Solving Subset Sum in Linear Time by Using Tissue P Systems with Cell Division

Daniel Díaz-Pernil, Miguel A. Gutiérrez-Naranjo,
Mario J. Pérez-Jiménez, and Agustín Riscos-Núñez

Research Group on Natural Computing
ETS Ingeniería Informática - University of Sevilla
{sbdani,magutier,marper,ariscosn}@us.es

Abstract. Tissue P systems with cell division is a computing model in the framework of Membrane Computing based on intercellular communication and cooperation between neurons. The ability of cell division allows us to obtain an exponential amount of cells in linear time and to design cellular solutions to **NP**-complete problems in polynomial time. In this paper we present a solution to the Subset Sum problem via a family of such devices. This is the first solution to a numerical **NP**-complete problem by using tissue P systems with cell division.

1 Introduction

In the cell-like model of P systems [6], membranes are hierarchically arranged in a tree-like structure. Its biological inspiration comes from the morphology of cells, where small vesicles are surrounded by larger ones. This biological structure can be abstracted into a tree-like graph, where the root represents the skin of the cell (i.e. the outermost membrane) and the leaves represent membranes that do not contain any other membrane (elementary membranes). Besides, two nodes in the graph are connected if they represent two membranes such that one of them contains the other one.

Recently, new models of P systems have been explored. One of them is the model of *tissue P systems* where the tree-like membrane structure is not considered anymore, being replaced by a general graph.

This model has two biological inspirations (see [4]): intercellular communication and cooperation between neurons. The common mathematical model of these two mechanisms is a net of processors dealing with symbols and communicating these symbols along channels specified in advance. The communication among cells is based on symport/antiport rules, which were introduced as communication rules for P systems in [5]. In symport rules objects cooperate to traverse a membrane together in the same direction, whereas in the case of antiport rules, objects residing at both sides of the membrane cross it simultaneously but in opposite directions.

This paper is devoted to the study of the computational efficiency of tissue P systems with cell division. In literature different models of cell-like P systems

have been successfully used in order to design efficient solutions to **NP**-complete problems (see, for example, [2] and the references therein). These solutions are obtained by generating an exponential amount of workspace in polynomial time and using parallelism to check simultaneously all the candidate solutions.

From the seminal definition of tissue P systems [3,4], several research lines have been developed and other variants have arisen (see [1] and references therein). One of the most interesting variants of tissue P systems was presented in [8], where the definition of tissue P systems is combined with the one of P systems with active membranes, yielding *tissue P systems with cell division*. The biological inspiration is clear: alive tissues are not *static* networks of cells, since cells are duplicated via mitosis in a natural way. One of the main features of such tissue P systems with cell division is related to their computational efficiency. In [8], a polynomial-time solution to the **NP**-complete problem SAT is shown, and in [1] a linear-time solution for the 3-COL problem was presented. In this paper we go on with the research in this model and present a linear-time solution to another well-known numerical **NP**-complete problem: the Subset Sum problem.

The paper is organised as follows: first we recall some preliminary concepts and the definition of tissue P systems with cell division. Next, recognising tissue P systems are briefly described. A linear–time solution to the Subset Sum problem is presented in the following section, including a short overview of the computation and of the necessary resources. Finally, some conclusions and lines for future research are presented.

2 Preliminaries

In this section we briefly recall some of the concepts used later on in the paper.

An *alphabet*, Σ, is a non empty set, whose elements are called *symbols*. An ordered sequence of symbols is a *string*. The number of symbols in a string u is the *length* of the string, and it is denoted by $|u|$. As usual, the empty string (with length 0) will be denoted by λ. The set of strings of length n built with symbols from the alphabet Σ is denoted by Σ^n and $\Sigma^* = \cup_{n\geq 0}\Sigma^n$. A *language* over Σ is a subset from Σ^*. A *multiset* m over a set A is a pair (A, f) where $f : A \to \mathbb{N}$ is a mapping. If $m = (A, f)$ is a multiset then its *support* is defined as $supp(m) = \{x \in A \,|\, f(x) > 0\}$ and its *size* is defined as $\sum_{x\in A} f(x)$. A multiset is empty (resp. finite) if its support is the empty set (resp. finite). If $m = (A, f)$ is a finite multiset over A, then it will be denoted as $m = \{\{a_1, \ldots, a_k\}\}$, where each element a_i occurs $f(a_i)$ times. Multisets can also be represented as strings in a natural way.

In what follows we assume the reader is already familiar with the basic notions and the terminology underlying P systems. For details, see [7].

3 Tissue P Systems with Cell Division

In the first definition of the model of tissue P systems [3,4] the membrane structure did not change along the computation. The main features of tissue P systems

with cell division, from the computational point of view, are that cells obtained by division have the same labels as the original cell, and if a cell is divided, then its interaction with other cells or with the environment is blocked during the mitosis process. In some sense, this means that while a cell is dividing it closes the communication channels with other cells and with the environment. This features imply that the underlying graph is dynamic, as nodes can be added during the computation by division and the edges can be deleted/re-established for dividing cells.

Formally, a *tissue P system with cell division* of initial degree $q \geq 1$ is a tuple of the form $\Pi = (\Gamma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_0)$, where:

1. Γ is a finite *alphabet*, whose symbols will be called *objects*.
2. $w_1, \ldots, w_q$ are strings over Γ.
3. $\mathcal{E} \subseteq \Gamma$.
4. $\mathcal{R}$ is a finite set of rules of the following form:
   (a) *Communication rules*: $(i, u/v, j)$, for $i, j \in \{0, 1, \ldots, q\}, i \neq j$, $u, v \in \Gamma^*$.
   (b) *Division rules*: $[a]_i \rightarrow [b]_i[c]_i$, where $i \in \{1, 2, \ldots, q\}$ and $a, b, c \in \Gamma$.
5. $i_0 \in \{0, 1, 2, \ldots, q\}$.

A tissue P system with cell division of degree $q \geq 1$ can be seen as a set of q cells labelled by $1, 2, \ldots, q$. We shall use 0 as the label of the environment, and i_0 for the output region (which can be the region inside a cell or the environment). Despite the fact that cell-like models include an explicit description of the initial membrane structure, this is not the case here. Instead, the underlying graph expressing connections between cells is implicit, being determined by the communication rules (the nodes are the cells and the edges indicate if it is possible for pairs of cells to communicate directly).

The strings $w_1, \ldots, w_q$ describe the multisets of objects placed initially in the q cells of the system. We interpret that $\mathcal{E} \subseteq \Gamma$ is the set of objects placed in the environment, each one of them in an arbitrarily large amount of copies.

The communication rule $(i, u/v, j)$ can be applied over two cells i and j such that u is contained in cell i and v is contained in cell j. The application of this rule means that the objects of the multisets represented by u and v are interchanged between the two cells.

The division rule $[a]_i \rightarrow [b]_i[c]_i$ can be applied over a cell i containing object a. The application of this rule divides this cell into two new cells with the same label. All the objects in the original cell are replicated and copied in each of the new cells, with the exception of the object a, which is replaced by the object b in the first new cell and by c in the second one.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). In one step, each object in a membrane can only be used for one rule (non-deterministically chosen when there are several possibilities), but any object which can participate in a rule of any form must do it, i.e, in each step we apply a maximal set of rules. This way of applying rules has only one restriction: when a cell is divided, the division rule is the only one which is applied for that cell in that step; the objects inside that cell do not move in that step.

4 Recognising Tissue P Systems with Cell Division

NP-completeness has been usually studied in the framework of *decision problems*. Let us recall that a decision problem is a pair (I_X, θ_X) where I_X is a language over a finite alphabet (whose elements are called *instances*) and θ_X is a total boolean function over I_X.

In order to study the computational efficiency, a special class of tissue P systems is introduced in [8]: *recognising*[1] *tissue P systems*. The key idea is the same one as from cell-like recognising P systems, that were introduced in [9] as the natural framework to study and solve decision problems within Membrane Computing. Note that deciding whether an instance of a problem has an affirmative or negative answer is equivalent to deciding if a string belongs or not to the language associated with the problem.

In literature, recognising cell-like P systems are associated in a natural way with P systems with *input*. The data related to an instance of the decision problem need to be provided to the P system in order to compute the appropriate answer. This is done by codifying in unary form each instance as a multiset placed in an *input membrane*. The output of the computation (`yes` or `no`) is sent to the environment. In this way, cell-like P systems with input and external output are devices which can be seen as black boxes, in the sense that the user provides the data before the computation starts, and then waits *outside* the P system until it sends to the environment the output in the last step of the computation.

A recognising tissue P system with cell division of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \Sigma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_{in}, i_0)$, where

- $(\Gamma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_0)$ is a tissue P system with cell division of degree $q \geq 1$ (as defined in the previous section).
- The working alphabet Γ has two distinguished objects `yes` and `no`, present in at least one copy in an initial multiset $w_1, \ldots, w_q$, but not present in $\mathcal{E}$.
- Σ is an (input) alphabet strictly contained in Γ.
- $i_{in} \in \{1, \ldots, q\}$ is the input cell.
- The output region i_0 is the environment.
- All computations halt.
- If $\mathcal{C}$ is a computation of Π, then either the object `yes` or the object `no` (but not both) must have been released into the environment, and only in the last step of the computation.

The computations of the system Π with input $w \in \Gamma^*$ start from a configuration of the form $(w_1, w_2, \ldots, w_{i_{in}} w, \ldots, w_q; \mathcal{E})$, that is, after adding the multiset w to the contents of the input cell i_{in}. We say that the multiset w is *recognised* by Π if and only if the object `yes` is sent to the environment, in the last step of all its associated computations. We say that $\mathcal{C}$ is an accepting (resp. rejecting) computation if the object `yes` (resp. `no`) appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$.

[1] In [8] they were called *recognizer* tissue P systems.

Definition 1. *We say that a decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ of recognising tissue P systems with cell division if the following holds:*

- *The family $\mathbf{\Pi}$ is* polynomially uniform *by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$.*
- *There exists a pair (cod, s) of polynomial-time computable functions over I_X such that:*
  - *for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;*
  - *the family $\mathbf{\Pi}$ is* polynomially bounded *with regard to (X, cod, s), that is, there exists a polynomial function p, such that for each $u \in I_X$ every computation of $\Pi(s(u))$ with input $cod(u)$ is halting and, moreover, it performs at most $p(|u|)$ steps;*
  - *the family $\mathbf{\Pi}$ is* sound *with regard to (X, cod, s), that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u))$ with input $cod(u)$, then $\theta_X(u) = 1$;*
  - *the family $\mathbf{\Pi}$ is* complete *with regard to (X, cod, s), that is, for each $u \in I_X$, if $\theta_X(u) = 1$, then every computation of $\Pi(s(u))$ with input $cod(u)$ is an accepting one.*

In the above definition we have imposed to every tissue P system $\Pi(n)$ a *confluent* condition, in the following sense: every computation of a system with the *same* input multiset must always give the *same* answer. The pair of functions (cod, s) are called a *polynomial encoding* of the problem in the family of P systems.

We denote by $\mathbf{PMC}_{TD}$ the set of all decision problems which can be solved by means of recognising tissue P systems with cell division in polynomial time.

5 The Subset Sum Problem

The Subset Sum problem is the following one: *Given a finite set A, a weight function, $w : A \to \mathbb{N}$, and a constant $k \in \mathbb{N}$, determine whether or not there exists a subset $B \subseteq A$ such that $w(B) = k$.*

Next, we shall prove that the Subset Sum problem can be solved in a linear time by a family of recognising tissue P systems with cell division. We shall address the resolution via a brute force algorithm.

We will use a tuple $(n, (w_1, \ldots, w_n), k)$ to represent an instance of the problem, where n stands for the size of $A = \{a_1, \ldots, a_n\}$, $w_i = w(a_i)$, and k is the constant given as input for the problem.

Theorem 1. `SUBSET SUM` $\in \mathbf{PMC}_{TD}$.

Proof. Let $A = \{a_1, \ldots, a_n\}$ be a finite set, $w : A \longrightarrow \mathbb{N}$ a weight function with $n = |A|$ and $k \in \mathbb{N}$. Let $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a *function* defined by $g(n, k) =$

$((n+k)(n+k+1)/2)+n$. This function is primitive recursive and bijective between $\mathbb{N}^2$ and $\mathbb{N}$ and computable in polynomial time. Let us denote by $u = (n,(w_1,\ldots,w_n),k)$, where $w_i = w(a_i)$, $1 \leq i \leq n$, the given instance of the problem. We define the polynomially computable function $s(u) = g(n,k)$.

We will provide a family of tissue P systems where each P system solves all the instances of the SUBSET SUM problem with the same size. The weight function w of the concrete instance will be provided via an input multiset determined via the function $cod(u) = \{\{v_i^j : w(a_i) = j \wedge 1 \leq i \leq n\}\} \cup \{\{q^k\}\}$, where v_i^j (i.e., j copies of object v_i) represents that j is the weight of the element a_i.

Next, we will provide a family of recognising tissue P systems with cell division which solve the SUBSET SUM problem in linear time. For each $(n,k) \in \mathbb{N}^2$ we will consider the system $\Pi(n,k) = (\Gamma, \Sigma, \omega_1, \omega_2, \mathcal{R}, \mathcal{E}, i_{in}, i_0)$, where

- $\Gamma = \Sigma \cup \{A_i, B_i, : 1 \leq i \leq n\}$
  $\cup \{a_i : 1 \leq i \leq n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 11\}$
  $\cup \{c_i : 1 \leq i \leq n+1\}$
  $\cup \{d_i : 1 \leq i \leq \lceil \log n \rceil + \lceil \log(k+1) \rceil + 4\}$
  $\cup \{e_i : 1 \leq i \leq \lceil \log n \rceil + 1\}$
  $\cup \{B_{ij} : 1 \leq i \leq n \wedge 1 \leq j \leq \lceil log(k+1) \rceil + 1\}$
  $\cup \{b, D, p, g_1, g_2, f_1, T, S, N, \texttt{yes}, \texttt{no}\}$
- $\Sigma = \{q\} \cup \{v_i : 1 \leq i \leq n\}$
- $\omega_1 = a_1\, b\, c_1\, \texttt{yes}\, \texttt{no}$
- $\omega_2 = DA_1 \cdots A_n$
- $\mathcal{R}$ is the following set of rules:
  1. *Division rules:*
     $r_{1,i} \equiv [A_i]_2 \rightarrow [B_i]_2[\lambda]_2$ for $i = 1,\ldots,n$
  2. *Communication rules:*
     $r_{2,i} \equiv (1, a_i/a_{i+1}, 0)$ for $i = 1,\ldots,n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 10$
     $r_{3,i} \equiv (1, c_i/c_{i+1}^2, 0)$ for $i = 1,\ldots,n$
     $r_4 \equiv (1, c_{n+1}/D, 2)$
     $r_5 \equiv (2, c_{n+1}/d_1 e_1, 0)$
     $r_{6,i} \equiv (2, e_i/e_{i+1}^2, 0)$ for $i = 1,\ldots,\lceil \log n \rceil$
     $r_{7,i} \equiv (2, d_i/d_{i+1}, 0)$ for $i = 1,\ldots,\lceil \log n \rceil + \lceil \log(k+1) \rceil + 3$
     $r_{8,i} \equiv (2, e_{\lceil \log n \rceil + 1} B_i / B_{i1}, 0)$ for $i = 1,\ldots,n$
     $r_{9,i,j} \equiv (2, B_{ij}/B_{ij+1}^2, 0)$ for $i = 1,\ldots,n$, $j = 1,\ldots,\lceil \log(k+1) \rceil$
     $r_{10,i} \equiv (2, B_{i\lceil \log(k+1) \rceil + 1} v_i/p, 0)$ for $i = 1,\ldots,n$
     $r_{11} \equiv (2, pq/\lambda, 0)$
     $r_{12} \equiv (2, d_{\lceil \log n \rceil + \lceil \log(k+1) \rceil + 4}/g_1 f_1, 0)$
     $r_{13} \equiv (2, f_1 p/\lambda, 0)$
     $r_{14} \equiv (2, f_1 q/\lambda, 0)$
     $r_{15} \equiv (2, g_1/g_2, 0)$
     $r_{16} \equiv (2, g_2 f_1/T, 0)$
     $r_{17} \equiv (2, T/\lambda, 1)$
     $r_{18} \equiv (1, bT/S, 0)$
     $r_{19} \equiv (1, S\texttt{yes}/\lambda, 0)$

$r_{20} \equiv (1, a_{n+\lceil \log n \rceil + \lceil \log(k+1) \rceil + 11} b/N, 0)$
$r_{21} \equiv (1, N\texttt{no}/\lambda, 0)$

- $\mathcal{E} = \Gamma - \{\texttt{yes}, \texttt{no}\}$
- $i_{in} = 2$, is the input cell
- $i_0 = env$, is the output cell

The design is structured in the following stages:

- *Generation Stage*: The initial cell labelled by 2 is divided into two new cells; and the divisions are iterated n times until a cell has been produced for each possible candidate solution. Simultaneously to this process, two counters (c_i and a_i) evolve in the cell labelled by 1: the first one controls the step in which the communication between cells 2 starts and the second one will be useful in the output stage.
- *Pre–checking Stage*: When this stage starts, we have 2^n cells labelled by 2, each of them encoding a subset of the set A. In each such a cell, as many objects p as the weight of the corresponding subset will be generated. Recall that there are k copies of the object q in each cell labelled by 2 (since they were introduced as part of the input multiset).
- *Checking Stage*: In each cell labelled by 2, the number of copies of objects p and q are compared. The way to do that is removing from the cell in one step all possible pairs (p, q). After doing so, if some objects p or q remain in the cell, then the cell was not encoding a solution of the problem; otherwise, the weight of the subset of A encoded on the cell equals to k and hence it encodes a solution to the problem.
- *Output Stage*: The system sends to the environment the right answer according to the results of the previous stage:
  - *Answer yes:* After the checking stage, there is a cell labelled by 2 without objects p nor q. In this case, such a cell sends an object T to the cell 1. This object T causes the cell 1 to expel an object `yes` to the environment (see rules r_{18} and r_{20}).
  - *Answer no:* In each cell labelled by 2 there exists an object p or q. In this case, no object T arrives to the cell labelled by 1 and an object `no` is sent to the environment.

The non-determinism of this family of recognising tissue P systems with cell division lies in the division rules. These division rules are not competitive: the non-determinism is due to the order in which the rules are applied. When the generation stage ends, the same configuration is reached regardless the order of application of the division rules: 2^n cells labelled by 2, each of them with the codification of a different subset of A.

6 An Overview of the Computation

First of all, we recall the polynomial encoding of the Subset Sum problem in the family $\mathbf{\Pi}$ constructed in the previous section. Let $u = (n, (w_1, \ldots, w_n), k)$ be an

instance of the problem, $s(u) = g(n,k)$ and $cod(u) = \{\{v_i^j : w(a_i) = j \wedge 1 \leq i \leq n\}\}$.

Next, we describe informally how the recognising tissue P system with cell division $\Pi(s(u))$ with input $cod(u)$ works. Let us start with the *generation stage*. Recall that if a division rule is triggered, the communication rules cannot be simultaneously applied. In this stage we have two parallel processes:

- On the one hand, in the cell labelled by 1 we have two counters: a_i, which will be used in the answer stage and c_i, which will be multiplied until getting 2^n copies in exactly n steps.
- On the other hand, in the cell labelled by 2, the division rules are applied. For each object A_i (which codifies a member of the set A) we obtain two cells labelled by 2: One of them has an element B_i and the other does not.

When all divisions have been done, after n steps, we will have 2^n cells with label 2 and each of them will contain the encoding of a subset of A. At this moment, the generation stage ends and the pre-checking stage begins.

For each cell 2, an object D is changed by a copy of the counter c. In this way, in the cell 1 2^n copies of D will appear and, in each cell labelled by 2 there will be an object c_{n+1}. The occurrence of such object c_{n+1} in the cells 2 will produce the apparition of two counters:

(a) The counter d_i lets the checking stage start, since it produces the apparition of the objects g_1 and f_1 after $\lceil \log n \rceil + \lceil \log(k+1) \rceil + 4$ steps.
(b) The counter e_i will be multiplied for obtaining n copies of $e_{\lceil \log n \rceil + 1}$ in the step $n + \lceil \log n \rceil + 2$. Then, we trade objects $e_{\lceil \log n \rceil + 1}$ and B_i against B_{i1} for each element A_i in the subset associated with the membrane. After that, for each $1 \leq i \leq n$ we get $k+1$ copies of $B_{i\lceil \log(k+1) \rceil + 1}$. Then for each element A_i in the subset associated with the membrane we get $\min\{k+1, w(a_i)\}$ copies of object p, in the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 5$.

The checking takes place in the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 6$, when all pairs of objects p and q present in any cell labelled by 2 are sent to the environment. In this way, if the weight of the subset associated with a cell is equal to k, then no object p or q remains in this cell in the next step. Otherwise, if the encoding is not exactly of weight k, then at least one object p or q will remain in the cell. In the next step the answer stage starts. Two cases must be considered for each cell:

- If no object p or q remain in the cell, the object f_1 does not evolve, g_1 evolves to g_2, and in the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 8$ the objects f_1 and g_2 are traded by T with the environment. In the next step T is sent to the cell 1, and in the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 10$, the objects T and b are sent to the environment traded by S. Finally in the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 11$ the objects S and `yes` are sent to the environment.
- If any object p or q remains in the cell, such object is sent to the environment together with the object f_1. This causes that the object b still remains in

the cell 2 after the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 10$. In this way, the objects b and $a_{n+\lceil \log n \rceil + \lceil \log(k+1) \rceil + 11}$ are traded by the object N with the environment, and in the step $n + \lceil \log n \rceil + \lceil \log(k+1) \rceil + 12$ the objects N and `no` are sent to the environment.

6.1 Necessary Resources

Next, we show that the family $\mathbf{\Pi} = \{\Pi(g(n,k)) \ : \ n,k \in \mathbb{N}\}$ defined in Theorem 1 is polynomially uniform by Turing machines. To this aim we are going to show that it is possible to build $\Pi(g(n,k))$ in polynomial time with respect to the size of u.

It is easy to check that the rules of a system $\{\Pi(g(n,k)) \ : \ n,k \in \mathbb{N}\}$ of the family are defined recursively from the values n and k. Besides, the necessary resources to build an element of the family are of polynomial order with respect to the same:

- Size of the alphabet: $n \cdot \lceil \log(k+1) \rceil + 6n + 2\lceil \log(k+1) \rceil + 3\lceil \log n \rceil + 28 \in O(n \cdot \log k)$
- Initial number of cells: $2 \in \theta(1)$.
- Initial number of objects: $n + 6 \in \theta(n)$.
- Number of rules: $n \cdot \lceil \log(k+1) \rceil + 5n + 2\lceil \log(k+1) \rceil + 3\lceil \log n \rceil + 26 \in O(n \cdot \log k)$
- Maximal length of a rule: 3.

7 Conclusions and Future Work

Natural Computing studies new computational paradigms inspired from various well-known natural phenomena in physics, chemistry and biology. This paper is devoted to a new field in Natural Computing: the study of the structure and functioning of cells as living organisms able to process and generate information.

Membrane Computing is a new cross-disciplinary field of Natural Computing which has reached an important success in its short life. In these years many results have been presented related to the computational power of membrane devices, but up to now no implementation *in vivo* or *in vitro* has been carried out. This paper deals with the study of *algorithms* to solve well-known problems and in this sense it is a theoretical result, mainly related to computational efficiency. Moreover, this paper represents a new step in the study of algorithms in the framework of P systems because it exploits tissue P Systems with Cell Division (a variant poorly studied) to solve an **NP**-complete problem.

The basic idea is to consider a distributed and parallel computing device, structured as the cells of a tissue, by means of arrangement of cells where various chemicals (we call them *objects*, to be free of any interpretation) evolve according to local reaction rules. Because the chemicals from the compartments of a cell are swimming in an aqueous solution, the data structure we consider is that of a *multiset* – a set with multiplicities associated with its elements. Also, in analogy with what happens in a cell, the rules are applied in a parallel and a non–deterministic manner.

P systems are computational devices whose power has to be studied in a deeper extent. In the last years, several papers have explored this power, both in the framework of cell-like P systems and tissue-like P systems with membrane creation. These papers have shown that **NP**-complete problems are solvable (in polynomial time) by families of recognising P systems in such models. In this paper we have shown that numerical **NP**-complete problems can also be solved (in polynomial time) by families of recognising tissue P systems with Cell Division, in a uniform way. The specific techniques for designing solutions to concrete problems (generation, evaluation, checking, and output stages) are quite different from a P system model to another, so the simulation of one model in the other one is not a trivial question.

Other lines to follow in the future are the extension of the techniques presented in this paper for the study of other numerical **NP**-complete problems and to develop a software for simulating these computational processes.

Acknowledgement

The authors wish acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the project of excellence TIC-581 of the Junta de Andalucía.

References

1. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez,A. A linear–time tissue P system based solution for the 3–coloring problem. *Theoretical Computer Science*, to appear.
2. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Romero-Campero, F.J. A linear solution for QSAT with Membrane Creation. *Lecture Notes in Computer Science* **3850**, (2006), 241–252.
3. Martín Vide, C., Pazos, J., Păun, Gh. and Rodríguez Patón, A. A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. *Lecture Notes in Computer Science* **2387**, (2002), 290–299.
4. Martín Vide, C., Pazos, J., Păun, Gh. and Rodríguez Patón, A. Tissue P systems. *Theoretical Computer Science*, **296**, (2003), 295–326.
5. Păun, A. and Păun, Gh. The power of communication: P systems with symport/antiport. *New Generation Computing*, **20**, 3, (2002), 295–305.
6. Păun, Gh. Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1, (2000), 108–143.
7. Păun, Gh. *Membrane Computing. An Introduction*. Springer–Verlag, Berlin, (2002).
8. Păun, Gh., Pérez-Jiménez, M.J. and Riscos-Núñez, A. Tissue P System with cell division. In Gh. Păun, A. Riscos-Núñez, A. Romero-Jiménez and F. Sancho-Caparrini (eds.), *Second Brainstorming Week on Membrane Computing*, Sevilla, Report RGNC 01/2004, (2004), 380–386.
9. Pérez-Jiménez, M.J., Romero-Jiménez, A. and Sancho-Caparrini, F. A polynomial complexity class in P systems using membrane division. In E. Csuhaj-Varjú, C. Kintala, D. Wotschke and Gy. Vaszyl (eds.), *Proceedings of the 5th Workshop on Descriptional Complexity of Formal Systems, DCFS 2003*, (2003), 284–294.