mathematics

MDPI

Article

A Protocol for Solutions to DP-Complete Problems through Tissue Membrane Systems

David Orellana-Martín [1,2,*], Antonio Ramírez-de-Arellano [1,2], José Antonio Andreu-Guzmán [1], Álvaro Romero-Jiménez [1,2] and Mario J. Pérez-Jiménez [1,2]

[1] Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain; aramirezdearellano@us.es (A.R.-d.-A.); josandguz@gmail.com (J.A.A.-G.); romero.alvaro@us.es (Á.R.-J.); marper@us.es (M.J.P.-J.)

[2] SCORE Laboratory, I3US, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

* Correspondence: dorellana@us.es

Abstract: Considering a class $\mathcal{R}$ comprising recognizer membrane systems with the capability of providing polynomial-time and uniform solutions for **NP**-complete problems (referred to as a "presumably efficient" class), the corresponding polynomial-time complexity class $\mathbf{PMC}_{\mathcal{R}}$ encompasses both the **NP** and **co** − **NP** classes. Specifically, when $\mathcal{R}$ represents the class of recognizer presumably efficient cell-like P systems that incorporate object evolution rules, communication rules, and dissolution rules, $\mathbf{PMC}_{\mathcal{R}}$ includes both the **DP** and **co** − **DP** classes. Here, **DP** signifies the class of languages that can be expressed as the difference between any two languages in **NP** (it is worth noting that **NP** $\subseteq$ **DP** and **co** − **NP** $\subseteq$ **co** − **DP**). As **DP**-complete problems are believed to be more complex than **NP**-complete problems, they serve as promising candidates for studying the **P** vs **NP** problem. This outcome has previously been established within the realm of recognizer P systems with active membranes. In this paper, we extend this result to encompass any class $\mathcal{R}$ of presumably efficient recognizer tissue-like membrane systems by presenting a detailed protocol for transforming solutions of **NP**-complete problems into solutions of **DP**-complete problems.

Keywords: complexity class; **DP**; membrane computing; tissue P systems

MSC: 68Q07; 68Q15

Citation: Orellana-Martín, D.; Ramírez-de-Arellano, A.; Andreu-Guzmán, J.A.; Romero-Jiménez, Á.; Pérez-Jiménez, M.J. A Protocol for Solutions to DP-Complete Problems through Tissue Membrane Systems. *Mathematics* **2023**, *11*, 2797. https://doi.org/10.3390/math11132797

Academic Editor: Andrea Scozzari

Received: 8 May 2023
Revised: 13 June 2023
Accepted: 20 June 2023
Published: 21 June 2023

1. Introduction

Roughly speaking, a *mechanical procedure* consists of a set of elementary tasks, possibly repeated, structured by a total order. A *mechanical solution* of an abstract problem is a mechanical procedure such that the execution of the corresponding tasks in the order prefixed by the procedure provides the correct solution of the problem. A *computing model* basically consists of a formal/mathematical definition of the intuitive concept of mechanical procedure. Consequently, in a computing model will be possible to define in a rigorous way what solving an abstract problem through a mechanical way means. A *computing paradigm* is a mathematical theory that allows us to consider computing models satisfying some syntactic and/or semantic properties previously established in the theory.

Membrane Computing is a branch of Natural Computing introduced by Gh. Păun at the end of 1998 [1]. It is a computing paradigm inspired by the architecture and the functioning of living cells, as well as from the way the cells are organized in tissues, organs or other higher-order structures. This paradigm provides distributed, parallel non-deterministic computing models whose computational devices are generically called *membrane systems*. This paper deals with tissue-like membrane systems inspired by the cell inter-communication in tissues, where the processor units, called cells, are considered to be the nodes of a directed graph. In this context, cells can communicate through some kind

of rewriting rules (called symport/antiport rules), which were introduced in membrane computing in [2]. The basic concept of tissue P systems [3,4] gave rise to several research lines and variants: see, for example [5–7]. These systems can present cooperative behavior in the following sense: two or more objects can interact to fire a rule. It can be seen in rules where two or more objects are necessary to be executed. Replication stands as a fundamental function within a cell, where, under optimal conditions, cell division (mitosis) yields two identical copies. Drawing inspiration from this phenomenon, cell division rules were incorporated into the framework of tissue P systems. These rules serve as a mechanism for generating an exponential workspace in terms of the number of objects and cells involved. By employing such rules, the two newly formed cells resulting from the division process contain precisely the same objects, differing at most by a pair of distinct objects.

It is a well-established fact that every decision problem can be associated with a language in a manner that solving the decision problem corresponds to recognizing the "corresponding" language. In the field of Membrane Computing, the concept of "Recognizer membrane systems" was introduced in [8] as a natural framework for solving decision problems through language recognition. A membrane system is considered a recognizer if it exhibits specific syntactic and semantic characteristics: (a) The working alphabet comprises two distinguished objects, namely, `yes` and `no`. (b) There exists an input alphabet that is strictly contained within the working alphabet, along with an input "compartment" (a distinguished membrane in the case of cell-like devices, or a distinguished cell in the case of tissue-like devices). (c) The initial content of each cell consists of a multiset of objects from the working alphabet, excluding those from the input alphabet. (d) All computations performed by the system eventually halt. (e) During each computation, either the object `yes` or the object `no` (but not both) must be released to the environment, and only at the final step.

For a given recognizer membrane system Π, the notation $\Pi + m$ represents the membrane system obtained by adding the multiset m to the content of the input compartment in the initial configuration of Π.

The construction of a recognizer membrane system must be done in polynomial time. Therefore, while all the instances of a decision problem can be solved by a single Turing machine, we must define an infinite family of recognizer membrane systems to solve it. In [9], the concept of a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer membrane systems solves a decision problem X in polynomial time in a uniform way is described. If such family $\mathbf{\Pi}$ can be generated by a deterministic Turing machine working in polynomial time, and there exists a pair (cod, s) of polynomial-time computable functions over the set of instances of X verifying the following: *complete* with regard to (X, cod, s). (for more details, see [9]). Given a computing model $\mathcal{R}$ of recognizer membrane systems, $\mathbf{PMC}_{\mathcal{R}}$ denotes the set of decision problems solvable by families from $\mathcal{R}$ in polynomial-time and in a uniform way. The class $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and under polynomial-time reduction [9]. Thus, if X is a decision problem belongs to a complexity class $\mathcal{K}$ then we deduce that $\mathcal{K} \cup \mathbf{co}\text{-}\mathcal{K} \subseteq \mathbf{PMC}_{\mathcal{R}}$. In [10], a protocol to generate a solution to a **DP**-complete [11] Problem from a solution to a **NP**-complete problem is given in the framework of recognizer P systems with active membranes, that is applied in [12]. In this work, a similar protocol is introduced. It is interesting to point out that any solution to a decision problem can be integrated with another solution to create a single solution to the product problem in order to execute each of them in parallel, with the corresponding speedup of the system.

The terms *efficiency* and *presumed efficiency* of a computing model were introduced in [13]. These concepts are associated with the ability of the model to provide polynomial-time solutions to computationally hard problems. Specifically, a computing model is said to be *efficient* (respectively, *presumably efficient*) if it has the ability to provide polynomial-time solutions for intractable problems (resp., **NP**-complete problems) [13]. The term *presumably efficient* refers to the fact that, as generally believed, if $\mathbf{P} \neq \mathbf{NP}$ then each **NP**-complete problem is an intractable one; consequently, under this hypothesis, any presumably efficient

model would be efficient. Thus, if $\mathcal{R}$ is a presumably efficient computing model of a recognizer tissue-like membrane system, then $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{R}}$, because of class $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and under polynomial-time reduction.

In [14], the authors found that $\mathbf{P}^{\#\mathbf{P}}$ is an upper bound of tissue P systems that use either division or separation rules and communication rules of any length. It is interesting to find lower bounds to know which complexity class characterizes the class of recognizer tissue P systems with symport/antiport rules and division/separation rules. In fact, in [15], the authors demonstrated that the class $\mathbf{P}^{\#\mathbf{P}}$ characterizes the class of recognizer tissue P systems with symport/antiport rules and division/separation rules, softly changing the definition of recognizing a language by means of a family of membrane systems. Apart from that, different results have been obtained both obtaining lower or upper bounds [16–18].

The rest of the paper is organized as follows: Next section is devoted to introducing the methodology to construct a solution to a product problem. In Section 3, the protocol is applied to a specific solution of SAT to create a solution to the SAT-UNSAT problem that is explained in Section 4. Finally, some conclusions and insights for future works are explained.

2. Methodology

Let us recall that for each $k \geq 1$, $\mathcal{TDC}(k)$ (respectively, $\mathcal{TSC}(k)$) denotes the computing model of recognizer tissue-like membrane systems with cell division (respectively, cell separation) and communication rules (of the type symport/antiport) with length at most k. On the one hand, the models $\mathcal{TDC}(1)$, $\mathcal{TSC}(1)$ and $\mathcal{TSC}(2)$ are non-efficient (see [19,20] for details). On the other hand, the models $\mathcal{TDC}(k)$, for $k \geq 2$, and $\mathcal{TSC}(k)$ for $k \geq 3$, are presumably efficient (see [21,22] for details). We call a membrane system *cooperative* if it has at least one rule that needs two objects to be fired. In this case, P systems from $\mathcal{TDC}(1)$ and $\mathcal{TSC}(1)$ are non-cooperative membrane systems, while P systems from $\mathcal{TDC}(k)$ and $\mathcal{TSC}(k)$ ($k \geq 2$) are cooperative membrane systems. Let $X_1 = (I_{X_1}, \theta_{X_1})$ and $X_2 = (I_{X_1}, \theta_{X_1})$ be decision problems, the product problem of X_1 and X_2, denoted by $X_1 \otimes X_2$, is defined as follows:

$$\theta_{X_1 \otimes X_2}(u_1 + u_2) = \begin{cases} 1 & \textit{iff } \theta_{X_1}(u_1) = 1 \wedge \theta_{X_2}(u_2) = 1 \\ 0 & \textit{otherwise} \end{cases},$$

being $u_1 \in I_{X_1}$ and $u_2 \in I_{X_2}$

The main contribution of this paper is to provide a lower bound ($\mathbf{DP} \cup \mathbf{co\text{-}DP}$) for $\mathbf{PMC}_{\mathcal{R}}$ thinner than $\mathbf{NP} \cup \mathbf{co\text{-}NP}$, in the case that $\mathcal{R}$ is a class of recognizer cooperative tissue-like P systems.

Theorem 1. *Let $\mathcal{R}$ be a computing model of recognizer cooperative tissue-like P systems. If X_1 and X_2 are decision problems belonging to the time complexity class $\mathbf{PMC}_{\mathcal{R}}$, then $X_1 \otimes X_2 \in \mathbf{PMC}_{\mathcal{R}}$.*

Proof. For $i = 1, 2$, let $\mathbf{\Pi}^{(i)} = \{\Pi^{(i)}(t) \mid t \in \mathbb{N}\}$ a family of recognizer tissue-like P systems from $\mathcal{R}$ solving X_i in polynomial-time in a uniform way. Let (cod_i, s_i) be a polynomial encoding from X_i into $\mathbf{\Pi}^{(i)}$. Let $p_i(n)$ be a polynomial function such that for each instance u_i from X_i, any computation of $\Pi^{(i)}(s_i(u_i)) + cod_i(u_i)$ performs at most $p_i(|u_i|)$ steps. Then, after $n = \max\{p_1(|u_1|), p_2(|u_2|)\}$ transition steps, both answers of systems $\Pi^{(1)}(s_1(u_1)) + cod_1(u_1)$ and $\Pi^{(2)}(s_2(u_2)) + cod_2(u_2)$ are provided in the environment.

In this situation, a family $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ of membrane systems from $\mathcal{R}$ will be defined from $\mathbf{\Pi}^{(1)}$ and $\mathbf{\Pi}^{(2)}$, in such a manner that $\mathbf{\Pi}$ provides a uniform and polynomial-time solution to the product problem $X_1 \otimes X_2$.

First, a pair of polynomial-time functions (cod, s) over the set of instances of $X_1 \otimes X_2$ is considered as follows: (a) $cod(u_1, u_2) = cod_1(u_1) + cod_2(u_2)$; and (b) $s(u_1, u_2) = \langle s_1(u_1), s_2(u_2) \rangle$, for each $(u_1, u_2) \in I_{X_1 \otimes X_2}$, where $\langle \cdot, \cdot \rangle$ denotes the Cantor pairing function, a (bijective and recursive) mapping from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$ defined as $\langle t_1, t_2 \rangle = [(t_1 + t_2) \cdot (t_1 + t_2 + 1)/2] + t_1$. The idea is that the instance (u_1, u_2) will be

processed by the system $\Pi(s(u_1, u_2))$ with input multiset $cod(u_1, u_2)$. This system will be denoted by $\Pi(\langle s_1(u_1), s_2(u_2)\rangle) + cod(u_1, u_2)$.

In the following, the membrane system $\Pi(\langle s_1(u_1), s_2(u_2)\rangle)$ will be explicitly defined from $\Pi^{(1)}(s_1(u_1))$ and $\Pi^{(2)}(s_2(u_2))$, for each instance (u_1, u_2) of the product problem $X_1 \otimes X_2$, so that: a computation of $\Pi(\langle s_1(u_1), s_2(u_2)\rangle) + cod(u_1, u_2)$ is an accepting one if and only if there exist accepting computations of the systems $\Pi^{(1)}(s_1(u_1)) + cod_1(u_1)$ and $\Pi^{(2)}(s_2(u_2)) + cod_2(u_2)$. Consequently, we will have: (a) if there exist accepting computations of the system $\Pi(\langle s_1(u_1), s_2(u_2)\rangle) + cod(u_1, u_2)$ then $\theta_{X_1 \otimes X_2} = 1$; and (b) if $\theta_{X_1 \otimes X_2} = 1$ then any computation of $\Pi(\langle s_1(u_1), s_2(u_2)\rangle) + cod(u_1, u_2)$ is an accepting one. According to this, (cod, s) will be a polynomial encoding from the product problem $X_1 \otimes X_2$ into the family $\mathbf{\Pi}$. It is worth pointing out that, in fact, only membrane systems $\Pi(t)$ such that $t \in range(s)$ are defined, and this is enough to achieve our goal.

In what follows, we will denote $t = \langle s_1(u_1), s_2(u_2)\rangle$, $t_1 = s_1(u_1)$ and $t_2 = s_2(u_2)$. Let us recall that given an arbitrary natural number $t \in \mathbb{N}$ there exists a unique pair of natural numbers t_1, t_2 such that $t = \langle t_1, t_2\rangle$. The underlying idea of the construction of $\Pi(t)$ is the following: when $cod(u_1, u_2)$ is considered as the input multiset of this system, the multisets $cod_1(u_1)$ and $cod_2(u_2)$ are sent to the corresponding input cells of systems $\Pi^{(1)}(t_1)$ and $\Pi^{(2)}(t_2)$, respectively. Then, computations of $\Pi^{(1)}(t_1) + cod_1(u_1)$ and $\Pi^{(2)}(t_2) + cod_2(u_2)$ will be simulated. The answers from both systems will be provisionally placed in a cell where the final decision is taken: accepting computations of $\Pi(t) + cod(u_1, u_2)$ only come from both accepting computations of $\Pi^{(1)}(t_1) + cod_1(u_1)$ and $\Pi^{(2)}(t_2) + cod_2(u_2)$.

Without loss of generality, we can assume that $\Gamma^{(i)}(t_i) \setminus \{\mathtt{yes}, \mathtt{no}\}$, for $i = 1, 2$, are mutually disjoint, being $\Gamma^{(i)}(t_i)$ the working alphabet of $\Pi^{(i)}(t_i)$. Likewise, we can assume the same with the corresponding sets of labels $H^{(1)}(t_1)$ and $H^{(2)}(t_2)$.

The syntactical ingredients of $\Pi(t)$ are the following:

- *Cells*. The cells of $\Pi(t)$ are the cells of $\Pi^{(1)}(t_1)$ and $\Pi^{(2)}(t_2)$, plus three additional cells.
- *Working alphabet*. The working alphabet is $\Gamma^{(1)}(t) \cup \Gamma^{(2)}(t) \cup \Gamma^{(3)}(t)$, where:

$$\begin{array}{lcl} \Gamma^{(1)}(t) & = & [\Gamma^{(1)}(t_1) \setminus \{\mathtt{yes}, \mathtt{no}\}] \cup \{\mathtt{yes}_1, \mathtt{no}_1\} \\ \Gamma^{(2)}(t) & = & [\Gamma^{(2)}(t_2) \setminus \{\mathtt{yes}, \mathtt{no}\}] \cup \{\mathtt{yes}_2, \mathtt{no}_2\} \\ \Gamma^{(3)}(t) & = & \{a' \mid a \in \Gamma^{(1)}(t_1) \cup \Gamma^{(2)}(t_2)\}. \end{array}$$

- *Input alphabet*. The input alphabet is $\Sigma(t) = \Sigma^{(1*)}(t_1) \cup \Sigma^{(2*)}(t_2)$, being $\Sigma^{(i*)}(t_i)$ the input alphabet of $\Pi^{(i)}(t_i)$, by replacing objects $\mathtt{yes}$ and $\mathtt{no}$ by objects $\mathtt{yes}_i$ and $\mathtt{no}_i$, for $i = 1, 2$, respectively.
- *Alphabet of the environment*. The alphabet $\mathcal{E}$ of the environment of $\Pi(t)$ is $\mathcal{E}_1 \cup \mathcal{E}_2$, where $\mathcal{E}_i$ is the alphabet of the environment of $\Pi^{(i)}(t_i)$, for $i = 1, 2$.
- *Set of labels*. The set of labels is $H^{(1)}(t_1) \cup H^{(2)}(t_2) \cup \{aux_0, aux_+, aux_-\}$, where aux_0, aux_+, aux_- are different from each other and none of them belong to the set $H^{(1)}(t_1) \cup H^{(2)}(t_2)$. Specifically, aux_0, aux_+, aux_- will be the labels associated with the three new cells.
- *Initial multisets*. The initial multisets of $\Pi(t)$ are the following:
  - (a) For $i = 1, 2$, if h is the label of a cell from $\Pi^{(i)}(t_i)$ whose initial multiset is $\mathcal{M}_h(t_i)$ then the initial multiset associated with h in $\Pi(t)$ is $\mathcal{M}_h(t) = \{a' \mid a \in \mathcal{M}_h(t_i)\}$, that is, for each object a initially placed in cells of $\Pi^{(i)}(t_i)$, a primed version a' is considered instead when $\Pi^{(i)}(t_i)$ is taken as a part of $\Pi(t)$.
  - (b) $\mathcal{M}_{aux_0}(t) = \mathcal{M}(t_1) \cup \mathcal{M}(t_2) \cup \{\mathtt{yes}, \mathtt{no}\}$,being $\mathcal{M}(t_i)$ the union of the initial multisets of $\Pi^{(i)}(t_i)$.
  - (c) $\mathcal{M}_{aux_+}(t) = \mathcal{M}_{aux_-}(t) = \emptyset$.
- *Set of rules*. The set of rules is $\mathcal{R}_{\Pi^{(1)}(t_1)} \cup \mathcal{R}_{\Pi^{(2)}(t_2)} \cup \mathcal{R}^*_t$, where $\mathcal{R}_{\Pi^{(i)}(t_i)}$ is the set of rules of $\Pi^{(i)}(t_i)$, for $i = 1, 2$, obtained through the replacement of objects $\mathtt{yes}$ and $\mathtt{no}$

by objects $\mathtt{yes}_i$ and $\mathtt{no}_i$, respectively, in each rule. In addition, $\mathcal{R}_t^*$ is the following set of rules:

1. *Rules for simultaneously transporting $cod_i(u_i)$ from cell labeled by aux_0 to the input cell $in_{\Pi^{(i)}_{(t_i)}}$ of $\Pi^{(i)}_{(t_i)}$, for $i = 1, 2$.*
   1.1. $(aux_0\,,\, a \,/\, \lambda\,,\, in_{\Pi^{(1)}(t_1)})$, for each $a \in \Sigma^{(1)}(t_1)$.
   1.2. $(aux_0\,,\, a \,/\, \lambda\,,\, in_{\Pi^{(2)}(t_2)})$, for each $a \in \Sigma^{(2)}(t_2)$.
2. *Rules to obtain the rules from $\Pi^{(i)}(t_i)$, for $i = 1, 2$, started in the second transition step.*
   2.1. $(h\,,\, a' \,/\, a\,,\, aux_0)$, for each $a \in [\Gamma^{(1)}(t_1) \cup \Gamma^{(2)}(t_2)] \setminus [\Sigma^{(1)}(t_1) \cup \Sigma^{(2)}(t_2)]$ and for each label h of a cell in $\Pi^{(1)}(t_1) \cup \Pi^{(2)}(t_2)$.
3. *Rules for transporting the pair of answers of the systems $\Pi^{(i)}(t_i)$, for $i = 1, 2$, from the environment to cell aux_+ or aux_-.*
   3.1. $(env\,,\, \mathtt{yes}_1\, \mathtt{yes}_2 \,/\, \lambda\,,\, aux_+)$.
   3.2. $(env\,,\, \mathtt{yes}_1\, \mathtt{no}_2 \,/\, \lambda\,,\, aux_-)$.
   3.3. $(env\,,\, \mathtt{no}_1\, \mathtt{yes}_2 \,/\, \lambda\,,\, aux_-)$.
   3.4. $(env\,,\, \mathtt{no}_1\, \mathtt{no}_2 \,/\, \lambda\,,\, aux_-)$.
4. *Rules for the affirmative answer of the system $\Pi(t) + cod(u_1, u_2)$:*
   4.1. $(aux_+\,,\, \mathtt{yes}_1 \,/\, \mathtt{yes}\,,\, aux_0)$.
   4.2. $(aux_+\,,\, \mathtt{yes} \,/\, \lambda\,,\, env)$.
5. *Rules for the negative answer of the system $\Pi(t) + cod(u_1, u_2)$:*
   5.1. $(aux_-\,,\, \mathtt{no}_1 \,/\, \mathtt{no}\,,\, aux_0)$.
   5.2. $(aux_-\,,\, \mathtt{no}_2 \,/\, \mathtt{no}\,,\, aux_0)$.
   5.3. $(aux_-\,,\, \mathtt{no} \,/\, \lambda\,,\, env)$.

- *Input cell*. The input cell is the cell labelled by aux_0.

The system $\Pi(t)$ can be graphically depicted as in Figure 1.

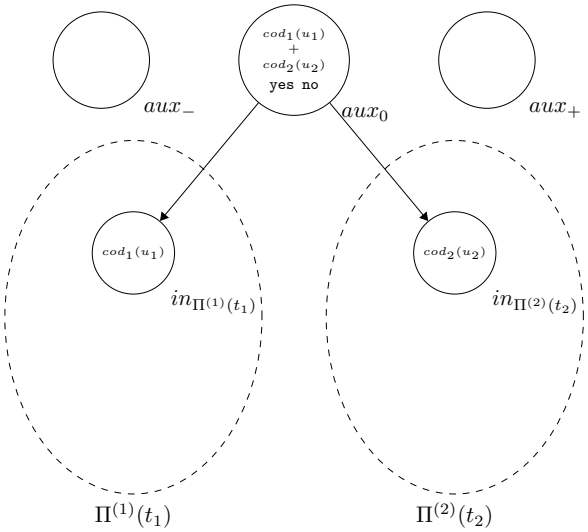

Figure 1. System $\Pi(t)$ described from systems $\Pi^{(1)}(t_1)$ and $\Pi^{(2)}(t_2)$.

□

An Overview of the Computations of $\Pi(t) + cod(u_1, u_2)$

The proposed solution can be structured in the following stages:

- *Transport stage*
  Once the input multiset $cod(u_1, u_2) = cod_1(u_1) + cod_2(u_2)$ is supplied to the input cell aux_0 of the system $\Pi(t)$, by applying the rules from **1.1**, and **1.2**, in the first computation step multisets $cod_1(u_1)$ and $cod_2(u_2)$ enter into the input cell $in_{\Pi^{(1)}(t_1)}$ and $in_{\Pi^{(2)}(t_2)}$, respectively. Simultaneously, in this first step, by applying the rules from

2.1, objects a' initially placed in each cell of $\Pi^{(i)}(t_i)$ is transformed in the corresponding non-primed object a. This stage takes only one transition step.

- *Simulation stage*
  Starting at the second transition step, computations of the system $\Pi^{(i)}(t_i)$ with input multiset $cod_i(u_i)$ are simulated by applying the corresponding rules from $\Pi^{(i)}(t_i)$, for $i = 1, 2$. Then, after at most $1 + \max\{p_1(|u_1|), p_2(|u_2|)\}$ transition steps, both systems $\Pi^{(1)}(t_1) + cod_1(u_1)$ and $\Pi^{(2)}(t_2) + cod_2(u_2)$ will send their answers to the environment. Therefore, this stage takes at most $1 + n$ steps.
- *Output stage*
  The system $\Pi(t)$ with input multiset $cod(u_1, u_2)$ sends the right answer to the environment according to the results obtained in the previous stage. Bearing in mind that rules **3.1**, **3.2**, **3.3** and **3.4** are cooperative, they will only be applicable when the environment receive the answers from $\Pi^{(1)}(t_1) + cod_1(u_1)$ and $\Pi^{(2)}(t_2) + cod_2(u_2)$. Let us assume that at instant k both answers reach the environment (obviously, k depends on the computations selected in the simulations and $k \leq 1 + n$).
  - *Affirmative answer*.
    In this case, the answers of $\Pi^{(1)}(t_1) + cod_1(u_1)$ and $\Pi^{(2)}(t_2) + cod_2(u_2)$ will be $\mathtt{yes}_1$ and $\mathtt{yes}_2$, respectively. By applying rule **3.1**, objects $\mathtt{yes}_1$ and $\mathtt{yes}_2$ are sent to cell aux_+, that is $\mathcal{C}_{k+1}(aux_+) = \{\mathtt{yes}_1, \mathtt{yes}_2\}$. By applying rule **4.1** we will have $\mathcal{C}_{k+2}(aux_+) = \{\mathtt{yes}, \mathtt{yes}_2\}$ and $\mathcal{C}_{k+2}(aux_0) = \{\mathtt{yes}_1, \mathtt{no}\}$. Finally, by applying rule **4.2** the system send out to the object yes and the system halts. It is graphically depicted in Figure 2.
  - *Negative answer*.
    In this case, at least one answer of $\Pi^{(1)}(t_1) + cod_1(u_1)$ or $\Pi^{(2)}(t_2) + cod_2(u_2)$ must be negative. By applying rule **3.2** or **3.3** or **3.4**, cell aux_- from configuration $\mathcal{C}_{k+1}$ will contain object $\mathtt{no}_1$ either object $\mathtt{no}_2$ either both of them. By applying rule **5.1** or rule **5.2** (only one of them because cell aux_0 from configuration $\mathcal{C}_{k+1}$ only contain one object no) will have $\mathtt{no} \in \mathcal{C}_{k+2}(aux_-)$. Finally, by applying rule **5.3** the system send out to the object no and the system halts. It is graphically depicted in Figure 3.

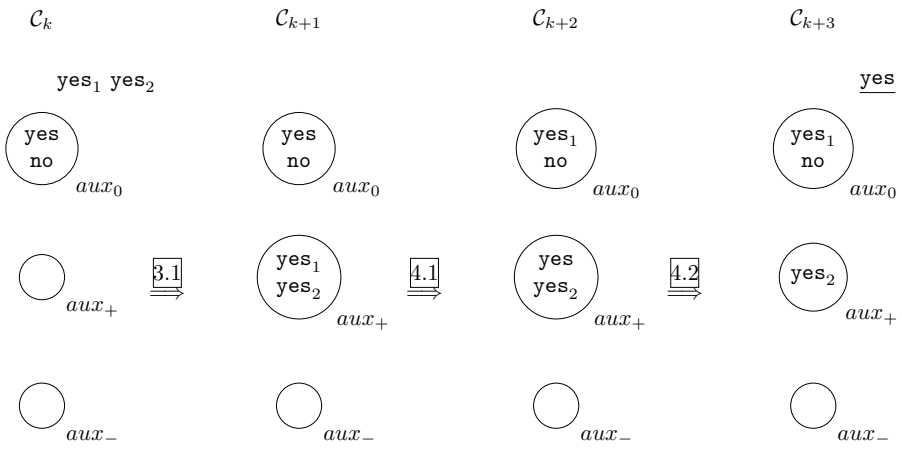

Figure 2. The answer of $\Pi^{(1)}(t_1)$ is yes and the answer of $\Pi^{(2)}(t_2)$ is yes.

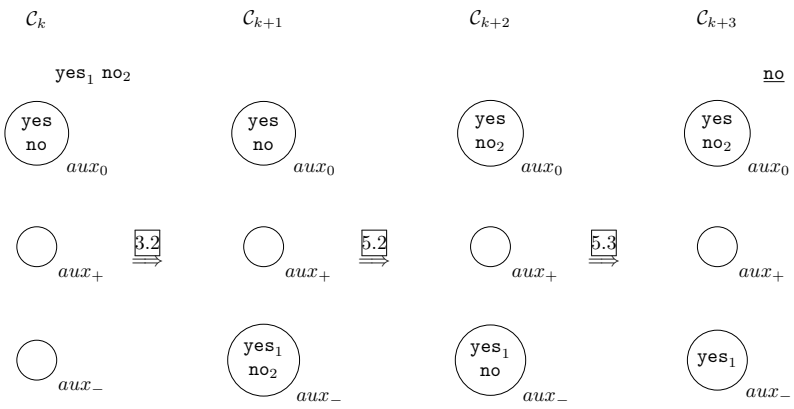

Figure 3. The answer of $\Pi^{(1)}(t_1)$ is `yes` and the answer of $\Pi^{(2)}(t_2)$ is `no`.

Remark 1. *Concerning the proof of Theorem 1, we would like to draw attention to several aspects:*

(a) Recognizer cooperative tissue-like membrane systems from class $\mathcal{R}$ are required to allow the use of communication rules with the length of at least 2, and the length of the new rules added is exactly 2.

(b) A explicit solution of the product problem product is provided from two respective solutions of the problems involved in that operation.

Corollary 1. *If $\mathcal{R}$ is a presumably efficient computing model of a recognizer tissue-like membrane system, then* **DP** $\cup$ **co-DP** $\subseteq$ **PMC**$_{\mathcal{R}}$.

Proof. First, let us note that if $\mathcal{R}$ is a presumably efficient computing model of a recognizer tissue-like membrane system and **P** $\neq$ **NP**, then systems from $\mathcal{R}$ are cooperative (see [19,20] for details). Let X be an **NP** complete problem such that $X \in$ **PMC**$_{\mathcal{R}}$. Then, the complement problem $\overline{X}$ is a **co-NP** complete problem such that belongs to class **PMC**$_{\mathcal{R}}$. From Theorem 1, we deduce that $X \otimes \overline{X} \in$ **PMC**$_{\mathcal{R}}$. Since the product problem $X \otimes \overline{X}$ is a **DP** complete problem and the complexity class **PMC**$_{\mathcal{R}}$ is closed under complement and under polynomial-time reduction, **DP** $\cup$ **co-DP** $\subseteq$ **PMC**$_{\mathcal{R}}$ follows. □

Remark 2. *According with the proofs of Theorem 1 and Corollary 1, a polynomial time and uniform solution to a* **DP***-complete problem by means of a family of recognizer cooperative tissue-like membrane systems, can be* ***explicitly designed*** *from a given polynomial time and uniform solution to a* **NP***-complete problem. In fact, given an* **NP***-complete problem X such that $X \in$ **PMC**$_{\mathcal{R}}$, where $\mathcal{R}$ is a class of presumably efficient computing model of a recognizer tissue-like membrane system, a* **DP***-complete problem, the product problem $X \otimes \overline{X}$, is associated with it, in such a manner that if $X \in$ **PMC**$_{\mathcal{R}}$ then $X \otimes \overline{X} \in$ **PMC**$_{\mathcal{R}}$, for such kind class $\mathcal{R}$ of recognizer tissue-like systems.*

3. Solving DP-Complete Problems by Using a New Methodology

In this section, the main Theorem is used to provide solutions to some **DP**-complete problems obtained from specific solutions to **NP**-complete problems in the framework of recognizer cooperative tissue-like membrane systems. In particular we use the efficient solution to the `SAT` problem by means of a family of recognizer tissue P systems with symport/antiport rules with length at most 2 and division rules from [23], that is based on the classical generation-checking-output workflow. As mentioned in the methodology, given a family of recognizer P systems $\mathbf{\Pi^{(1)}}$ from $\mathcal{R}$ efficiently solving a problem X from the complexity class **C**, then a family of recognizer P systems $\mathbf{\Pi^{(2)}}$ from $\mathcal{R}$ that solves the complementary problem of X, being $\overline{X} \in$ **co** $-$ **C**.

The `SAT-UNSAT` problem is defined as follows: Let φ_1 (respectively, φ_2) be a propositional logic formula with n_1 variables (resp., n_2 variables) and p_1 clauses (resp., p_2 clauses). We say that the pair (φ_1, φ_2) is true if and only if φ_1 is satisfiable and φ_2 is unsatisfi-

able. Formally, let $\mathtt{SAT}-\mathtt{UNSAT}=(I_{\mathtt{SAT}-\mathtt{UNSAT}},\theta_{\mathtt{SAT}-\mathtt{UNSAT}})$, where $I_{\mathtt{SAT}-\mathtt{UNSAT}}=\{(\varphi_i,\varphi_2)\mid \varphi_1,\varphi_2$ are propositional logic formulae in CNF and in a simplified form$\}$.

$$\theta_{\mathtt{SAT}-\mathtt{UNSAT}}((\varphi_1,\varphi_2))=\begin{cases}1 & \textit{iff}\ \varphi_1\in L_{\mathtt{SAT}}\wedge\varphi_2\in L_{\mathtt{UNSAT}}\\ 0 & \textit{otherwise}\end{cases}$$

Let $\Gamma^{(1)}$ be the working alphabet of $\mathbf{\Pi^{(1)}}$ and $\Gamma^{(2)}$ be the working alphabet of $\mathbf{\Pi^{(2)}}$. Given that we need that $\Gamma^{(1)}\cap\Gamma^{(2)}=\emptyset$, we let $\Gamma^{(2)}=\{\mathbf{a}\mid a\in\Gamma^{(1)}\}$; that is, objects from $\Gamma^{(2)}$ will be boldfaced versions of objects from $\Gamma^{(1)}$. In $\Gamma^{(1)}$ (respectively, $\Gamma^{(2)}$) we will change objects $\mathtt{yes}$ and $\mathtt{no}$ by objects $\mathtt{yes}_1$ and $\mathtt{no}_1$ (resp., $\mathtt{yes}_2$ and $\mathtt{no}_2$). In addition, let $H^{(1)}$ be the working alphabet of $\mathbf{\Pi^{(1)}}$. The set of labels $H^{(2)}$ of $\mathbf{\Pi^{(2)}}$ will be defined as $H^{(2)}=\{\mathbf{h}\mid h\in H^{(1)}\}$. For each quadruple of natural numbers $n_1,p_1,n_2,p_2\in\mathbb{N}$, we will consider the recognizer tissue P system with cell division and symport/antiport rules of length at most 2.

$$\Pi(\langle\langle n_1,p_1\rangle,\langle n_2,p_2\rangle\rangle)=(\Gamma,\mathcal{E},\Sigma,\mathcal{M}_1,\ldots,\mathcal{M}_{n_1p_1+3},\mathcal{M}_{\mathbf{1}},\ldots,\mathcal{M}_{\mathbf{n_2p_2+3}},\mathcal{M}_{aux_+},\mathcal{M}_{aux_-},\mathcal{M}_{aux_0},\mathcal{R},i_{in},i_{out})$$

of degree $n_1p_1\mathbf{n_2p_2}+6$ defined as follows:

(a)

$$\begin{aligned}\Gamma &= \Gamma'\cup\{a'\mid a\in\Gamma'\},\\ \Gamma' &= \Sigma\cup\mathcal{E}\cup\{\mathtt{yes},\mathtt{no}\}\cup\{\mathtt{yes}_1,\mathtt{no}_1,\alpha,\beta_0,\gamma_0\}\cup\{c_j\mid 1\le k\le p_1\}\cup\\ &\quad\{\alpha_k\mid 0\le k\le n_1p_1-1\}\cup\{a_{i,j}\mid 1\le i\le n_1,1\le j\le p_1\}\cup\\ &\quad\{T_{i,j},F_{i,j}\mid 1\le i\le n_1,1\le j\le p_1\}\cup\\ &\quad\{x_{i,j,k},\overline{x}_{i,j,k}\mid 1\le i\le n_1,1\le j\le p_1,0\le k\le n_1p_1\}\cup\\ &\quad\{\mathtt{yes}_2,\mathtt{no}_2,\mathbf{ff},\mathbf{fi_0},\mathbf{fl_0}\}\cup\{\mathbf{c_j}\mid 1\le k\le p_2\}\cup\\ &\quad\{\mathbf{ff_k}\mid 0\le k\le n_2p_2-1\}\cup\{\mathbf{a_{i,j}}\mid 1\le i\le n_2,1\le j\le p_2\}\cup\\ &\quad\{\mathbf{T_{i,j}},\mathbf{F_{i,j}}\mid 1\le i\le n_2,1\le j\le p_2\}\cup\\ &\quad\{\mathbf{x_{i,j,k}},\overline{\mathbf{x}}_{\mathbf{i,j,k}}\mid 1\le i\le n_2,1\le j\le p_2,0\le k\le n_2p_2\}\end{aligned}$$

(b)

$$\begin{aligned}\mathcal{E} &= \{\alpha_k\mid n_1p_1\le k\le 2n_1p_1+2\}\cup\{\beta_k\mid 1\le k\le 2n_1p_1+4\}\cup\\ &\quad\{\gamma_k\mid 1\le k\le 2n_1p_1+5\}\cup\\ &\quad\{x_{i,j,k},\overline{x}_{i,j,k}\mid 1\le i\le n_1,1\le j\le p_1,n_1p_1+1\le k\le 2n_1p_1\}\cup\\ &\quad\{\mathbf{ff_k}\mid n_2p_2\le k\le 2n_2p_2+2\}\cup\{\mathbf{fi_k}\mid 1\le k\le 2n_2p_2+4\}\cup\\ &\quad\{\mathbf{fl_k}\mid 1\le k\le 2n_2p_2+5\}\cup\\ &\quad\{\mathbf{x_{i,j,k}},\overline{\mathbf{x}}_{\mathbf{i,j,k}}\mid 1\le i\le n_2,1\le j\le p_2,n_2p_2+1\le k\le 2n_2p_2\}\end{aligned}$$

(c)

$$\begin{aligned}\Sigma &= \{x_{i,j,0},\overline{x}_{i,j,0}\mid 1\le i\le n_1,1\le j\le p_1\}\cup\\ &\quad\{\mathbf{x_{i,j,0}},\overline{\mathbf{x}}_{\mathbf{i,j,0}}\mid 1\le i\le n_2,1\le j\le p_2\}\end{aligned}$$

(d) $\mathcal{M}_i=\emptyset$ for $i\in\{aux_+,aux_-\}$, $\mathcal{M}_{aux_0}=\{\beta_0^*,\gamma_0^*\}\cup\{a_{i,j}^*\mid 1\le i\le n_1,1\le j\le p_1\}\cup\{c_j^*\mid 1\le j\le p_1\}\cup\{\alpha^*\}\cup\{\alpha_0^*\}\cup\{\mathbf{fi_0},\mathbf{fl_0}\}\cup\{\mathbf{a_{i,j}}\mid 1\le i\le n_2,1\le j\le p_2\}\cup\{\mathbf{c_j}\mid 1\le j\le p_2\}\cup\{\mathbf{ff}\}\cup\{\mathbf{ff_0}\}\cup\{\mathtt{yes}_1,\mathtt{no}_1,\beta_0^*,\gamma_0^*\}\cup\{a_{i,j}^*\mid 1\le i\le n_1,1\le j\le p_1\}\cup\{c_j^*\mid 1\le j\le p_1\}\cup\{\alpha^*\}\cup\{\alpha_0^*\}\cup\{\mathtt{yes}_2,\mathtt{no}_2,\mathbf{fi_0^*},\mathbf{fl_0^*}\}\cup\{\mathbf{a_{i,j}^*}\mid 1\le i\le n_2,1\le j\le p_2\}\cup\{\mathbf{c_j^*}\mid 1\le j\le p_2\}\cup\{\mathbf{ff^*}\}\cup\{\mathbf{ff_0^*}\}$, $\mathcal{M}_i=\{a'\mid a\in\mathcal{M}_i$ in the original solution$\}$, $\mathcal{M}_{\mathbf{i}}=\{\mathbf{a}'\mid a\in\mathcal{M}_i$ in the original solution$\}$

(e) The set of rules $\mathcal{R}$ is the following:

$$\left.\begin{array}{l}[\,a_{i,j}\,]_{n_1p_1+2}\to[\,T_{i,j}\,]_{n_1p_1+2}[\,F_{i,j}\,]_{n_1p_1+2}\\ (n_1p_1+2,T_{i,j}F_{i,j'},/\lambda,env)\end{array}\right\}\text{for } 1\le i\le n_1,1\le j,j'\le p_1$$

$$\left.\begin{array}{l}(n_1p_1+1,x_{i,j,0}/\lambda,i+n_1\cdot(j-1))\\ (n_1p_1+1,\overline{x}_{i,j,0}/\lambda,i+n_1\cdot(j-1))\end{array}\right\}\text{for } \begin{array}{l}1\le i\le n_1,\\ 1\le j\le p_1\end{array}$$

$$\left.\begin{array}{l}[\,x_{i,j,k}\,]_{i+n_1\cdot(j-1)} \to [\,x_{i,j,k+1}\,]_{i+n_1\cdot(j-1)}[\,x_{i,j,k+1}\,]_{i+n_1\cdot(j-1)}\\ [\,\overline{x}_{i,j,k}\,]_{i+n_1\cdot(j-1)} \to [\,\overline{x}_{i,j,k+1}\,]_{i+n_1\cdot(j-1)}[\,\overline{x}_{i,j,k+1}\,]_{i+n_1\cdot(j-1)}\end{array}\right\}\text{for}\ \begin{array}{l}1\le i\le n_1,\\ 1\le j\le p_1,\\ 0\le k\le n_1p_1-1\end{array}$$

$$\left.\begin{array}{l}(i+n_1\cdot(j-1), x_{i,j,k}/x_{i,j,k+1}, env)\\ (i+n_1\cdot(j-1), \overline{x}_{i,j,k}/\overline{x}_{i,j,k+1}, env)\end{array}\right\}\text{for}\ \begin{array}{l}1\le i\le n_1,\\ 1\le j\le p_1,\\ n_1p_1\le k\le 2n_1p_1-1\end{array}$$

$$\left.\begin{array}{l}(n_1p_1+2, T_{i,j}/x_{i,j,2n_1p_1}, i+n_1\cdot(j-1))\\ (n_1p_1+2, F_{i,j}/\overline{x}_{i,j,2n_1p_1}, i+n_1\cdot(j-1))\end{array}\right\} for\ \begin{array}{l}1\le i\le n_1,\\ 1\le j\le p_1\end{array}$$

$$\left.\begin{array}{l}(n_1p_1+2, c_jx_{i,j,2n_1p_1}/\lambda, env)\\ (n_1p_1+2, c_j\overline{x}_{i,j,2n_1p_1}/\lambda, env)\end{array}\right\} for\ \ 1\le i\le n_1, 1\le j\le p_1$$

$$\left.[\,\alpha_k\,]_{n_1p_1+3} \to [\,\alpha_{k+1}\,]_{n_1p_1+3}\,[\,\alpha_{k+1}\,]_{n_1p_1+3}\ \right\} for\ \ 0\le k\le n_1p_1-1$$

$$(n_1p_1+3, \alpha_{n_1p_1+k}/\alpha_{n_1p_1+k+1}, env)\ \ for\ \ 0\le k\le n_1p_1+1$$

$$(n_1p_1+1, \beta_k/\beta_{k+1}, env)\ \ for\ \ 0\le k\le 2n_1p_1+3$$

$$(n_1p_1+1, \gamma_k/\gamma_{k+1}, env)\ \ for\ \ 0\le k\le 2n_1p_1+4$$

$$(n_1p_1+2, \alpha/\alpha_{2n_1p_1+2}, n_1p_1+3)$$

$$(n_1p_1+2, \alpha_{2n_1p_1+2}c_j/\lambda, env)\ \ for\ \ 1\le j\le p_1$$

$$(n_1p_1+1, \beta_{2n_1p_1+4}, \gamma_{2n_1p_1+5}/\lambda, n_1p_1+3)$$

$$(n_1p_1+1, \mathtt{no}_1/\beta_{2n_1p_1+4}, n_1p_1+3)$$

$$(n_1p_1+3, \mathtt{no}_1/\lambda, env)$$

$$(n_1p_1+1, \beta_{2n_1p_1+4}/\alpha_{2n_1p_1+2}, n_1p_1+2)$$

$$(n_1p_1+1, \alpha_{2n_1p_1+2}\mathtt{yes}_1/\lambda, env)$$

$$\left.\begin{array}{l}[\,\mathbf{a_{i,j}}\,]_{\mathbf{n_2p_2+2}} \to [\,\mathbf{T_{i,j}}\,]_{\mathbf{n_2p_2+2}}[\,\mathbf{F_{i,j}}\,]_{\mathbf{n_2p_2+2}}\\ (\mathbf{n_2p_2+2, T_{i,j}F_{i,j'}}, /\lambda, env)\end{array}\right\}\text{for}\ \ \mathbf{1\le i\le n_2, 1\le j, j'\le p_2}$$

$$\left.\begin{array}{l}(\mathbf{n_2p_2+1, x_{i,j,0}}/\lambda, \mathbf{i+n_2\cdot(j-1)})\\ (\mathbf{n_2p_2+1}, \overline{\mathbf{x}}_{\mathbf{i,j,0}}/\lambda, \mathbf{i+n_2\cdot(j-1)})\end{array}\right\}\text{for}\ \begin{array}{l}\mathbf{1\le i\le n_2},\\ \mathbf{1\le j\le p_2}\end{array}$$

$$\left.\begin{array}{l}[\,\mathbf{x_{i,j,k}}\,]_{\mathbf{i+n_2\cdot(j-1)}} \to [\,\mathbf{x_{i,j,k+1}}\,]_{\mathbf{i+n_2\cdot(j-1)}}[\,\mathbf{x_{i,j,k+1}}\,]_{\mathbf{i+n_2\cdot(j-1)}}\\ [\,\overline{\mathbf{x}}_{\mathbf{i,j,k}}\,]_{\mathbf{i+n_2\cdot(j-1)}} \to [\,\overline{\mathbf{x}}_{\mathbf{i,j,k+1}}\,]_{\mathbf{i+n_2\cdot(j-1)}}[\,\overline{\mathbf{x}}_{\mathbf{i,j,k+1}}\,]_{\mathbf{i+n_2\cdot(j-1)}}\end{array}\right\}\text{for}\ \begin{array}{l}\mathbf{1\le i\le n_2},\\ \mathbf{1\le j\le p_2},\\ \mathbf{0\le k\le n_2p_2-1}\end{array}$$

$$\left.\begin{array}{l}(\mathbf{i+n_2\cdot(j-1), x_{i,j,k}/x_{i,j,k+1}}, env)\\ (\mathbf{i+n_2\cdot(j-1)}, \overline{\mathbf{x}}_{\mathbf{i,j,k}}/\overline{\mathbf{x}}_{\mathbf{i,j,k+1}}, env)\end{array}\right\}\text{for}\ \begin{array}{l}\mathbf{1\le i\le n_2},\\ \mathbf{1\le j\le p_2},\\ \mathbf{n_2p_2\le k\le 2n_2p_2-1}\end{array}$$

$$\left.\begin{array}{l}(\mathbf{n_2p_2+2, T_{i,j}/x_{i,j,2n_2p_2}, i+n_2\cdot(j-1)})\\ (\mathbf{n_2p_2+2, F_{i,j}/\overline{x}_{i,j,2n_2p_2}, i+n_2\cdot(j-1)})\end{array}\right\} for\ \begin{array}{l}\mathbf{1\le i\le n_2},\\ \mathbf{1\le j\le p_2}\end{array}$$

$$\left.\begin{array}{l}(\mathbf{n_2p_2+2, c_jx_{i,j,2n_2p_2}}/\lambda, env)\\ (\mathbf{n_2p_2+2, c_j}\overline{\mathbf{x}}_{\mathbf{i,j,2n_2p_2}}/\lambda, env)\end{array}\right\} for\ \ \mathbf{1\le i\le n_2, 1\le j\le p_2}$$

$$\left.[\,\mathbf{ff_k}\,]_{\mathbf{n_2p_2+3}} \to [\,\mathbf{ff_{k+1}}\,]_{\mathbf{n_2p_2+3}}\,[\,\mathbf{ff_{k+1}}\,]_{\mathbf{n_2p_2+3}}\ \right\} for\ \ \mathbf{0\le k\le n_2p_2-1}$$

$$(\mathbf{n_2p_2+3, ff_{n_2p_2+k}/ff_{n_2p_2+k+1}}, env)\ \ for\ \ \mathbf{0\le k\le n_2p_2+1}$$

$$(\mathbf{np+1, fi_k/fi_{k+1}}, env)\ \ for\ \ \mathbf{0\le k\le 2n_2p_2+3}$$

$$(\mathbf{n_2p_2+1, fl_k/fl_{k+1}}, env)\ \ for\ \ \mathbf{0\le k\le 2n_2p_2+4}$$

$$(\mathbf{n_2p_2+2, ff/ff_{2n_2p_2+2}, n_2p_2+3})$$

$$(\mathbf{n_2p_2+2, ff_{2n_2p_2+2}c_j}/\lambda, env)\ \ for\ \ \mathbf{1\le j\le p_2}$$

$$(\mathbf{n_2p_2+1, fi_{2n_2p_2+4}, fl_{2n_2p_2+5}}/\lambda, \mathbf{n_2p_2+3})$$

$$(\mathbf{n_2p_2+1}, \mathtt{yes}_2/\mathbf{fi_{2n_2p_2+4}}, \mathbf{n_2p_2+3})$$

$$(\mathbf{n_2p_2+3}, \mathtt{yes}_2/\lambda, env)$$

$$(\mathbf{n_2p_2+1, fi_{2n_2p_2+4}/ff_{2n_2p_2+2}, n_2p_2+2})$$

$$(\mathbf{n_2p_2+1, ff_{2n_2p_2+2}}\mathtt{no}_2/\lambda, env)$$

Apart from these, all rules indicated in Section 2 must be added.

(f) $i_{in} = aux_0$ and $i_{out} = env$

For details about the behavior of the original SAT solution, we refer the reader to [23].

4. Insights from the Evolution of the System

We recall that non-bold indexes and objects correspond to the indexes and objects from the solution to SAT and bold indexes and objects correspond to the indexes and objects from the solution to UNSAT. Figures 4 and 5 show the evolution of the number of cells labeled by $i + n_1 \cdot (j-1), n_1p_1 + 2$ and $n_1p_1 + 3$ (respectively, $\mathbf{i} + \mathbf{n_2} \cdot (\mathbf{j} - \mathbf{1}), \mathbf{n_2p_2 + 2}$ and $\mathbf{n_2p_2 + 3}$). First of all, it can be observed that the execution of division rules of cells labeled by $n_1p_1 + 2, n_1p_1 + 3, \mathbf{n_2p_2 + 2}$ and $\mathbf{n_2p_2 + 3}$ start in the second computational step, since the first one is reserved to the movements of objects from the cell aux_0 to their corresponding cell. In addition, since in the original solution the division of membranes labeled by $i + n_1 \cdot (j-1)$ and $\mathbf{i} + \mathbf{n_2} \cdot (\mathbf{j} - \mathbf{1})$ start at step 2, they start to be simulated at step 3 in $\Pi(\langle u_1, u_2 \rangle) + cod((u_1, u_2))$.

Since division rules of cells labeled by $n_1p_1 + 2$ and $\mathbf{n_2p_2 + 2}$ can be non-consecutive due to the application of rules $(n_1p_1 + 2, T_{i,j}F_{i,j'}, /\lambda, env)$ and $(\mathbf{n_2p_2 + 2}, \mathbf{T_{i,j}F_{i,j'}}, /\lambda, env)$, the number of membranes labeled by $n_1p_1 + 2$ and $\mathbf{n_2p_2 + 2}$ in configuration $1 + k$ can be different to 2^k.

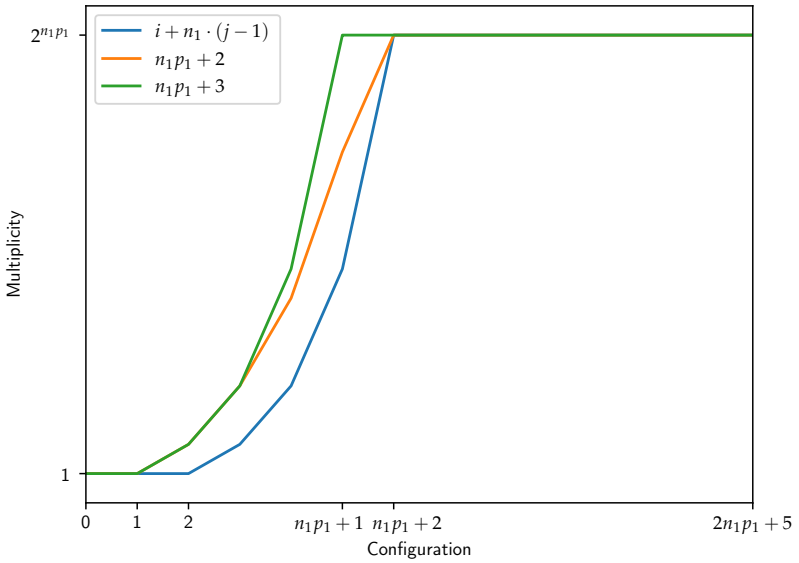

Figure 4. Evolution of the number of membranes of the SAT solution.

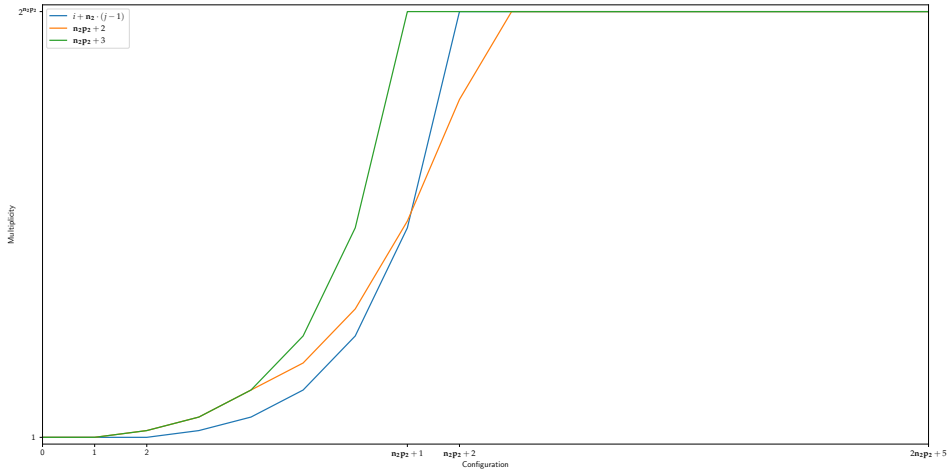

Figure 5. Evolution of membranes of the UNSAT solution.

The original solution takes $2np + 8$ steps in the negative case and $2np + 6$ in the case that there exists a truth assignment that makes true the input formula φ, being n the number of variables and p the number of clauses. In Figure 6, we can see that the number of steps taken in the computation of $\Pi(\langle u_1, u_2 \rangle) + cod((u_1, u_2))$ depends on the number of steps of the original computation. In this case, $c_1 < c_2$. In the figure, $m = max(c_1, c_2) + 4$, being c_i the computation time of $\Pi^{(i)}(t_i)$, taking into account that it takes one step for the objects to prepare both $\Pi^{(1)}(t_1)$ and $\Pi^{(2)}(t_2)$ and three steps for the output stage. The cyan (first element in legend) line lies below the red (fourth element in legend) line until $n_1p_1 + 2$ and below the green (third element in legend) line from that point.

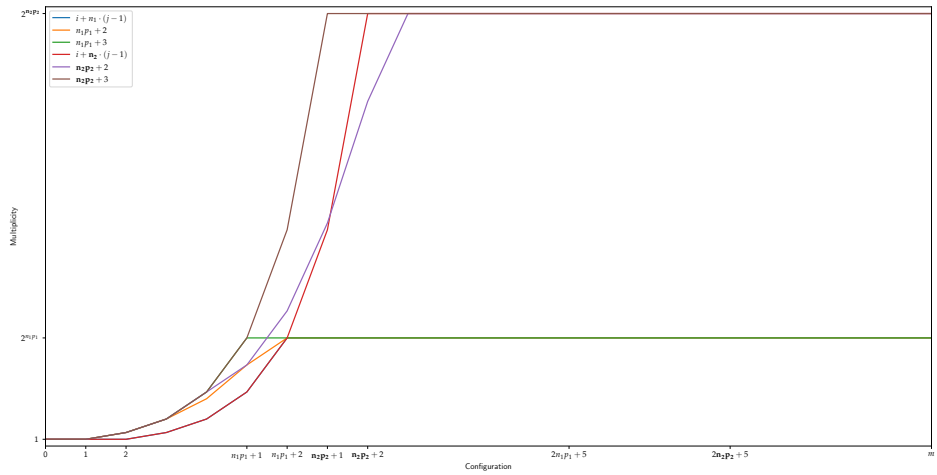

Figure 6. Global evolution of the number of membranes of the SAT − UNSAT solution.

5. Conclusions

Considering $\mathcal{R}$ as a presumed efficient computing model of recognizer tissue-like membrane systems, meaning it possesses the capability to solve **NP**-complete problems in polynomial time, the time complexity class $\mathbf{PMC}_{\mathcal{R}}$ encompasses the complexity classes **NP** and **co-NP** [9]. In this paper, we enhance this finding by presenting a specific protocol for converting a solution of an **NP**-complete problem into a solution for a **DP**-complete problem. Here, **DP** represents the set of languages that can be expressed as the difference between two languages in **NP** or, equivalently, as the intersection of a language in **NP** and a language in **co-NP**. Thus, the following result has been obtained: $\mathbf{DP} \cup \mathbf{co\text{-}DP} \subseteq \mathbf{PMC}_{\mathcal{R}}$ In fact, a more general result has been obtained: let X_1 and X_2 be two decision problems in the class $\mathbf{PMC}_{\mathcal{R}}$, being $\mathcal{R}$ a class of recognizer cooperative tissue P systems. Then, the problem $X_1 \otimes X_2$ belongs also to the class $\mathbf{PMC}_{\mathcal{R}}$. The protocol to transform the solutions to X_1 and X_2 to a solution to the problem $X_1 \otimes X_2$ takes only four extra steps with respect to the maximum number of steps of the solutions of X_1 and X_2, thus making it an efficient way to transform the solutions. Of course, this methodology cannot be applied if the tissue P systems are non-cooperative, because some objects need this behavior to move from one region to another region, this making it impossible to implement this collaboration between objects.

In [15], the authors provided an exact characterization of the class $\mathbf{PMC}_{\mathcal{TFC}}$, which is even more challenging than the result presented in this study. However, it is important to consider two key factors. Firstly, the definition of recognizer membrane systems in [15] differs from the definition given in [9]. Secondly, the strength of this paper lies not only in the lower bound but also in the automatic protocol provided for transforming a solution of a **NP**-complete problem into a **DP**-complete problem. This protocol can be readily implemented and simulated using software simulators like P-Lingua, designed specifically for membrane systems.

In [10], the authors introduced a similar methodology for transforming a solution of a **NP**-complete problem into a solution of a **DP**-complete problem within the framework of

P systems with active membranes. In their work, each object in the input multiset traversed various membranes to reach the input membrane of the corresponding subsystem, which required at least $n \cdot d$ computational steps. Here, n represented the number of objects in the input multiset, and d denoted the depth of the input membrane. In contrast, in this study, each input object reaches the corresponding input cell in a single transition. Additionally, the composite solution presented in this work utilizes three extra membranes, whereas the alternative solution in the aforementioned work employed only a single extra cell. It would be interesting to have parallel simulators for tissue P systems to enable the simulation of these composite solutions.

Author Contributions: Conceptualization, D.O.-M. and M.J.P.-J.; methodology, D.O.-M. and Á.R.-J.; software, J.A.A.-G.; validation, D.O.-M. and A.R.-d.-A.; formal analysis, D.O.-M. and M.J.P.-J.; investigation, D.O.-M. and Á.R.-J.; writing—original draft preparation, D.O.-M., A.R.-d.-A. and J.A.A.-G.; writing—review and editing, D.O.-M. and A.R.-J.; supervision, D.O.-M. and M.J.P.-J. All authors have read and agreed to the published version of the manuscript.

Funding: D. Orellana-Martín acknowledges Contratación de Personal Investigador Doctor. (Convocatoria 2019) 43 Contratos Capital Humano Línea 2. Paidi 2020, supported by the European Social Fund and Junta de Andalucía.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Păun, G. Computing with membranes. *J. Comput. Syst. Sci.* **2000**, *61*, 108–143. [CrossRef]
2. Păun, A.; Păun; G. The power of communication: P systems with symport/antiport. *New Gener. Comput.* **2002**, *20*, 295–305. [CrossRef]
3. Martín Vide, C.; Pazos, J.; Păun, G.; Rodríguez Patón, A. A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. In Proceedings of the 8th Annual International Conference, COCOON 2002, Singapore, 15–17 August 2002; Volume 2387, pp. 290–299.
4. Martín Vide, C.; Pazos, J.; Păun, G.; Rodríguez Patón, A. Tissue P systems. *Theor. Comput. Sci.* **2003**, *296*, 295–326. [CrossRef]
5. Alhazov, A.; Freund, R.; Oswald, M. Tissue P Systems with Antiport Rules ans Small Numbers of Symbols and Cells. In Proceedings of the 9th International Conference, DLT Palermo, Italy 4–8 July 2005; pp. 100–111.
6. Bernardini, F.; Gheorghe, M. Cell Communication in Tissue P Systems and Cell Division in Population P Systems. *Soft Comput.* **2005**, *9*, 640–649. [CrossRef]
7. Krishna, S.N.; Lakshmanan, K.; Rama, R. Tissue P Systems with Contextual and Rewriting Rules. In Proceedings of the International Workshop, WMC-CdeA 2002, Curtea de Arges, Romania, 19–23 August 2002; Volume 2597, pp. 339–351.
8. Pérez-Jiménez, M.J.; Romero-Jiménez, A.; Sancho-Caparrini, F. Decision P systems and the $\mathbf{P} \neq \mathbf{NP}$ conjecture. In Proceedings of the International Workshop, WMC-CdeA 2002, Curtea de Arges, Romania, 19–23 August 2002; Volume 2597, pp. 388–399.
9. Pérez-Jiménez, M.J.; Romero-Jiménez, A.; Sancho-Caparrini, F. Complexity classes in models of cellular computing with membranes. *Nat. Comput.* **2003**, *2*, 265–285. [CrossRef]
10. Valencia-Cabrera, L.; Orellana-Martín, D.; Martínez-del-Amor, M.Á.; Pérez-Hurtado, I.; Pérez-Jiménez, M.J. From NP-completeness to DP-completeness: A Membrane Computing perspective. *Complexity* **2020**, *2020*, 6765097. [CrossRef]
11. Papadimitriou, C.H.; Yannakis, M. The complexity of facets (and some facets of complexity). In Proceedings of the 24th ACM Symposium on the Theory of Computing, San Francisco, CA, USA, 5–7 May 1982; pp. 229–234.
12. Riscos-Núñez, A.; Valencia-Cabrera, L. From SAT to SAT-UNSAT using P systems with dissolution rules. *J. Membr. Comput.* **2022**, *4*, 97–106. [CrossRef]
13. Macías-Ramos, L.F.; Pérez-Jiménez, M.J.; Riscos-Nú nez, A.; Valencia-Cabrera, L. Membrane fission versus cell division: When membrane proliferation is not enough. *Theor. Comput. Sci.* **2015**, *608*, 57–65 [CrossRef]
14. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Tissue P Systems Can be Simulated Efficiently with Counting Oracles. In *Membrane Computing. CMC 2015. Lecture Notes in Computer Science*; Rozenberg, G., Salomaa, A., Sempere, J., Zandron, C., Eds.; Springer: Cham, Switzerland, 2015; Volume 9504.
15. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Characterising the complexity of tissue P systems with fission rules. *J. Comput. Syst. Sci.* **2017**, *90*, 115–128. [CrossRef]

16. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Simulating counting oracles with cooperation. *J. Membr. Comput.* **2020**, *2*, 303–310. [CrossRef]
17. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Solving QSAT in sublinear depth. In Proceedings of the 19th International Conference, CMC 2018, Dresden, Germany, 4–7 September 2018.
18. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Membrane Division, Oracles, and the Counting Hierarchy. *Fundam. Inform.* **2015**, *138*, 97–111. [CrossRef]
19. Gutiérrez-Escudero, R.; Pérez-Jiménez, M.J.; Rius-Font, M. Characterizing tractability by tissue-like P systems. In Proceedings of the 10th International Workshop, WMC 2009, Curtea de Arges, Romania, 24–27 August 2009; Volume 5957, pp. 289–300.
20. Pan, L.; Pérez-Jiménez, M.J.; Riscos-Núñez, A.; Rius, M. New frontiers of the efficiency in tissue P systems. In *Proceedings of the Asian Conference on Membrane Computing (ACMC 2012)*; Pan, L., Paun, G., Song, T., Eds.; Huazhong University of Science and Technology: Wuhan, China, 2012; pp. 61–73.
21. Porreca, A.E.; Murphy, N.; Pérez-Jiménez, M.J. An optimal frontier of the efficiency of tissue P systems with cell division. In Proceedings of the Tenth Brainstorming Week on Membrane Computing, Seville, Spain, 30 January–3 February 2012; ; Volume II, pp. 141–166.
22. Pérez-Jiménez, M.J.; Sosík, P. An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Separation. *Fundam. Inform.* **2015**, *138*, 45–60. [CrossRef]
23. Orellana-Martín, D.; Valencia-Cabrera, L.; Pérez-Jiménez, M.J. An optimal solution to the `SAT` problem with tissue P systems. In Proceedings of the Eighteenth Brainstorming Week on Membrane Computing, Seville, Spain, 4–7 February 2020; pp. 91–100.