore, it has its defined size and shape.

Cell division (mitosis) is an important part of the model, considered already in [4]. Here, however, we define it in the bottom-up manner, so as:

- to emerge from many local interactions;
- to be initiated by the cytoskeleton dynamics;
- to use standard growth mechanisms to complete the two parts of the cell after their division.

2.1 Preliminaries

We recall some notation which is standard in the area of membrane computing. The reader is referred to [6] for an introduction and overview of membrane systems, and to [8] for recent development.

For a multiset M we denote by $|M|_a$ the multiplicity of objects a in M. A multiset M with the underlying set O can be represented by a string $x \in O^*$ (by O^* we denote the free monoid generated by O with respect to the concatenation and the identity λ) such that the number of occurrences of $a \in O$ in x represents the value $|M|_a$. For a set A we denote by $\mathcal{M}(A)$ the set of all multisets with elements from A.

Let further $\mathbb{R}^n$ denote the n-dimensional Euclidean space.

2.2 Definition

Definition 1. *A* geometrical P system *in* $\mathbb{R}^n$ *is a tuple*

$$\Pi = (O, P, Q, E, r_{int}, G, R, S),$$

where

O *is the set of floating objects;*

$P = P_G \cup P_R$ *is the set of proteins, consisting of two mutually disjoint subsets:* glue proteins P_G *and* reaction proteins P_R;

Q *is the set of fixed objects, with* Q, O, P *all mutually disjoint.*
Each fixed object $s \in Q$ *is described by its vertices* $\mathbf{x}_1, \ldots, \mathbf{x}_m$, *a set of* connectors $\{c_1, \ldots, c_k\}$ *determining its connectivity to other objects, and a set of proteins on* s. *The shape of* s *is defined as the convex hull of its vertices. Formally:*

$$s = ((\mathbf{v}_1, \ldots \mathbf{v}_m), \{c_1, \ldots, c_k\}, \mu, p_s)$$

for some $k \geq 0$, $m \geq 2$, *where*
- $\mathbf{v}_i \in \mathbb{R}^n$, $1 \leq i \leq m$, *are coordinates of vertices relative to a reference point* $\boldsymbol{0}$ *located within* s;
- $c_1, \ldots, c_k$ *are connectors;*
- $\mu \in P_R^*$ *is the multiset of metabolic proteins on* s;
- $p_s \in P_G$ *is the surface protein of* s.

Each connector adopts the form $c = ((\mathbf{x}_1, \ldots, \mathbf{x}_l), p, \varphi)$, *where*
- $\mathbf{x}_i \in \mathbb{R}^n$, $1 \leq i \leq l$, *are connector coordinates relative to a reference point* $\boldsymbol{0}$ *located within* s; *the connector can be point-shaped (*$l = 1$*), edge-shaped (*$l = 2$*) or face-shaped (*$l \geq 3$*);*
- $p \in P_G$ *is a glue protein;*
- $\varphi \in \langle 0, 2\pi)$ *is the connection angle of* c.

$E \subseteq O$ *is the set of environmental objects present in the outer environment in an unlimited number of copies;*

r_{int} *is the radius of protein-protein interactions;*

$G \subseteq (P_G \times P_G \times O^*)$ *is the glue relation: if* $(p_1, p_2, u) \in G$, *then the glue proteins* p_1 *and* p_2 *within the distance* r_{int} *can bind together, and a "signal" multiset* u *of floating objects is produced. Note that* G *is generally a non-symmetric relation with respect to* p_1 *and* p_2.

R *is a finite set of evolution rules of the geometrical P system;*

$S \subset (Q \times \mathbb{R}^n \times \langle 0, 2\pi \rangle^n)$ *is the set of initial objects: each pair* $(s, \mathbf{x}, \varphi) \in S$ *denotes a fixed object* s *with its center at position* $\mathbf{x} \in \mathbb{R}^n$ *in the (Euclidean) environment, and rotated by the angles* $\varphi \in \langle 0, 2\pi)^n$ *along coordinate axes.*

Note that, unlike many models of P systems, there is no pre-defined membrane structure. Instead, such a structure can be gradually formed from specific fixed objects (referred to as *tiles*) by the self-assembly process controlled by the rules creating fixed objects, and the glue relation.

An *evolution rule* from the set R can be applied when the multiset of objects (of either type) on the left-hand side of the rule is present. All these object

must be in the same region, either one enclosed by a membrane or the surrounding environment. Furthermore, application of some rules is subject to another condition.

The application of a rule results in producing the multiset of objects at its right-hand side in the same region, unless the rule states otherwise. The objects at the left-hand side are consumed.

Metabolic rules

The metabolic rules allow reactions of floating objects or their interchange between membranes and the outer environment. The first two types of rules (simple and catalyzed) describing reactions of floating objects correspond to cooperative and catalytic rules in membrane systems with objects [5, 6]. The symport and antiport rules are inspired by [2] but their notation and style of application is due to P systems with proteins on membranes [3, 4].

Type	Rule	Effect
simple	$u \to v$	objects in u react to produce v
catalytic	$[p\|u] \to [p\|v]$	the same but p must be present
symport	$u[p\| \ ] \to [p\|u]$ $[p\|u] \to u[p\| \ ]$	u passes a membrane
antiport	$u[p\|v] \to v[p\|u]$	interchange u and v

In all these rules, $u, v \in O^+$ are nonempty multisets of floating objects and p, q are proteins. The symport and antiport rules can be applied only when the protein p is located on a 2D or 3D object. In the former case the objects in u (resp. v) pass to the other side of the 2D object possibly forming a part of a membrane. In the latter case the objects u/v pass inside/outside the 3D object.

Rules creating fixed objects $u \to t$,

where $t \in Q$ and $u \in O^+$. These rules create fixed objects and attach them to other fixed objects via protein bonds. A protein participating in a protein bond is called *bound*, otherwise it is called *free.*

To apply the rule, t must be able to connect to an existing fixed object. Formally, t must contain a glue protein q, and a fixed object s must exist in the same region with a free protein p, such that $(p, q, v) \in G$ for some $v \in O^*$. The fixed object t is created and attached by q to p, while producing also the "signal" objects in v.

If any part of t would intersect an existing fixed object, then the growth is blocked and t cannot attach to s, unless stated otherwise.

If t bears other glue proteins which, as a result of attachment of t to s, are now placed within the distance r_{int} to free proteins on s or other fixed objects,

and if these protein pairs can glue due to the relation G, they immediately also bind together.

The role of connectors as vertices of fixed objects may suggest that protein bonds always connect one vertex to another, in accordance with the glue relation. This is indeed one of the options. Note, however, that the definition does not specify the connecting mechanism. Consider, e.g., two-dimensional objects called *tiles*. Glue proteins can be related to tile edges and the tiles can connect edge-to-edge, as it is usual in tiling systems. Generally, the connecting mechanism can differ for various types of objects.

Rules destroying fixed objects $us \to v$,

where $s \in Q$, $u, v \in O^+$. In the presence of the "destructor" multiset of floating objects u, the existing fixed object s is destroyed. All the glue proteins on objects attached to s are set free. The destruction consumes the objects in u and produces the multiset v of "waste" objects. This may result in a gap in the existing structure of fixed objects, which can be later filled by another fixed object s' with corresponding glues.

Division (mitosis) rules $p - q \mid u \to p \mid q$,

where $p - q$ is a pair of bound glue proteins whose bond is broken in the presence of the multiset of floating objects $u \in O^+$, which is consumed. These rules support the process of mitosis (cell division).

After breaking their bond, the glue proteins p and q remain inactive (unable to bind again) until the process of mitosis is finished. This occurs when fixed objects (typically tiles) on which p and q are placed, are completely disconnected, i.e., all paths between them in the connection graph of Π are cut.

Denote the two fixed object containing p (resp. q) by s (resp. t). When s and t are disconnected, the structure containing them splits into two parts. Each part consists of s (or t, respectively) and all the fixed objects still connected to it via the connection graph of Π. If the broken bonds caused the isolation of other fixed objects unconnected to either s or t, they are lost.

Using the rules creating fixed objects (described above) and their self-assembly mechanism, both divided fragments can be subject to further growth, eventually forming two new cells. We assume, for simplicity, that the multiset of floating objects present in the cell is divided into two almost equal (rounded) parts, and each stays together with the s and the t fragment, respectively, even if these fragments are temporarily open to the outer environment, until new membranes are completed around them.

The self-healing property If more sites with glue proteins are available for the growth of a new fixed object s, then the site which would establish the maximum number of protein bonds with s is preferred (has a higher probability to be chosen).

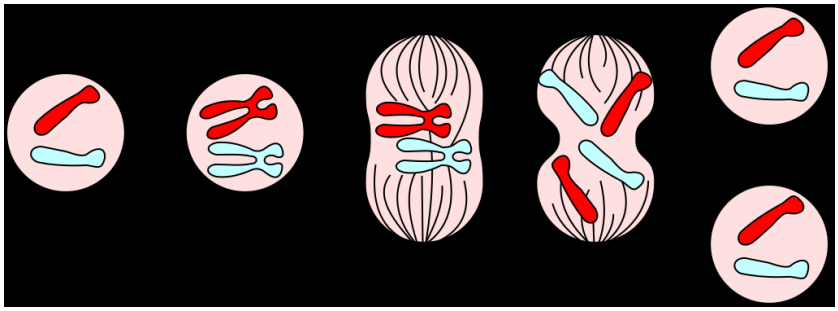


Fig. 1. Major events in mitosis. Licensed under Wikimedia Commons, authored by Mysid.

This establishes the self-healing property: assume that a fixed object s within an existing structure (membrane or cytoskeleton) was destroyed, leaving a gap with two or more free glue proteins around. When a new object s is created, it would more probably fill this gap than extend a growing structure where it would bind to a lower number of glue proteins.

Configuration of the system is determined by the

- interconnection graph of fixed objects present in the system;
- positions and angles of all fixed objects;
- multiset of floating objects within all closed regions delimited by membranes (composed of tiles).

The initial configuration is given by a discrete graph whose vertices are the objects in S, together with their positions and angles.

The system passes between configuration by application of rules in the set R. These rules are applied in discrete steps following the *maximal parallelism* principle: an applicable multiset of rules is non-deterministically chosen so that no more rule can be added to it. Each object can participate in at most one rule at each step. However, when proteins on a fixed objects act in some rules, the fixed object itself is not involved and it can simultaneously act in another (for instance erasing) rule. All the rules in the chosen maximum applicable multiset are applied in parallel.

3 An example: the cytoskeleton and the mitosis

We give an example of a geometrical P system modelling the growth of a cellular membrane, then the growth of cytoskeleton within the membrane, and finally the process of cell mitosis. Unlike many other models of P systems which assume cell division as an atomic operation, here we fully rely on local interaction

of fixed objects (2D tiles, 1D tubules) forming the membrane and the cytoskeleton, and controlling the growth and the mitosis processes. In accordance with the introductory section, consider the ability of self-reproduction of elementary cells, building simultaneously their internal cytoskeleton structure, be the goal of computation in this example.

Consider a geometrical P systems in three-dimensional space $\mathbb{R}^3$. To simplify the description (while following Definition 1), let the set of fixed objects Q be divided into two mutually disjoint subsets:

Q_1 is the set of pentagonal *tiles* (elements forming membranes) in $\mathbb{R}^2$. Let $c_1, \ldots, c_5$ be its edge connectors, each being a triple $c_i = ((\mathbf{x}_i, \mathbf{x}_{i \bmod 5+1}), p_i, \varphi_i)$ (as in Definition 1), corresponding to the edges of the pentagon. Note that φ_i is the angle between s and another tile which would eventually attach its edge to s.
The remaining point connectors $c_6, \ldots, c_m$ are placed on the tile and one-dimensional fixed objects (tubules) can connect to them.

Q_2 is the set of *tubules*, each geometrically represented as a straight segment in $\mathbb{R}^1$. Each of its point connectors is placed on either end of the tubule.

Let a fixed object growth rule $u \to t$ be applied, with $t \in Q$ and $u \in O^+$. Let t contain a connector with glue protein q, and let there be a fixed object s with a free glue protein p on its connector such that $(p, q, v) \in G$ for some $v \in O^*$. Let φ be the connecting angle of the p-connector. To attach t to s, the following gluing conditions apply:

(a) If both s and t are tiles in Q_1 :
Both p and q must be edge proteins specified by connectors as above. Edges with p and q glue together with the angle φ between s and t.

(b) If s is a tile or a tubule and $t \in Q_2$ is a tubule:
Connect q to p at the site of p and position t randomly so that the angle between s and t is φ.

When a growing tubule t would intersect an existing tile, and the protein on the tip of t can glue to the surface protein of the tile, then t is shortened and its tip connector is attached to the tile due to the glue relation.

Priorities among rules Some models of P systems use a priority relation among rules, motivated by different propensities of reactions. If the induced ordering of rules is complete, a deterministic P system results [1]. Usually an incomplete ordering of rules is permitted, dividing the rules into prioritized groups. Rules inside each group are chosen non-deterministically until no applicable rule remain, and then they are all applied in parallel.

In this example, metabolic rules have *higher priority* than the other types of rules. I.e., first the maximal multiset of metabolic rules is applied, and then the unused floating objects can be subject to other rules. This allows to regulate the fixed structure behavior (growth, destruction, division) by metabolic processes, and a dynamical homeostasis can be easily established.

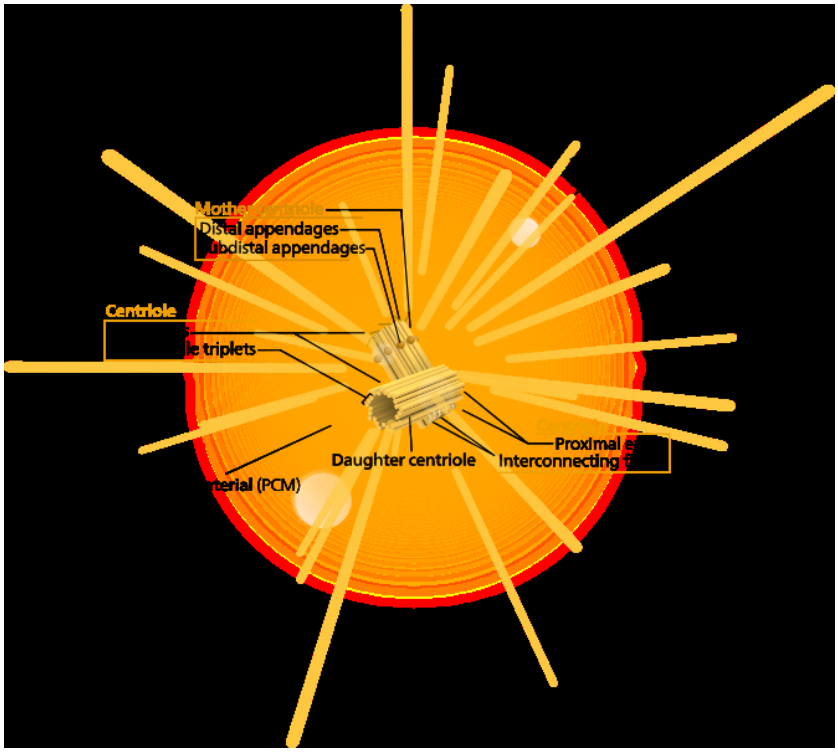


Fig. 2. The structure of the centrosome formed by two centrioles. Source: Wikimedia Commons, author Kelvin Song

3.1 Formal description

Consider a geometrical P system in 3-dimensional space

$$\Pi = (O, P, Q, E, r_{int}, G, R, S),$$

where

$O = \{a, b, c, d, e, f\}$
$P = P_G \cup P_R$ with $P_R = \{p_1, p_2, p_3, p_4\}$ and $P_G = \{p_a, p_b, p_c, p'_a, p'_b, p'_c, p_d, p_e, p_t, p_x\}$;
$Q = Q_1 \cup Q_2$ with

$Q_1 = \{q_1, q_2, q_3, q_4\}$, where q_1, q_2 are larger tiles forming the outer cell membrane, and q_3, q_4 are smaller tiles forming nuclear membrane. All are shaped as regular pentagons.
Tile q_1 contains a connector starting the cytoskeleton growth with protein p_d, and q_2 contains p_1 serving as membrane channel. Formally:

$$\begin{array}{ll} q_1 = ((\mathbf{x}_i \mid 1 \le i \le 5), & \textit{(vertices)} \\ \{((\mathbf{x}_i, \mathbf{x}_{i \bmod 5+1}), p_a, \varphi_p) \mid 1 \le i \le 5\} & \textit{(edge connectors)} \\ \cup \{(\mathbf{0}, p_d, \pi/2)\}, & \textit{(point connector)} \\ p_1 p_4, p_x), & \textit{(surface protein)} \end{array}$$

where $\mathbf{x}_i = 10 \cdot (\cos(\frac{2}{5}\pi(i-1)),\ \sin(\frac{2}{5}\pi(i-1)))$, $1 \le i \le 5$ are vertices of a pentagon, and $\varphi_p = 2.0345$ is the inner angle between two sides in dodecahedron.

$$\begin{array}{l} q_2 = ((\mathbf{x}_i \mid 1 \le i \le 5), \\ \quad \{((\mathbf{x}_1, \mathbf{x}_2), p_a, \varphi_p), ((\mathbf{x}_2, \mathbf{x}_3), p_b, \varphi_p), ((\mathbf{x}_3, \mathbf{x}_4), p_c, \varphi_p), ((\mathbf{x}_4, \mathbf{x}_5), p_c, \varphi_p), \\ \quad\ ((\mathbf{x}_5, \mathbf{x}_1), p_b, \varphi_p)\}, \\ \quad p_1 p_4, p_x), \end{array}$$

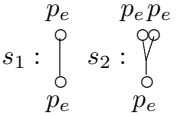
$$\begin{array}{l} q_3 = ((\mathbf{x}'_i \mid 1 \le i \le 5), \\ \quad \{((\mathbf{x}'_i, \mathbf{x}'_{i \bmod 5+1}), p'_a, \varphi_p) \mid 1 \le i \le 5\}, \\ \quad p_3, p_t), \end{array}$$

$$\begin{array}{l} q_4 = ((\mathbf{x}'_i \mid 1 \le i \le 5), \\ \quad \{((\mathbf{x}'_1, \mathbf{x}'_2), p_a, \varphi_p), ((\mathbf{x}'_2, \mathbf{x}'_3), p_b, \varphi_p), ((\mathbf{x}'_3, \mathbf{x}'_4), p_c, \varphi_p), ((\mathbf{x}'_4, \mathbf{x}'_5), p_c, \varphi_p), \\ \quad\ ((\mathbf{x}'_5, \mathbf{x}'_1), p_b, \varphi_p)\}, \\ \quad p_3, p_x), \end{array}$$

where $\mathbf{x}'_i = \mathbf{x}_i/5,\ 1 \le i \le 5$.

$Q_2 = \{s_1, s_2\}$, the set of elementary tubules, containing

– tubule with vertices 0 and 2, with a connector at each end, and with reaction protein p_2 :
$s_1 = ((0, 2), \{(0, p_e, 0), (2, p_e, 0)\}, p_2, p_x)$;



– tubule with vertices 0 and 2, with three fork-arranged connectors, and with reaction protein p_2 :
$s_2 = ((0, 2), \{(0, p_e, 0), (2, p_e, \pi/20), (2, p_e, -\pi/20)\}, p_2, p_x)$;

$E = \{a\}$
$r = 0.1$ is the radius of protein interaction.
$G = \{(p_d, p_e, \lambda), (p_e, p_e, \lambda), (p_e, p_t, f)\} \cup \{(p, p, \lambda) \mid p \in \{p_a, p_b, p_c, p'_a, p'_b, p'_c\}\}$ defines the glue relation;
R contains the following rules:

Symport/antiport rules for transport through the membrane:
$a[p_1|\ ] \to [p_1|a]$;
$a[p_1|d] \to d[p_1|a]$;
$e[p_1|d] \to d[p_1|e]$;

Catalyzed rules:
$[p_2|c] \to [p_2|b]$;
$[p_3|b] \to [p_3|d]$;
$[p_3|b] \to [p_3|d]$;
$[p_4|a] \to [p_4|c]$;

Rules creating fixed objects:
$aaa \to q_1$; (outer membrane tiles creation from three objects a)
$aaa \to q_2$;
$b^{12} \to q_3$; (nuclear membrane tile creation from 12 objects b)
$b^{12} \to q_4$;

$c \to s_1$; (tubule creation)
$ccc \to s_2$; (tubule creation)

Division rules:
$p_c - p_c|ff \to p_c|p_c)$; (division of the outer membrane)
$p'_c - p'_c|ff \to p'_c|p'_c$; (division of the nuclear membrane)
$S = \{(q_1, \mathbf{x}_0), (q_3, 0.2 \cdot \mathbf{x}_0)\}$, where

$$\mathbf{x}_0 = (0, 0, -5\sqrt{6 + 2\sqrt{5}})$$

is the distance between the center of the dodecahedron and the center of any of its faces.

The initial configuration contains the two seed tiles q_1 and q_3. The sequence of evolution steps of the system Π will be the following:

1. In the first step, five outer membrane tiles q_2 are created by the rule $aaa \to q_2$, and attached to five connecting proteins p_a of the seed tile q_1. Their other proteins p_b at matching positions attach together, too.
2. In the second step, five more tiles q_2 are attached to the existing construction.
3. In the third step, the last pentagonal tile q_1 is attached, closing the dodecahedron-shaped membrane ("soccer ball"). Its content is initially empty, by assumption.

4. During the previous two steps, the reaction proteins p_4 on tiles already have changed some objects a to c applying the rule $[p_4|a] \rightarrow [p_4|c]$. From now on, proteins p_1 on tiles let pass 12 objects a inside the cell in every step, applying the rule $a[p_1|\,] \rightarrow [p_1|a]$. These are subsequently changed to c's.
5. In the fourth step, assume that one object c is consumed by the rule $[c \rightarrow s_1]$, producing the tubule s_1 and connecting it to the single connector p_d on tile q_1. (Alternatively, tubule s_2 can be produced and connected, consuming three c's.)
6. There is one free protein p_e hence one new tubule s_1 or s_2 is produced and attached. Let it be s_2 this time.
7. There are two free proteins p_e, so that two new tubules s_1 or s_2 are created and attached to the growing cytoskeleton structure. Simultaneously, the protein p_2 on existing tubule acts in the rule $[p_2|c] \rightarrow [p_2|b]$, changing one object c to b (an analogy of a metabolic process).
8. Step by step, the structure of tubules will grow, containing still more proteins p_2 Each of them will then act in the catalyzed rules $[p_2|c] \rightarrow [p_2|b]$. Whenever more than 12 of these proteins are present, the number of objects c in the cell starts to gradually decrease, as they are consumed faster then introduced from the environment. When there remain no more extra objects c, the cytoskeleton growth will stop and the membrane will be filled with growing number of b's.
9. Since the P system is non-deterministic, many variants of cytoskeleton growth are possible. The growth example at Fig. 3 corresponds to elements $s_1 s_2$ which then branches to sequences $s_1 s_1$ and $s_1 s_2 s_1 s_2$.
10. In the latter phase of the evolution, when enough b's are accumulated inside the cell, the formation of the nuclear membrane starts. Each dozen of b's can give rise to one nuclear membrane tile: $b^{12} \rightarrow q_4$, which can attach to the existing seed tile q_3. These tiles gradually form a similar (but smaller) dodecahedron as the outer membrane.
11. The tiles q_3 and q_4 contain reaction protein p_3 which catalyzes the rule $[p_3|b] \rightarrow [p_3|d]$. Hence, as the formation of the nuclear membrane progresses, it gets slower as more and more b's are changed to d's instead of being used to grow new tiles q_3.
12. The complete nuclear membrane has two tiles q_3 at its opposite faces, while the remaining faces are q_4's. A dynamical homeostasis is established:
    - at each step, 12 c's are introduced and, by the same rules, up to 12 d's are expelled from the cell;
    - simultaneously, 12 other c's already within the cell are changed to b's; and
    - 12 other b's within the cell are changed to d's.
13. When the cytoskeleton (growing simultaneously and randomly from both poles of the cell) reaches the nuclear membrane, the glue relation allows its attachment to proteins p_3 on its polar tiles. Each such contact releases a signal object f (as defined in the glue relation).
14. Each pair of objects f breaks one bonds $p_c - p_c$ or $p'_c - p'_c$ in the equatorial part of both membranes (connecting a pair of tiles q_2 or q'_2).

15. When all these bonds (10 in each membrane) are broken, the cell splits into two parts (a half of the outer membrane, a part of the cytoskeleton and the corresponding half of the nucleus). Floating objects are fairly divided between the two halves.
16. After the mitosis, using the same growth rules as before, both outer membrane fragments would be completed by new tiles. After their completion, both nuclei would be completed by the same mechanism. New cytoskeleton starts to grow from the new cellular membrane tiles and the whole process can repeat.

In such a way, an unlimited cell division is started. It can be regulated, for instance, by the limited amount of "food" (floating objects necessary to build new structures) in the environment.

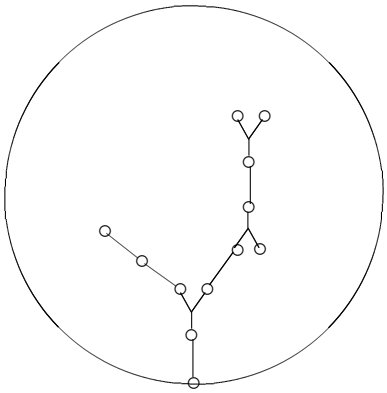


Fig. 3. A possible growth of tubules of the P system in Example 3.1

We abstract from certain details occurring in biological cells during the mitosis, as centrioles or contractions of the cytoskeleton. Notice that, as our cytoskeleton growth is random, the mitosis in the model can be sometimes blocked due to these reasons:

1. The growing cytoskeleton does not establish 20 contacts with the polar tiles s_3 of the nucleus until reaching the dynamical homeostasis and stopping the growth. Hence not all the bonds $p_c - p_c$ and $p_d - p_d$ are broken and the membrane structure still sticks together.
2. The cytoskeleton growing from one of the two outer membrane tiles s_1 would connect to *both* nuclear polar tiles s_3. Hence it prevents the nucleus to split into halves.

4 Simulation results

TO BE COMPLETED

5 Computational power and efficiency - UNFINISHED

In this section we demonstrate that geometrical P systems are capable of universal computation, in the Turing sense. Let us define the output of the system as a multiset of specific objects from $O - E$ in the environment when the system stops its computation. (An input can be defined analogously, as a special multiset present initially in the environment.)

Proposition 1. *Geometrical P systems can generate NRE.*

Proof. (Sketch) By a simulation of register machine, where each register would be represented by a specific type of object: first, a cell will be formed and then the computation starts. FULL PROOOF WILL BE HERE.

Concerning the efficiency of geometrical P systems due to computational complexity measures, we conjecture that, due to the existence of mitosis rules, it is possible to solve NP-hard problems in a polynomial time. Several models of P systems have been already proven to possess this capability, among them P systems with proteins on membranes and cell division. There are, however, some differences between that model and geometrical membrane P systems, so further research is necessary to verify the conjecture. For the time being this remains an *open problem.*

6 Conclusions

We have introduced the *geometrical P system* – computing model inspired by the cell biology, which extends the membrane computing approach with explicit geometrical arrangement of atomic objects. Basic abstract operations in the model include reactions among objects, their transport through protein channels, and their mutual interconnection, leading to construction and destruction of complex geometrical structures.

WHAT WAS THE MOTIVATION FOR THE MODEL, WHAT THE MODEL WAS EXPECTED TO EVENTUALLY PERFORM – TO BE FILLED IN BY MAX?

We have shown that geometrical P systems are computationally universal in the Turing sense. We have also demonstrated their capability to grow complex cell-inspired information processing structures, providing a model of the cytoskeleton growth which in turn controls the mitosis.

Further research directions include, on one hand, the study of computational efficiency of the model. On the other hand, we see as a crucial step forward to extend the model with evolutionary properties – the capability to evolve unenthropically towards more efficient behavior related to its specific goals, which can be of many kinds. To this end, the model should be equipped with a kind of

abstract genetic code defining shapes of fixed objects and placement of connectors and other proteins on them. Perhaps the evolution of new floating objects and proteins and their mutual reactions should be allowed, too, reflecting the evolution of new organic molecules. In this way we could step towards a model of self-modyfing bio-computing cell-inspired device, perhaps similar to some artificial life approaches [REFERENCES HERE]. The evolution, together with the embeddedness in a complex environment, could also increase the computing potential of the model beyond the Turing's limit [7].

WHAT ELSE?

References

1. Gheorghe, M., Manca, V., Romero-Campero, F.J.: Deterministic and stochastic p systems for modelling cellular processes. Natural Computing 9(2), 457–473 (2010), http://dx.doi.org/10.1007/s11047-009-9158-4
2. Paun, A., Paun, G.: The power of communication: P systems with symport/antiport. New Generation Computing 20(3), 295–305 (2002)
3. Păun, A., Popa, B.: P systems with proteins on membranes. Fundamenta Informaticae 72(4), 467–483 (2006)
4. Păun, A., Popa, B.: P systems with proteins on membranes and membrane division. In: Ibarra, O., Dang, Z. (eds.) DLT 2006. Lecture Notes in Computer Science, vol. 4036, pp. 292–303. Springer, Berlin (2006)
5. Păun, G.: Computing with membranes. J. Comput. System Sci. 61, 108–143 (2000)
6. Păun, G.: Membrane Computing – An Introduction. Springer, Berlin (2002)
7. Van Leeuwen, J., Wiedermann, J.: Beyond the turing limit: Evolving interactive systems. Lecture Notes in Computer Science 2234, 90–109 (2001)
8. The P Systems Web Page, `http://ppage.psystems.eu/`, [cit. 2012-5-29]