On Self-Assembly in Population P Systems

Francesco Bernardini[1], Marian Gheorghe[1], Natalio Krasnogor[2], and
Jean-Louis Giavitto[3]

[1] Department of Computer Science, The University of Sheffield
Regent Court, Portobello Street, Sheffield S1 4DP, UK
{F.Bernardini,M.Gheorghe}@dcs.shef.ac.uk
[2] Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science and Information Technology
University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK
Natalio.Krasnogor@nottingham.ac.uk
[3] Laboratoire de Méthodes Informatiques UMR 8042
CNRS-Univesité d'Evry, Tour Evry 2, GENOPOLE 523
Place de terasses de l'agora, 91000, Evry, France
giavitto@lami.univ-evry.fr

Abstract. We introduce a model of self-assembly P systems as devices that use some of the features of population P systems to progressively grow a graph structure by forming new bonds between the existing cells and some new cells which are brought into the system step by step. The new cells are then able to self-assemble locally either at the level of cells or at the level of neighbourhoods of cells by using bond-making rules according to a specific self-assembly model. We describe two self-assembly models, called respectively parallel single-point self-assembly and parallel multi-point self-assembly. Then, we precisely state the problem of programmable self-assembly for P systems as the problem of uniquely generating a given graph by means of self-assembly P systems. In this respect, we show how to define a self-assembly P systems that uniquely generates a complete binary tree by using a "minimal" set of resources.

1 Introduction

Self-assembly is the ubiquitous process by which simple individual components autonomously assemble into intricate complexes, which is now being studied in many different research areas of molecular biology, nanotechnology, robotics, and natural computing. In this respect, a number of (abstract) models for self-assembly have been proposed and the problem of having programmable self-assembly models has been identified as a key issue in self-assembly related research. Programmable self-assembly means defining a formalism that can help the systematic (or, even better, automatic) design of an appropriate set of components and the associated interactions which will make these components autonomously, robustly and efficiently assemble to form a desired shape or pattern [5]. In the existing literature, two main approaches to the study of programmable self-assembly models have been considered: an incremental/generative approach (e.g, tiles [8], amorphous computing [1]) where a shape is generated in an incremental way by progressively adding to the existing structure a certain number of components in correspondence to some specific "growing points"; a distributed approach where the desired shape results from the spatial re-organization of some already existing components [4]. In this paper, we propose an incremental/generative approach for the self-assembly of a graph that is based on P systems, a fairly new computational model which abstracts from the structure and functioning of living cells [6]. In particular, we focus on the population P system variant introduced in [2], which provides a formalism for modelling abstract systems consisting of a population of individual components, called cells, which are linked together to form a graph structure; cells interact each other by means of the existing set of links, which is continuously updated by means of some bond-making rules specifying how to add/remove links between the cells in the system. Here, bond-making rules are used in a self-assembly process to progressively enlarge an existing graph structure by forming new bonds between the existing cells and some new cells which are brought into the system step by step; bond-making rules must be used according to a specific self-assembly model. In this respect, we present two self-assembly models where bond-making rules are restricted to be used "locally" either at the level of cells or at the level of neighbourhood of cells. Then, we precisely state the problem of programmable self-assembly, the problem of defining a self-assembly P system that uniquely generates a given target graph. Finally we show how a complete binary tree can be uniquely generated by a self-assembly P system by using a "minimal" set of resources.

2 Preliminaries

We recall here some basic notions and notations commonly used in membrane computing as well as some formal language concepts we need in the rest of the paper. We refer to [6], [7] for further details.

An alphabet is a finite non-empty set of abstract symbols. Given an alphabet O, we denote by O^* the set of all possible strings over O, including the empty string λ. The length of a string $x \in O^*$ is denoted by $|x|$ and, for each $a \in O$, $|x|_a$ denotes the number of occurrences of the symbol a in x. A multiset over O is a mapping $M : O \longrightarrow \mathbf{N}$ such that, $M(a)$ defines the multiplicity of a in the multiset ($\mathbf{N}$ denotes the set of natural numbers). Such a multiset can be represented by a string $a_1^{M(a_1)} a_2^{M(a_2)} \dots a_n^{M(a_n)} \in O^*$ and by all its permutations with $a_j \in O$, $M(a_j) \neq 0$, $1 \leq j \leq n$. In other words, we can say that each string $x \in O^*$ identifies a finite multiset over O defined by $M_x = \{\, (a, |x|_a) \mid a \in O \,\}$. Moreover, given two strings $x, y \in O^*$, we denote by xy their concatenation, which corresponds to the union of the multisets represented by the string x, y.

A finite undirected graph is a pair $G = (V, E)$ where $V \subseteq \mathbf{N}$ if a finite set of nodes, and $E \subseteq V \times V$ is a finite set of unordered pairs called edges; the edges in the graph G are denoted by using the notation $\{i, j\}$, with $i, j \in V$. We restrict our discussion to finite undirected graphs and therefore we will simply use the term graph. A graph $G = (V, E)$ is said to be cyclic if and only if E contains at least a subset of edges of the form $\{i, i_1\}, \{i_1, i_2\}, \ldots, \{i_{n-1}, i_n\}, \{i_n, i\}$, with $n \geq 1$; a graph $G = (V, E)$ is said to be connected if and only if, for each $i \neq j \in V$, either $\{i, j\} \in E$ or there exist $i_1 \neq i_2 \neq \ldots \neq i_{n-1} \neq i_n \in V$, with $n \geq 1$, such that, $\{i, i_1\} \in E$, $\{i_t, i_{t+1}\} \in E$, for each $1 \leq t \leq n-1$, and $\{i_n, j\} \in E$. A tree is a connected acyclic graph where all the nodes are thought as being descendants of an unique node called root; the depth d of a tree is the length of the longest path from the root to another node different from the root. The nodes placed at depth d are called leaves whereas, the nodes placed at depth p, with $1 \leq p \leq d-1$, are called intermediate nodes. A complete n-ary (binary if $n = 2$) tree of depth $d \geq 0$, with $n \geq 1$, is a tree where, for each $0 \leq p \leq d$, the number of nodes placed at level p is exactly n^p. Finally, given two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, we say G_1 is isomorphic to G_2, and we write $G_1 \approx G_2$ if and only if, there exists a bijective mapping $h : V_1 \longrightarrow V_2$ such that, for each $i, j \in V_1$, $\{i, j\} \in E_1$ iff $\{h(i), h(j)\} \in E_2$.

3 Self-Assembly P Systems

We call self-assembly P systems a family of P systems describing a population of cells that self-assemble together to form a graph structure. Cells are the basic functional units of the system and they correspond to nodes in a graph which, at any moment, defines the structure of the system. The edges in such a graph represent links which tightly bond the cells to each other. Such a configuration consisting in a population of cells linked to form a graph structure is called an assembly of cells. Each cell in a given assembly contains a finite multiset of objects which is continuously updated by means of a finite set of transformation rules and communication rules. Transformation rules are used inside the cells to consume some objects in order to produce some new ones; communication rules are instead used to move objects from one cell to the other by using the edges in the graph as if they were communication channels. As well as this, cells in a given assembly can form bonds with some new cells that, step by step, are brought into the system in order to enlarge the current population of cells and form a new graph structure. These bonds are created by some bond-making rules which specify how to connect two cells in the system depending on their respective contents. More precisely, in each step of a self-assembly process, by starting from a given assembly of cells, we first update the content of each cell by using their respective sets of transformation and communication rules; then we introduce into the system some new cells which self-assemble by using bond-making rules to connect themselves to the existing structure according to a chosen self-assembly model.

The definition of self-assembly P systems proposed here is developed alongside the population P system model introduced in [2] where bond-making rules were used for the first time in order to define P systems with a dynamic graph structure.

Definition 1. *A self-assembly P system is a construct*

$$\mathcal{P} = (O, L, \Gamma, \sigma, R, B)$$

where:

- *O is a finite alphabet of symbols called objects;*
- *L is a finite alphabet of symbols called labels;*
- *Γ is a finite set containing t distinct cell templates of the form $C_i = (x_i, l_i, m_i)$ with $1 \leq i \leq t$, $x_i \in O^*$, $l \in L$, and $m_i \geq 1$ the number of bonds that can be formed by that cell;*
- *$\sigma = (w, q, b)$, is the seed cell with $w \in O^*$, $q \in L$, $b \geq 1$ the number of bonds that can be formed by the seed cell;*
- *R is a finite set of rules of the forms:*
  1. *$[\, x \rightarrow y\,]_l$, with $x \in O^+$, $y \in O^*$, $l \in L$ (transformation rules),*
  2. *$[x; y, in]_l$, with $x, y \in O^*$, $l \in L$ (communication rules);*
- *B is a finite set of bond-making rules of the form $(l, x; y, l')$, with $l, l' \in L$, $x, y \in O^*$ and, for some $1 \leq i \leq t$ and $z \in O^*$, $C_i = (x_i, l_i, m_i) \in \Gamma$, $x_i = yz$, $l_i = l'$.*

The symbols in O are used for the objects that can be contained inside the cells whereas, the symbols in L are instead used for labelling the cells and they are necessary to retrieve the subset of rules from R to be used inside a specific cell.

The set Γ contains a finite number of distinct cell templates $C_1, C_2, \ldots, C_t$, with $t \geq 1$; the templates in Γ can be instantiated by cloning an arbitrary number of copies, which can then be added to a given assembly of cells as to enlarge the current structure of the assembly. At the beginning, the initial assembly of cells is given by the seed cell σ and the graph containing only the node associated with this cell and no edges.

Each cell in a self-assembly P system, as well as each cell template, is characterised by a finite multiset of objects defining its content, by a label from L identifying the rules which can be used inside that cell and by a positive integer providing a bound for the total number of bonds which can be formed by that cell. The value of this bound is decreased by one every time a cell form a new bond and this makes sure that, at any moment, the current value of such a bound corresponds to the number of bonds which can still be formed by that cell. A cell can form a new bond if and only if its value of the bound on the number of bonds is greater or equal to 0. A clone of a cell template is a cell that inherits from a template in Γ the initial information about its content, its label, and its bound on the number of bonds.

A transformation rule $[x \rightarrow y]_l$ in R is an usual multiset rewriting rule specifying that, inside a cell with label l, an occurrence of a multiset x can be replaced by an occurrence of a multiset y. A communication rule $[x; y, in]_l$ in R instead specifies that, in presence of a multiset x, a cell with label l can receive an occurrence of the multiset y from one of its neighbouring cells; communication rules are executed non-deterministically; a neighbouring cell is a cell that is directly linked to the cell where the communication rule is applied.

Finally, we have a finite set of bond-making rules in B containing rules of the form $(l, x; y, l')$, with $x, y \in O^*$, $l, l' \in L$. A bond making rule $(l, x; y, l')$ must be read from left to right and it specifies that a cell with label l', containing an occurrence of the multiset x and already present in the current assembly of cells, can form a bond with a new cell which is being added to the current assembly in order to enlarge the existing graph structure if and only if, this cell is a clone of cell template in Γ containing an occurrence of the multiset y and having label l'. Objects are not consumed by bond-making rules but they are rather used as "resources" to be allocated to the bond-making rules in order to determine the number of bonds that can be effectively formed between each cell in the current assembly and the new cells that are being added in order to enlarge the existing graph structure. In particular, the same occurrence of a given multiset of objects can be used only by one bond making rule at a time.

Next, we formally introduce the notion of an assembly of cells in a self-assembly P system $\mathcal{P}$ and clarify the notion of a derivation in such a system.

Definition 2. *Let $\mathcal{P} = (O, L, \Gamma, \sigma, R, B)$ be a self-assembly P system as specified in Definition 1. An assembly of cells in $\mathcal{P}$ is a tuple $\mathcal{A} = (\sigma_1, \sigma_2, \ldots, \sigma_n, \gamma)$ where:*

- *$\sigma_i = (w_i, q_i, b_i)$, for each $1 \leq i \leq n$, is a cell with $w_i \in O^*$ and $q_i \in L$, and $b_i \geq 0$ the number of bonds that can be formed by that cell;*
- *$\gamma = (\{1, 2, \ldots, n\}, E)$, with $E \subseteq \{\{i, j\} \mid 1 \leq i \neq j \leq n\}$, is a connected graph defining the structure of the assembly.*

We also say that the assembly $\mathcal{S} = (\sigma_1, (\{1\}, \emptyset))$, with $\sigma_1 = \sigma$, is the seed assembly of $\mathcal{P}$.

Now, given a self-assembly P system $\mathcal{P}$, a step of derivation is performed in two separate stages: a stage of evolution-communication and a stage of self-assembly.

1. Evolution-communication: we apply the rules in R inside each cell in the current assembly in a non-deterministical maximal parallel manner. This stage of evolution-communication can be considered as being the same as in [2] where an analogous evolution-communication stage is defined that deals with the same type of rules.
2. Self-assembly: a certain number of clones of the cell templates in Γ are added to the current assembly of cells by connecting them to the existing structure by using the bond-making rules in B according to a specific self-assembly model.

This latter stage of self-assembly corresponds to the stage of bond-making considered in [2] for defining the notion of a computation in a population P system. Here the difference with respect to [2] is that we do not destroy any existing bond but we rather increase the structure by adding new bonds and new cells. Specifically, in our model, a bond, as well as a cell, once introduced in an assembly of cells, can never be removed from in any further step of derivation

In the next two subsections, we will present two different self-assembly models which are defined by imposing particular restrictions on the use of the bond-making rules.

3.1 Parallel Single-Point Self-Assembly

We consider here a self-assembly model where, at each stage of self-assembly process, each cell in the current assembly serves as an acretion point where new cells cloned from the template set can attach. This process occurs simultaneously for each of the cells already assembled and hence the name "parallel single-point" self-assembly. More specifically, for each already assembled cell we non-deterministically select a maximal set of new clones to be connected to that cell forming a bond between the later and each clone in the selected set; this set of new cells must be maximal with respect to both the particular choice of bond-making rules, the current distribution of objects inside the cell, and the number of bonds that can be effectively formed by that cell which, by definition, is bounded by a fixed constant. The main restriction in this self-assembly model is that, after the application of the bond-making rules, each new cell ends-up connected to the graph defining the structure of the assembly only by means of a single bond that is created between this new cell and a specific pre-existing cell. Note that bond cannot be formed between two new cells introduced during the same stage of self-assembly. Moreover, as we want the resulting graph to be connected, we also impose the constraint that a new cell is added if and only if a new bond can be effectively formed.

More formally, let $\mathcal{P} = (O, L, \Gamma, \sigma, R, B)$ be a self-assembly P system as specified in Definition 1 and let $\mathcal{A} = (\sigma_1, \sigma_2, \ldots, \sigma_n, \gamma)$, for some $n \geq 1$, be an assembly of cells in $\mathcal{P}$ as specified in Definition 2. We write $A \overset{sa}{\Longrightarrow}_{\mathcal{P}} \mathcal{A}'$, and we say $\mathcal{A}'$ is derived from $\mathcal{A}$ by single-point self-assembly, if and only if $\mathcal{A}'$ is an assembly of cells in $\mathcal{P}$ which is obtained from $\mathcal{A}$ in the following way.

1. For each cell $\sigma_i = (w_i, q_i, b_i)$ in $\mathcal{A}$, with $w_i \in O^*$, $q_i \in L$, $b_i > 0$, $1 \leq i \leq n$, we select a maximal set of bond-making rules from B to be used to link this cell with a maximal number of clones of the cell templates in Γ. This is set of bond-making rules is constructed by assigning in a non-deterministical way the objects in w_i to the rules B as far as it is possible (i.e., all the objects that can be assigned to some bond-making rules must be assigned to some bond-making rule). However, the total number of bond-making rules to be applied must not be greater than b_i. Then, for each bond-making rule $(q_i, x; y, l')$ in B selected to be applied during this self-assembly stage, and for each occurrence of the multiset x in w_i assigned to that rule, we introduce into the new assembly $\mathcal{A}'$ a new cell $\sigma'_h = (w'_h, q'_h, b'_h - 1)$ such that: $h > n$ is a new index which has not yet been used for any other cell in $\mathcal{A}'$, $w'_h = yz = x_j$, $q'_h = l_j$, $b'_h = m_j$, for some $C_j = (x_j, l_j, m_j) \in \Gamma$ and $y, z \in O^*$. At the same time a node h and an edge $\{i, h\}$ are added to the graph γ' in $\mathcal{A}'$. The cell σ_i, is instead replaced in $\mathcal{A}'$ by a cell σ'_i where the value b_i is decreased by the number of bonds formed by this cell in this stage of self-assembly.
2. For each cell σ_i in $\mathcal{A}$, with $1 \leq i \leq n$, which no new cells can be linked to, we add to the assembly $\mathcal{A}'$ a cell $\sigma''_i = \sigma_i$ and a node i in the graph γ'.
3. For each edge $\{i, j\}$ in the graph γ from the assembly $\mathcal{A}$, with $1 \leq i \neq j \leq n$, we add the same edge $\{i, j\}$ to the graph γ'.

4. Finally, we renumber the cells, the nodes, and the edges in $\mathcal{A}'$ in an one-to-one manner with values from $\{1, 2, \ldots, n'\}$, with $n' \geq n \geq 1$ the current number of cells in $\mathcal{A}'$, in such a way to preserve the correspondence between cells, nodes and edges.

Parallel single-point self-assembly has a limited capacity of forming complex graph structures as it works under the assumption that, during a self-assembly stage, a new cell can form only one single bond with a specific cell in the current assembly. Moreover, from that moment on, this new structure will never be altered apart for the introduction of more new cells. Specifically, if we denote by $\Rightarrow_{\mathcal{P}}$ a derivation step in a P system $\mathcal{P}$ which uses single-point self-assembly, and by $\Rightarrow^{+}_{\mathcal{P}}$ its transitive closure, then it is easy to see that the following lemma holds.

Lemma 1. *Let $\mathcal{P} = (O, L, \Gamma, \sigma, R, B)$ be a self-assembly P system as specified in Definition 1. Let $\mathcal{S} = (\sigma_1, (\{1\}, \emptyset))$, with $\sigma_1 = \sigma$, be the initial assembly of cells in $\mathcal{P}$. For each assembly of cell $\mathcal{A}$ such that $\mathcal{S} \Rightarrow^{+} \mathcal{A}$, the graph γ defining the structure of the assembly A is a tree.*

This result is a consequence of the fact that the number of nodes and the number of edges added to the current graph, during a stage of self-assembly, is always equal to the number of edges added at the same time. Moreover, each new node introduced in such a stage results connected to the pre-existing graph by means of at least one edge. This means that, during the self-assembly stage, no new cycles can be created inside the graph defining the structure of the new assembly of cells. Therefore, if we start with an acyclic graph, this property of not containing cycle will be preserved during each step of derivation of a P system that uses parallel single-point self assembly. Thus, in the case of the seed assembly, what we obtain is always a tree which can be thought as being rooted in the seed cell.

3.2 Parallel Multi-Point Self-Assembly

We pass now to consider a self-assembly model where "growing points" for the current graph structure are represented by "neighbourhood" of cells in the current assembly. This means a new cell, which is added to the current assembly during a self-assembly stage, can form more than one bond with many different pre-existing cells but all these cells must be neighbouring cells of a certain cell being itself connected to the same new cell; that is, a new cell can only be connected to cells which are all reachable in one step from a given starting point. In particular, each new cell is now going to form as many bond as possible with respect to a particular choice of bond making rules, the distribution of objects inside the cells from the chosen neighbourhood, and the number of bonds that can be effectively formed with these cells. Moreover, new cell must result connected to the pre-existing graph by means of at least one edge and bonds cannot be formed between new cells added during the same self-assembly stage. This self-assembly model is called parallel multi-point self-assembly as, in each self-assembly stage, many new cells can be added in parallel at the same by forming many new bonds with many different cells already present in the current assembly.

More formally, let $\mathcal{P} = (O, L, \Gamma, \sigma, R, B)$ be a self-assembly P system as specified in Definition 1 and let $\mathcal{A} = (\sigma_1, \sigma_2, \ldots, \sigma_n, \gamma)$, for some $n \geq 1$, be an assembly of cells in $\mathcal{P}$ as specified in Definition 2. We write $A \overset{sa}{\Longrightarrow}_{\mathcal{P}} \mathcal{A}'$, and we say A' is derived from $\mathcal{A}$ by parallel multi-point self-assembly, if and only if $\mathcal{A}'$ is an assembly of cells in $\mathcal{P}$ which is obtained from $\mathcal{A}$ in the following way:

1. Given a clone σ' of a cell template in Γ, we non-deterministically select a neighbourhood of cells η_i, for some $1 \leq i \leq n$, such that η_i contains cells that are directly linked to cell σ_i and all of them including cell σ_i can form a bond with the clone σ'; this set must be maximal with respect to a particular assignment of the objects contained in all these cells to the bond-making rules in B in the sense that no other cell directly linked with σ_i can form a bond with the clone σ'. The clone σ' is then added to the assembly σ' together with a corresponding node and an edge between this new node and each pre-existing node corresponding to a cell in η_i. At the same time, for each cell involved in this bond making process, we update the bound on the number of bonds that can be formed by that cell so to keep track of the bonds that have been just formed.
2. The previous operation of adding a clone is performed in a non-deterministic maximal parallel manner by inserting into the new assembly $\mathcal{A}'$ as many clones as possible according to the current distribution of objects inside the cells and to a particular assignments of these objects to the bond-making rules in B. In particular the following conditions must be satisfied: the total number of bonds formed by a cell during this self-assembly stage is less or equal to the current number of bonds that can be formed by that cell, if two cells compete for the same occurrence of the same multiset placed inside the same cell then, only the cell forming the greater number of bonds is effectively inserted in $\mathcal{A}'$.
3. For each cell σ_i in $\mathcal{A}$, with $1 \leq i \leq n$, which no new cells can be linked to, we add to the assembly $\mathcal{A}'$ a cell $\sigma_i'' = \sigma_i$ and a node i in the graph γ'.
4. For each edge $\{i, j\}$ in the graph γ from the assembly $\mathcal{A}$, with $1 \leq i \neq j \leq n$, we add the same edge $\{i, j\}$ to the graph γ'.

Now it is easy to see that multi-point self-assembly is less restrictive than single-point self-assembly and it can lead to the formation of cyclic graph. Specifically, if we denote by $\Rightarrow_{\mathcal{P}}$ a derivation step in a P system $\mathcal{P}$ which uses multi-point self-assembly, and its transitive closure by $\Rightarrow_{\mathcal{P}}^{+}$, then the following lemma holds.

Lemma 2. *Let $\mathcal{P} = (O, L, \Gamma, \sigma, R, B)$ be a self-assembly P system as specified in Definition 1. Let $\mathcal{S} = (\sigma_1, (\{1\}, \emptyset))$, with $\sigma_1 = \sigma$, be the initial assembly of cells in $\mathcal{P}$. For each assembly of cell $\mathcal{A}$ such that $\mathcal{S} \Rightarrow^{+} \mathcal{A}$, the graph γ defining the structure of the assembly A may contain some cycles.*

Proof. Consider the self-assembly P system $\mathcal{P} = (\{a, b\}, \{\$, \#\}, \Gamma, \sigma, \emptyset, B)$ with: $\Gamma = \{(a, \$, 3), (b, \$, 3), (ab, \#, 3)\}, \sigma = (a, \$, 2)$, and $B = \{(\$, a; b, \$), (\$, a; b, \#)\} \cup$ $\cup \{(\$, b; a, \#), (\#, b; b, \$), (\#, a; a, \$)\}$.

The seed assembly of $\mathcal{P}$ is the assembly $\mathcal{S} = ((a, \$, 2)_1, (\{1\}, \emptyset))$. In the first step of derivation, we can add to cell 1 either a cell $(b, \$, 3)$ by using the bond-making rule $(\$, a; b, \$)$, or a cell $(ab, \#, 3)$ by using the bond-making rule $(\$, a; b, \#)$. Let us suppose the first bond-making rule is used in the first step of derivation so to obtain a new assembly of cells $\mathcal{A}_1$ such that:

$$\mathcal{A}_1 = ((a, \$, 1)_1, (b, \$, 2)_2, (\{1, 2\}, \{\{1, 2\}\})).$$

Now, given the assembly $\mathcal{A}_1$, we can add to cell 1 either a cell $(b, \$, 3)$ by using the bond-making rule $(\$, a; b, \$)$, or a cell $(ab, \#, 3)$ by using the bond-making rule $(\$, a; b, \#)$; we can add to cell 2 a cell $(ab, \#, 3)$ by using the bond-making rule $(\$, b; a, \#)$. Moreover, as we are considering multi-point self-assembly, each clone of the cell $(ab, \#, 3)$ can potentially form two bonds at the same time by using the rule $(\$, a; b, \#)$ in parallel with the $(\$, b; a, \#)$. Specifically, a copy of this cell is the unique cell which is added in the next step of derivation, as this is the cell that can form the greatest number of bonds by preventing any other cells from forming any other bond. In this way we immediately obtain an assembly of cells where the corresponding graph contains the cycle $\{1, 2\}$, $\{2, 3\}$, $\{3, 1\}$. □

4 Uniquely Self-Assembly a Graph

In this section we deal with the problem of defining a self-assembly P system which is able to produce as result of its derivations a given target graph; this graph is supposed to be connected and with no loop edges (i.e., edges linking a node with itself). In particular, we want this graph to be uniquely generated by the defined P system, that is, all the possible derivations must always produce, after a finite number of steps, a similar assembly of cells where the corresponding graph is isomorphic to the given target graph. As well as this, all these derivations must "halt" immediately after having produced this particular assembly of cells; halting, in this context, means the self-assembly P system produces an assembly of cells where no more transformation or communication rules can be applied to the objects placed inside the cells and no more bond-making rules can be applied to the current graph structure.

More precisely, let $\Rightarrow_{\mathcal{P}}$ be the notion of derivation step in a self-assembly P system $\mathcal{P}$ as specified in the previous section and let $\Rightarrow_{\mathcal{P}}^{+}$ be its transitive closure. Moreover, given an assembly of cell $\mathcal{A}$, we denote by $\mathcal{A}.\gamma$ the graph defining the structure of the assembly $\mathcal{A}$. Then, we say $\mathcal{P}$ uniquely generates a graph G if and only if:

- there exists an assembly of cells $\mathcal{A}$ such that $\mathcal{S} \Rightarrow_{\mathcal{P}}^{+} \mathcal{A}$ and $\mathcal{A}.\gamma \approx G$;
- for all assembly of cells $\mathcal{A}$ with $\mathcal{A}.\gamma \not\approx G$, if $\mathcal{S} \Rightarrow_{\mathcal{P}}^{+} \mathcal{A}$ then, there exists $\mathcal{A}'$ with $\mathcal{A}'.\gamma \approx G$ such that $\mathcal{A} \Rightarrow_{\mathcal{P}}^{+} \mathcal{A}'$;
- for all assembly of cells $\mathcal{A}$ with $\mathcal{A}.\gamma \approx G$, if $\mathcal{S} \Rightarrow_{\mathcal{P}}^{+} \mathcal{A}$ then, there does not exist $\mathcal{A}'$ such that $\mathcal{A} \Rightarrow_{\mathcal{P}}^{+} \mathcal{A}'$.

This means all the derivations in the self-assembly P system $\mathcal{P}$ halt by always producing a graph structure isomorphic to the graph G; if that is the case then we write $\mathcal{P} \vdash^{u} G$.

Now, we can precisely state the two main problems related to the unique self-assembly of a graph by means of self-assembly P systems.

Problem 1. For every connected graph $G = (\{1, 2 \ldots, n\}, E)$, with E being included in the set $\{\{i, j\} \,|\, 1 \leq i \neq j \leq n\}$, does exist a self-assembly P system such that $\mathcal{P} \vdash^u G$?

It is obvious that the answer and the solution to this problem highly depend on the particular self-assembly model chosen. Specifically, in the case of single-point self-assembly, Lemma 1 provides a negative answer to Problem 1 whereas, in the case of multi-point self-assembly, Problem 1 still remains open.

Let $G = (\{1, 2 \ldots, n\}, E)$, with $E \subseteq \{\{i, j\} \,|\, 1 \leq i \neq j \leq n\}$, be a connected graph. If a self-assembly P system with at most o objects, at most l labels, at most c cell templates, at most b bond making rules and at most r transformation and communication rules per each cell label, exists such that $\mathcal{P} \vdash^u G$ then, it is denoted by $\mathcal{P}_{o,l,c,b,r}$. We say $\mathcal{P}_{o,l,c,b,r}$ is optimal if there does not exist $\mathcal{P}'_{o',l',c',b',r'}$ uniquely generating G with at least one of these primed parameters being less than the corresponding one in the first self-assembly P systems and the others having the same values.

Problem 2. Given a connected graph $G = (\{1, 2 \ldots, n\}, E)$, with E being included in the set $\{\{i, j\} \,|\, 1 \leq i \neq j \leq n\}$. If a self-assembly P system $\mathcal{P}_{o,l,c,b,r}$ exists such that $\mathcal{P}_{o,l,c,b,r} \vdash^u G$ then, is $\mathcal{P}_{o,l,c,b,r}$ optimal?

The following lemma states that a complete binary tree with depth $d \geq 0$ can be uniquely generated by a P system $\mathcal{P}_{2d+2,2,2,2(d-1),d+1}$. We do not know whether this is optimal or not but we claim it is not.

Lemma 3. *Let T be a complete binary tree with depth $d \geq 0$. We can always construct a self-assembly P systems that uniquely generate T by using: $2d + 2$ different objects, 2 different labels, 2 different cell templates, $2(d - 1)$ different bond-making rules, and at most $d + 1$ transformation and communication rules per cell.*

Proof. Let T be a complete binary tree with depth $d \geq 0$. We construct a self-assembly P system $\mathcal{P}$ that uses parallel single-point self-assembly and such that:

$$\mathcal{P} = (O, L, \Gamma, \sigma, R, B),$$

with:

- $O = \{\, a, b \,\} \cup \{\, \$_p, \$'_p \,|\, 0 \leq p \leq d \,\}$;
- $K = \{c_1, c_2\}$;
- $\Gamma = \{(a, c_1, 3), (a, c_2, 3)\}$;
- $\sigma = (b, c_1, 2)$;
- $R = \{[\, \$_p \rightarrow \$'_{p+1}\$'_{p+1} \,]_{c_1} \,|\, 0 \leq p \leq d - 1\}$
  $\cup\{[\, \$'_p \rightarrow \$_{p+1}\$_{p+1} \,]_{c_2} \,|\, 0 \leq p \leq d - 1\}$
  $\cup\{\, [a; \$_p\$_p, in]_{c_1} \,|\, 0 \leq p \leq d - 1\} \cup \{\, [a; \$'_p\$'_p, in]_{c_1} \,|\, 0 \leq p \leq d - 1\}$
  $\cup\{\, [\, b \rightarrow \$_0\$_0 \,]_{c_1} \,\}$;

- $B = \{\, (c_1, \$_p; a, c_2) \mid 0 \leq p \leq d-1\}$
  $\cup\{\, (c_2, \$'_p; a, c_1) \mid 0 \leq p \leq d-1\}$.

The seed assembly of $\mathcal{P}$ is the assembly $\mathcal{S} = ((b, c_1, 2)_1, (\{1\}, \emptyset))$. In the first stage of evolution-communication, we apply inside the seed cell the rule $[\, b \to \$_0\$_0\,]_{c_1}$ in order to produce inside this cell two copies of the object $\$_0$. Then, in the self-assembly stage, we connect the seed cell with two new cells $\sigma_2 = (a, c_2, 3)$, $\sigma_3 = (a, c_2, 3)$ by using the bond-making rule $(c_1, \$_0; a, c_2)$ twice. Next, inside the seed cell, we apply the rule $[\, \$_0 \to \$'_1\$'_1\,]_{c_1}$ in order to produce four copies of the object $\$'_1$. Both cell σ_3 and σ_2, in the next step of evolution-communication, receive two copies of this object by using, inside both of them, a rule $[a; \$'_p\$'_p, in]_{c_1}$; these objects are then used both cell σ_3 and σ_2 to attract two new cells labeled by c_1 by using the bond making rule $(c_2, \$_1; a, c_2)$ four times. This process can be then iterated for each level $p \leq d-1$ by adding to the current structure, during each step, exactly 2^p cells; the process halts immediately after having produced inside the new cells objects of the form $\$_d$, which no rules can be applied to these cells. This mean the tree is correctly generated in d steps by starting from the root and adding the leaves in the last step of derivation. □

Notice that the same construction can be applied to any complete n-ary tree of depth $d \geq 0$ by just augmenting the number of objects $\$_p$ that are produced inside the cells placed at level $0 \leq p \leq d-1$.

5 Conclusions

Self-assembly P systems are devices that use some of the features of population P systems [2] to progressively increase a graph structure by forming new bonds between the existing cells and some new cells which are brought into the system step by step. Specifically, with respect to [2], bond-making rules can be used only to increase the number of links in the graph defining the structure of the system and they can never be used to alter the structure of an already formed assembly of cells. As well as this, bond-making rules are restricted to be applied locally in correspondence of a certain neighbourhood of cells where self-assembly can take place between cells that are supposed to be "attracted" in that particular vicinity. Moreover, self-assembly P systems use transformation and communication rules to continuously update the internal configuration of the cells and vary the distribution of objects between various cells in the system. The problem of defining in a self-assembly P systems that, for a given self-assembly model, are able to generate any graph of any form remains an open. In fact, here we have only been able to show how to generate a complete binary tree in an "efficient" way by means of a P system that uses a "limited" number of resources. In particular, this is achieved by exploiting the features of transformation and communication rules which allow cells to update this internal configuration and exchange objects with its neighbouring cells. In general, many other features of P systems may be introduced in self-assembly P systems so to have systems consisting in a finite number of cells which are able to re-organise themselves by means of local and limited interactions in order to produce a desired shape or pattern.

Considering the high sensitivity of the final shape of the graph with respect to the rules, the study of self-assembly P systems would certainly benefit from a simulation tool. This kind of tool can be used to "play" several variations and to check the result on some cases. We consider the use of the MGS [3] programming language to develop such simulator. As a matter of fact, MGS provides both multiset rewriting and graph rewriting as well as the ability to create and transform, by rewriting rules, a graph of multisets.

Acknowledgements

The research of F.B. and M.G. has been supported by the Engineering and Physical Sciences Research Council of United Kingdom (EPSRC), grant GR/R84221/01. The research of N.K. has been supported by EPSRC grants EP/D021847/1 and EP/D023343/1. The authors wish to thank the anonymous referees for their comments that allow us to improve the content of this paper.

References

1. Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, F., T., Nagpal, R., Rauch, E., Sussman, J., G., Weiss, R.: Amorphous Computing. *Communication of the ACM* 43 (2000) 74-82
2. Bernardini, F., Gheorghe, M.: Population P Systems. *Journal of Universal Computer Science* 20 (2002) 509-539
3. Giavitto,J.-L., Michel, O.: The Topological Structure of Membrane Computing. *Fundamenta Informaticae* 49 (2002) 107-129
4. Klavins, E.: Automatic Synthesis of Controllers for Assembly and Formation Forming. In: *Proceedings of the International Conference on Robotics and Automation.* (2002)
5. Krasnogor, N., Gustafson, S.: A Family of Conceptual Problems in the Automated Design of Systems Self-Assembly. In: *Proceedings of the 2nd International Conference on the Foundations of Nanoscience: Self-Assembled Architectures and Devices.* (2005)
6. Păun, Gh.: *Membrane Computing. An Introduction.* Springer-Verlag, Berlin Heidelberg New York (2002)
7. Rozenberg, G., Salomaa, A. (eds.): *Handbook of Formal Languages,* Vol. 1-3 Springer-Verlag, Berlin Heidelberg New York (1997)
8. Winfree, E.: *Algorithmic Self-Assembly of DNA.* PhD Thesis, California Institute of Technology (1998).