



From SAT to SAT-UNSAT using P systems with dissolution rules

Agustín Riscos-Núñez^{1,2} · Luis Valencia-Cabrera^{1,2}

Received: 15 July 2021 / Accepted: 9 March 2022
© The Author(s) 2022

Abstract

DP is the class of problems that are the differences between two languages from **NP**. Most difficult problems from **DP** are called **DP**-complete problems, that can be seen as the conjunction of an **NP**-complete problem and a **co-NP**-complete problem. It is easy to see that the problem **P** vs **NP** is equivalent to the problem **P** vs **DP**, and therefore **DP**-complete problems would be better candidates to attack the conjecture, since they seem to be harder than **NP**-complete problems. In this paper, a methodology to transform an efficient solution of an **NP**-complete problem into an efficient solution of a **DP**-complete problem is applied. More precisely, a solution to SAT is given by means of a uniform family of recognizer polarizationless P systems with active membranes with dissolution rules and division rules for both elementary and non-elementary membranes, and later it is transformed into a solution to the problem SAT-UNSAT.

Keywords Complexity class **DP** · Membrane computing · Product problem · Satisfiability problem

Mathematics Subject Classifications 68Q07 · 68Q15

1 Introduction

Membrane Computing is a computing paradigm introduced by Gh. Păun in 1998 [12]. It gives rise to computing models called *P systems* or *membrane systems*, that are inspired by the behavior and structure of living cells. Some different types of P systems exist, the three more popular are: cell-like membrane systems, having a hierarchical structure of membranes; tissue-like membrane systems, where cells are connected with other cells and with the environment and can interchange objects between them; and neural-like membrane systems, where neurons are connected as in the brain.

One of the most prominent research lines in this framework is computational complexity theory, where different classes of P systems are analyzed to know which problems can be solved efficiently in them [15]. For this purpose,

recognizer membrane systems were introduced in [16], being systems capable of solving decision problems giving a certain answer in the last step of the computation. An efficient solution to a decision problem is usually given by means of an infinite family of P systems in contrast to the solution of a problem by means of a unique Turing machine. The difference between them is that, while in Turing machine, we have access to an infinite tape, with P systems, we are allowed to access only to a finite number of resources. Therefore, we can define a family of P systems $\Pi = \{\Pi(u) \mid u \in I_X\}$, being I_X the set of instances of the decision problem X .

Inspired by the concept of *uniformity* in circuits, a uniform family of P systems solving a decision problem X is defined as a family $\Pi = \{\Pi(n) \mid n \in \mathbb{N}\}$. Let (cod, s) be an ordered pair of polynomial-time computable functions. We say that each P system $\Pi(n)$ solves all the instances u of X such that $n = s(u)$. Function s thus corresponds to the intuitive concept of *size* of the instance, while $cod(u)$ represents an encoding of the instance into a multiset of objects, that will be present in the input compartment of the P system at the initial configuration. The instance $u \in I_X$ will be processed by the P system $\Pi(s(u))$ with input multiset $cod(u)$, denoted by $\Pi(s(u)) + cod(u)$. The family Π is polynomially bounded, sound and complete with regard to (X, cod, s) , being cod the polynomial-time computable encoding of the

✉ Agustín Riscos-Núñez
ariscosn@us.es

Luis Valencia-Cabrera
lvalencia@us.es

¹ Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Seville, Spain

² SCORE Lab, I3US, Universidad de Sevilla, Seville, Spain

instance (i.e. the input of the system) and s the polynomial-time computable function that “selects” the P system corresponding to the size of the input; that is, the system will return yes if and only if $\theta_X(u) = 1$, and it will halt in a time bounded by a polynomial.

We denote that a problem X can be efficiently solved by a uniform family of recognizer membrane systems from \mathcal{R} by $X \in \mathbf{PMC}_{\mathcal{R}}$. The class $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and under polynomial-time reductions. Let X be a C-complete problem. If $X \in \mathbf{PMC}_{\mathcal{R}}$, then $\mathbf{C} \cup \mathbf{co} - \mathbf{C} \subseteq \mathbf{PMC}_{\mathcal{R}}$. For more details, see [15].

A computing model is said to be *efficient* (respectively, *presumably efficient*) with respect to a computational resource if it has the ability to provide solutions that use a reasonable (i.e. polynomial) amount of the resource. In this work, the computational resource is time, and therefore we will say that a computing model is said to be *efficient* if it has the ability to provide polynomial-time solutions for intractable problems (resp., NP-complete problems) [7]. The term *presumably efficient* refers to the fact that, as generally believed, if $\mathbf{P} \neq \mathbf{NP}$ then each NP-complete problem is intractable and, consequently, any presumably efficient model would be efficient. The class of problems efficiently solvable by a computing model \mathcal{R} is denoted by $\mathbf{PMC}_{\mathcal{R}}$. This complexity class is closed under complementary and under polynomial-time reductions [15]. The computing model \mathcal{AM} of recognizer P systems with active membranes and electrical charges is presumably efficient [13], as well as the computing model $\mathcal{AM}^0(+d, +ne)$ of recognizer polarizationless P systems with active membranes which makes use of dissolution rules and division for elementary and non-elementary membranes [4]. However, the computing model $\mathcal{AM}^0(-d)$ of recognizer polarizationless P systems with active membranes not making use of dissolution rules is not efficient, even when allowing division rules for elementary and non-elementary membranes [4]. Thus, if \mathcal{R} is a presumably efficient computing model of recognizer membrane systems, then $\mathbf{NP} \cup \mathbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{R}}$.

Several variants of P systems have been demonstrated to be presumably efficient. A solution to QSAT, a well-known PSPACE-complete problem, has been given in the framework of P systems with active membranes and division rules for elementary and non-elementary membranes but only evolution rules of the RHS limited to length 1 [9]. When creation rules are used instead of division rules as a method of generating an exponential workspace in polynomial time, it is possible to get rid of polarizations to obtain a solution to the QSAT problem [8]. A new research line taking tissue P systems with symport/antiport rules where objects can evolve during the application of the rules has been developed recently. Several results have been obtained, and new solutions to NP-complete and PSPACE-complete problems have been given in this framework [17].

In [18], a methodology to translate solutions of problems X_1 and X_2 to the *product problem* $X_1 \otimes X_2$ is presented. Given an efficient solution to an NP—complete problem by means of a uniform family of recognizer membrane systems, it is easy to obtain a solution to its corresponding **co-NP**—complete problem, only by changing the role of objects $y \in s$ and $n \circ$ through the computation. Therefore, this methodology can be particularly used to transform a solution of an NP-complete problem X into a solution of the DP-problem $X \otimes \bar{X}$. Obtaining this kind of solutions, we can attack the P vs DP problem giving efficiency frontiers between models of membrane systems. Moreover, it is easy to see that this problem is equivalent to the P vs NP problem, since any solution to one of the problems would lead to the same solution in the other one.

In this paper, we apply the methodology explained in that paper to the problem SAT with a uniform family of recognizer polarizationless P systems with active membranes with dissolution rules and division rules for both elementary and non-elementary membranes; that is, the main result of this paper is that $\mathbf{DP} \cup \mathbf{co} - \mathbf{DP} \subseteq \mathbf{PMC}_{\mathcal{AM}^0(+d,+ne)}$. This result can be extracted (and an even stronger) result can be extracted from [5] and [6]. The main idea of this protocol is that it can be automated.

The rest of the paper is organized as follows: In Section 2, some concepts will be introduced to make the work self-contained. The complexity class DP is introduced in Section 3. In Section 5, an efficient solution to SAT in $\mathcal{AM}^0(+d, +ne)$ is given, and then the methodology will be applied in Section 6. We finish the work with some conclusions and research lines being investigated.

2 Preliminaries

Some concepts will be introduced for making the work self-contained.

An *alphabet* is a non-empty set whose elements are called *symbols*. A *string* u over an alphabet is an ordered finite sequence of symbols. The *length* of a string u , denoted by $|u|$, is the number of occurrences of symbols that it contains. The empty string (with length 0) contains no symbols, and it is denoted by λ . The set of all strings over an alphabet Γ is denoted by Γ^* . A *language* over Γ is a subset of Γ^* . If L_1, L_2 are languages over Γ , the set-theoretic *difference* of L_1 and L_2 (in this order), denoted by $L_1 \setminus L_2$, is defined as follows: $L_1 \setminus L_2 = \{u \in \Gamma^* \mid u \in L_1 \wedge u \notin L_2\}$.

A *multiset* m over an alphabet Γ is a mapping from Γ onto the set \mathbb{N} of natural numbers. A multiset m over Γ is finite (respectively, empty) if the set $\{x \in \Gamma \mid m(x) > 0\}$ is a finite (respectively, empty) set. If m, m' are multisets over Γ , then the *union* of m and m' , denoted by $m + m'$, is the multiset

over Γ defined as follows: $(m + m')(x) = m(x) + m'(x)$, for each $x \in \Gamma$.

2.1 The Cantor pairing function

The Cantor pairing function is the mapping from $\mathbb{N} \times \mathbb{N}$ onto \mathbb{N} defined as follows: the natural number $[(t_1 + t_2) \cdot (t_1 + t_2 + 1)/2] + t_1$ is associated with every ordered pair (t_1, t_2) of natural numbers. We denote it by $\langle t_1, t_2 \rangle$, that is, $\langle t_1, t_2 \rangle = [(t_1 + t_2) \cdot (t_1 + t_2 + 1)/2] + t_1$. The Cantor pairing function is a primitive recursive function and bijective from $\mathbb{N} \times \mathbb{N}$ onto \mathbb{N} . Consequently, for each natural number t , there exists a unique pair of natural numbers t_1, t_2 such that $t = \langle t_1, t_2 \rangle$.

2.2 Decision problems and languages

A decision problem X is an ordered pair (I_X, θ_X) , where I_X is a language over a finite alphabet Σ_X and θ_X is a total boolean function over I_X . The elements of I_X are called *instances* of the problem X . If $\theta_X(u) = 1$ (respectively, $\theta_X(u) = 0$) then we say that the answer of the problem for instance u is “yes” (respectively, “no”). Each decision problem X has associated a language L_X over Σ_X : the set of instances whose answer is “yes”. Conversely, every language L over an alphabet Σ has associated a decision problem whose set of instances is Σ^* , and for each $u \in \Sigma^*$ the answer of the problem is “yes” if and only if $u \in L$.

The *complement problem* \bar{X} of a decision problem $X = (I_X, \theta_X)$ is the decision problem with the same set of instances, characterized by the following property: for each instance $u \in I_X$, the answer of \bar{X} for u is “yes” if and only if the answer of X for u is “no”. Obviously, $\bar{\bar{X}} = X$.

The class **P** (respectively, **NP**) is the class of decisions problems efficiently solvable by deterministic (resp., non-deterministic) Turing machines. The class **co-NP** is the class of decision problems such that if $X \in \mathbf{co-NP}$, then the complementary problem \bar{X} is in **NP**.

In accordance with this definition, a problem X is an **NP**-complete problem if and only if its complement problem \bar{X} is a **co-NP**-complete problem.

2.3 The satisfiability problem

Let us recall that the set of propositional Boolean variables VP is a countably infinite set $\{x_1, x_2, \dots\}$ such that each of their elements can take only two possible Boolean values (*truth values*) from $\{0, 1\}$. *Boolean expressions* are captured through the recursive definition of the minimal set BE verifying: (a) any propositional Boolean variable is in BE ; (b) if φ is in BE then its negation $\neg\varphi$ is also in BE ; and (c) if φ_1 and φ_2 are in BE , then its disjunction $\varphi_1 \vee \varphi_2$ is also in BE . Also, if φ_1 and φ_2 are in BE then its conjunction $\varphi_1 \wedge \varphi_2$ is

also in BE . Every Boolean expression φ has a finite number of Boolean variables associated with it, being denoted by $Var(\varphi)$. A truth assignment σ for a Boolean expression φ is a mapping from $Var(\varphi)$ onto the set $\{0, 1\}$. According to the classical definition of truth tables for negation and disjunction, every truth assignment σ for a Boolean expression φ provides a unique Boolean value to it. If the associated truth value is 1, then it is said that σ makes the Boolean expression φ true.

A *literal* is a Boolean variable or the negation of a Boolean variable. A *clause* is a finite disjunction of literals. Moreover, a Boolean expression is in conjunctive normal form if it is a finite conjunction of clauses. We say that a Boolean expression is in a *simplified conjunctive normal form* if it is in conjunctive normal form, literals are not repeated in each clause, and a literal and its negation do not appear in the same clause. We say that a Boolean expression φ is *satisfiable* if and only if there exists a truth assignment for φ making such an expression true. Let us recall that the satisfiability problem, denoted by **SAT**, is the problem of deciding whether a given Boolean formula in a simplified conjunctive normal form is satisfiable. The complement problem $\bar{\mathbf{SAT}}$ is denoted by **UNSAT**: *determining whether or not a Boolean expression in a simplified conjunctive normal form is unsatisfiable*. That is, given a Boolean expression φ in a simplified conjunctive normal form, determines whether all truth assignments σ for φ make it false. It is well known that the **SAT** problem is an **NP** complete problem [3]. Thus, the **UNSAT** problem is a **co-NP** complete problem.

3 The complexity class DP

The complexity class **DP** was introduced by C.H. Papadimitriou and M. Yannakis in 1982 [10] as follows:

Definition 1 A language L is in the class **DP** if and only if there are two languages L_1 and L_2 such that $L_1, L_2 \in \mathbf{NP}$ and $L = L_1 \setminus L_2$.

The complexity class **DP** can be easily characterized by the following property: a language L is in **DP** if and only if there exist two languages L_1 and L_2 in **NP** such that $L = L_1 \cap \bar{L}_2$. Class **DP** is the second level in the Boolean hierarchy. Besides, classes in the Boolean hierarchy can be described by finite unions of **DP** sets (or by finite intersections of **co-DP** sets). For more details, see [2, 19].

The complexity class **DP** can also be expressed in terms of decision problems as follows: a decision problem X is in class **DP** if and only if the language L_X associated with it belongs to class **DP**; that is, if there are two decision problems $X_1 = (I_{X_1}, \theta_{X_1})$ and $X_2 = (I_{X_2}, \theta_{X_2})$ such that L_{X_1}, L_{X_2} are

languages in **NP** and $L_X = L_{X_1} \cap \overline{L_{X_2}}$. It is easy to prove that $\mathbf{NP} \cup \mathbf{co-NP} \subseteq \mathbf{DP} \cap \mathbf{co-DP}$.

Next, we define the *product of two decision problems* and some properties of the new concept are analyzed.

Definition 2 Let $X_1 = (I_{X_1}, \theta_{X_1})$ and $X_2 = (I_{X_2}, \theta_{X_2})$ be two decision problems. The *product problem* of X_1 and X_2 , denoted by $X_1 \otimes X_2$, is the decision problem $(I_{X_1 \otimes X_2}, \theta_{X_1 \otimes X_2})$ defined as follows: $I_{X_1 \otimes X_2} = I_{X_1} \times I_{X_2}$, and for each $(u_1, u_2) \in I_{X_1 \otimes X_2}$ we have $\theta_{X_1 \otimes X_2}(u_1, u_2) = 1$ if and only if $\theta_{X_1}(u_1) = 1$ and $\theta_{X_2}(u_2) = 1$.

A product problem Y_X can be naturally associated to a problem X . This problem Y_X is the *product problem*, defined as follows: $Y_X = (I_{Y_X}, \theta_{Y_X})$ such that $I_{Y_X} = \{(u_1, u_2) \mid u_1 \in I_X, u_2 \in \overline{I_X}\}$, being \overline{X} the complementary problem of X ; and $\theta_{Y_X}((u_1, u_2)) = 1$ if and only if $\theta_X(u_1) = \theta_{\overline{X}}(u_2) = 1$.

According to these definitions, the following results concerning to the product of two decision problems are obtained.

Proposition 1 Let X_1, X_2 be decision problems.

- If $X_1 \in \mathbf{NP}$ and $X_2 \in \mathbf{co-NP}$ then $X_1 \otimes X_2 \in \mathbf{DP}$.
- If X_1 is an **NP** complete problem and X_2 is a **co-NP** complete problem then $X_1 \otimes X_2$ is a **DP** complete problem.
- If X_1 is an **NP** complete problem then the product problem $X_1 \otimes \overline{X_1}$ is a **DP** complete problem.

Since the SAT problem is an **NP** complete problem we deduce that the product problem $\text{SAT} \otimes \overline{\text{SAT}}$ is a **DP** complete problem. This problem is referred as the SAT-UNSAT problem [11] and can be expressed as follows: given two Boolean expressions φ_1, φ_2 in simplified conjunctive normal forms, determine whether or not φ_1 is satisfiable and φ_2 is unsatisfiable.

4 Recognizer polarizationless P systems with active membranes

Polarizationless P systems with active membranes were introduced as a weaker version of P systems with active membranes, where no polarizations (or only one polarization) are taken into account. A recognizer P system with active membranes of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \Sigma, \mu, H, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$, where:

1. Γ is a finite (working) alphabet.
2. $\Sigma \not\subseteq \Gamma$ is the input alphabet.

3. μ is a rooted tree whose nodes are bijectively labelled by elements of the set $H = \{1, \dots, q\}$.
4. $\mathcal{M}_i, 1 \leq i \leq q$ are multisets over Γ .
5. \mathcal{R} is the set of rules over Γ of the following forms:
 - (a) $[a \rightarrow u]_h$, for $h \in H, a \in \Gamma, u \in \Gamma^*$ (object-evolution rules).
 - (b) $a[]_h \rightarrow [b]_h$, for $h \in H, a, b \in \Gamma$ (send-in communication rules).
 - (c) $[a]_h \rightarrow b[]_h$, for $h \in H, a, b \in \Gamma$ (send-out communication rules).
 - (d) $[a]_h \rightarrow b$, for $h \in H, a, b \in \Gamma$ (dissolution rules).
 - (e) $[a]_h \rightarrow [b]_h [c]_h$, for $h \in H, a, b, c \in \Gamma$ (elementary division rules).
 - (f) $[[]_{h_1} []_{h_2}] \rightarrow [[]_{h_1}]_h [[]_{h_2}]_h$, for $h, h_1, h_2 \in H$ (non-elementary division rules).
6. $i_{in} \in H, i_{out} = env$.

A configuration of a P system is given by the state of the system at a certain moment, that is formed by the multisets of objects in each region (both membranes and the environment), and the current membrane structure. A (finite) sequence of configurations (C_0, C_1, \dots, C_n) is called a computation, given that: (a) The first configuration C_0 is given by the objects from the initial multisets and the input and (b) for each configuration $C_{t+1}, t \in \mathbb{N}, C_{t+1}$ can be obtained from C_t by the application of rules by the semantics given in [14]. A solution to a certain decision problem X by means of a uniform family of P systems from \mathcal{R} is given by (Π, cod, s) , such that for each $n \in \mathbb{N}$, there exists $\Pi(n) \in \Pi$ that solves all the instances of X of size n . In this sense, a solution is characterized by two main keypoints: (a) All the computations halt; and (b) if $\theta_X(u) = 1$, then the P system $\Pi(s(u)) + cod(u)$ will halt and will send an object **yes** to the environment at the last step of the computation; otherwise, it will return an object **no** to the environment.

For a broader definition of recognizer membrane systems and polarizationless P systems with active membranes, we refer the reader to [14].

5 A solution to SAT in $\mathcal{AM}^0(+d)$

Let $\langle x, y \rangle$ be the Cantor pairing function. Then,

$$\Pi(\langle n, p \rangle) = (\Gamma, \Sigma, \mu, H, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$$

is a recognizer P system with active membranes, with dissolution rules and division rules for elementary and non-elementary membranes that solves the instances with n variables and p clauses, where:

1. $\Gamma = \Sigma \cup \{\text{yes, no, } p, z, z^*, \#\} \cup \{d_k \mid 0 \leq k \leq 5n + p + 1\} \cup \{a_{i,j} \mid 1 \leq i \leq n, 0 \leq j \leq 2i - 1\} \cup \{c_j \mid 1 \leq j \leq p\} \cup \{x_{i,j}^*, \bar{x}_{i,j}^* \mid 1 \leq i \leq n, 1 \leq j \leq p\} \cup \{t_i, f_i, f_i^* \mid 1 \leq i \leq n\}$;
2. $\Sigma = \{x_{i,j}, \bar{x}_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq p\}$;
3. $\mu = [[\dots []_{2n+p+3} \dots]_2]_1$;
4. $H = \{1, \dots, 2n + p + 3\}$;
5. $\mathcal{M}_1 = \emptyset, \mathcal{M}_2 = \{d_0\}, \mathcal{M}_{2n+p+3} = \{d_0\} \cup \{a_{i,0} \mid 1 \leq i \leq n\}, \mathcal{M}_k = \emptyset, 3 \leq k \leq 2n + p + 2$;
6. The set \mathcal{R} contains the following rules:

6.1. Rules to generate the 2^n possible truth assignments:

$$\begin{aligned} r_{1,1,k} &\equiv [d_k \rightarrow d_{k+1}]_{2n+p+3} \text{ for } 0 \leq k \leq 2n - 1 \\ r_2 &\equiv [d_{2n}]_{2n+p+3} \rightarrow p r_{3,k,i} \equiv [a_{i,k} \rightarrow a_{i,k+1}]_{2n+p+3} \\ &\text{for } 1 \leq i \leq n, 0 \leq k \leq 2i - 2 \\ r_{4,i} &\equiv [a_{i,2i-1}]_{2n+p+3} \rightarrow [t_i]_{2n+p+3} [f_i]_{2n+p+3} \\ &\text{for } 1 \leq i \leq n \\ r_{5,i} &\equiv []_{i+1} []_{i+1}]_i \rightarrow []_{i+1}]_i []_{i+1}]_i \\ &\text{for } 3 \leq i \leq 2n + p + 2 \end{aligned}$$

6.2 Rules to check which clauses are satisfied:

$$\left. \begin{aligned} r_{6,i} &\equiv [t_i]_{2+p+2i} \rightarrow z \\ r_{7,i} &\equiv [z \rightarrow z^*]_{2+p+2i-1} \\ r_{8,i} &\equiv [f_i \rightarrow f_i^*]_{2+p+2i} \\ r_{9,i} &\equiv [f_i^*]_{2+p+2i} \rightarrow z^* \\ r_{10,i} &\equiv [z^*]_{2+p+2i-1} \rightarrow \# \end{aligned} \right\} \text{for } 1 \leq i \leq n$$

$$\left. \begin{aligned} r_{11,i,j} &\equiv [x_{i,j} \rightarrow x_{i,j}^*]_{2+p+2i} \\ r_{12,i,j} &\equiv [\bar{x}_{i,j} \rightarrow \bar{x}_{i,j}^*]_{2+p+2i} \\ r_{13,i,j} &\equiv [x_{i,j}^* \rightarrow \lambda]_{2+p+2i} \\ r_{14,i,j} &\equiv [\bar{x}_{i,j}^* \rightarrow c_j]_{2+p+2i} \\ r_{15,i,j} &\equiv [x_{i,j}^* \rightarrow c_j]_{2+p+2i-1} \\ r_{16,i,j} &\equiv [\bar{x}_{i,j}^* \rightarrow \lambda]_{2+p+2i-1} \end{aligned} \right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p$$

6.3 Rules to check if all clauses are satisfied:

$$r_{17,j} \equiv [c_j]_{2+j} \rightarrow \# \text{ for } 1 \leq j \leq p$$

6.4 Rules to check if there is at least an object p in membrane 2: $r_{18,k} \equiv [d_k \rightarrow d_{k+1}]_2$ for $1 \leq k \leq 5n + p$

$$\begin{aligned} r_{19} &\equiv [p]_2 \rightarrow \text{yes} \\ r_{20} &\equiv [d_{5n+p+1}]_2 \rightarrow \text{no} \\ r_{21} &\equiv [\text{yes}]_1 \rightarrow \text{yes}[]_1 \\ r_{22} &\equiv [\text{no}]_1 \rightarrow \text{no}[]_1 \end{aligned}$$

7. $i_{in} = p + 2$ and $i_{out} = env$.

An instance φ with n variables and p clauses from SAT is defined as follows: Let $Var(\varphi) = \{x_1, \dots, x_n\}$. Then, $\varphi = C_1 \wedge \dots \wedge C_p$, where $C_j = l_{j,1} \vee \dots \vee l_{j,r_j}$, $l_{j,k} \in \{x_i, \neg x_i\}, x_i \in Var(\varphi)$. $cod(u)$ is the set that contains $x_{i,j}$

if the variable x_i appears in the clause C_j and \bar{x}_i if $\neg x_i$ appears in the clause C_j .

5.1 A glance to the computation

The computation is divided into four stages.

1. **Generation of truth assignments:** In this stage, 2^n membranes with label $2n + p + 2$ are generated. Each object $a_{i,i-1}$ generates two new membranes each two steps with rules from 6.1, one of them with an object t_i and the other with an object f_i , representing the value acquired of the corresponding variable i . This stage takes $2n$ computational steps.
2. **First checking stage:** In this stage, rules from 6.2 will be used to check which clauses of φ are satisfied. For this, objects $t_i, x_{i,j}$ and $\bar{x}_{i,j}$ will interact in membranes $2 + p + 2i$ and $2 + p + 2i - 1$ to return an object c_j in the case that the literal i satisfies the clause j . Each variable is checked in 3 computational steps. Therefore, this stage takes $3n$ computational steps. In the Figure 1, each possible configuration can be observed.
3. **Second checking stage:** In this stage, with rules from 6.3, objects c_j dissolve the corresponding membrane labelled by $2 + j$. If membrane 3 is dissolved, the corresponding truth assignment makes true the whole formula, and an object p appears in membrane 2.
4. **Output stage:** If an object p is present in membrane labelled by 2 in the configuration \mathcal{C}_{5n+p} , there is at least one truth assignment that makes true the formula φ . Therefore, an object yes will be sent to the environment. Otherwise, rules from 6.4 will transform the object d_{5n+p+1} in an object no, that will be sent to the environment. This stage takes 2 steps in the affirmative case and 3 steps in the negative case. This process can be observed in Figure 2.

6 Application of the methodology to an NP-complete problem

Let

$$\begin{aligned} \Pi^{(1)}(\langle n_1, p_1 \rangle) &= (\Gamma^{(1)}, \Sigma^{(1)}, \mu^{(1)}, H^{(1)}, \mathcal{M}_1^{(1)}, \\ &\dots, \mathcal{M}_{2n_1+p_1+3}^{(1)}, \mathcal{R}^{(1)}, i_{in}^{(1)}, i_{out}^{(1)}) \end{aligned}$$

be a recognizer polarizationless P system with dissolution rules that solve the instances of SAT of n_1 variables and p_1 clauses as described above. We can define a recognizer polarizationless P system with dissolution rules

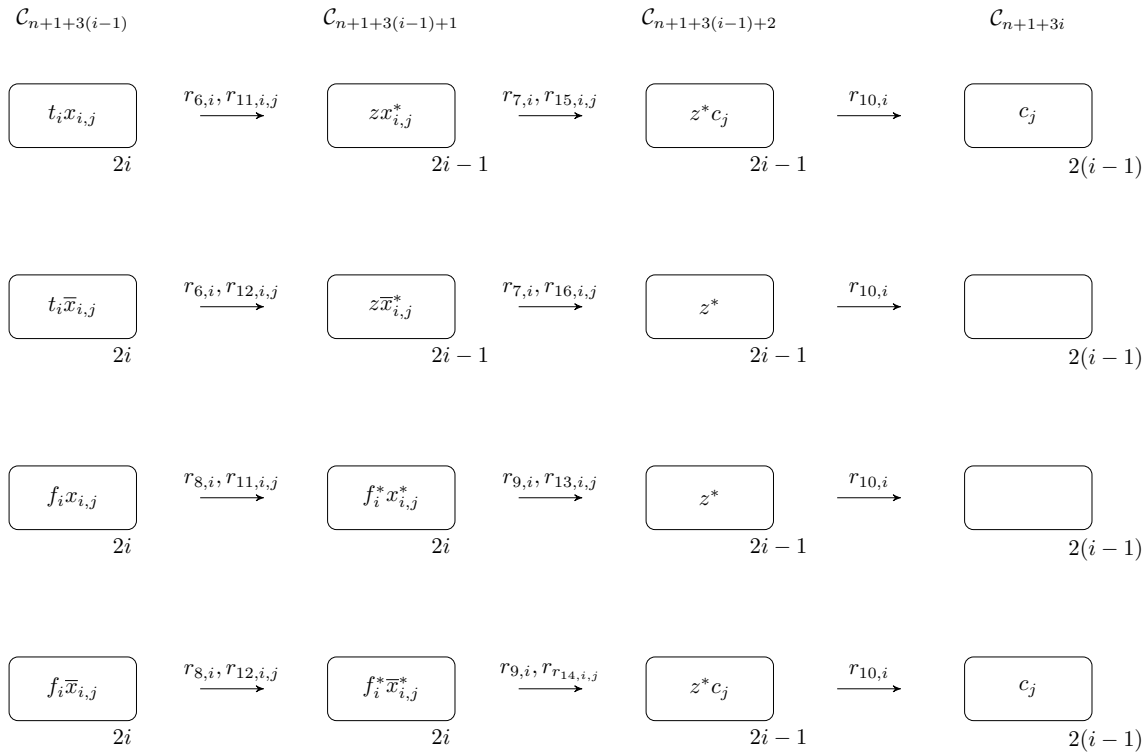


Fig. 1 Possible evolution of checking a single variable i . The membrane shown in each step corresponds with the last non-dissolved membrane of the branch. For instance, in the top row, object t_i dissolves membrane $2i$ in the first step, and therefore a membrane labelled by $2i - 1$ would be the last non-dissolved membrane of that branch. Some renaming steps are made for synchronisation, so that the checking takes always 3 computational steps, for each one of the

four cases. Each row corresponds to a different possibility of assignment: The first row, the behaviour of a positive literal with a positive assignment; the second row, it is the case of a negative literal with a positive assignment; the third row corresponds to a positive literal with a negative assignment; and in the fourth row there exists a negative literal with a negative assignment

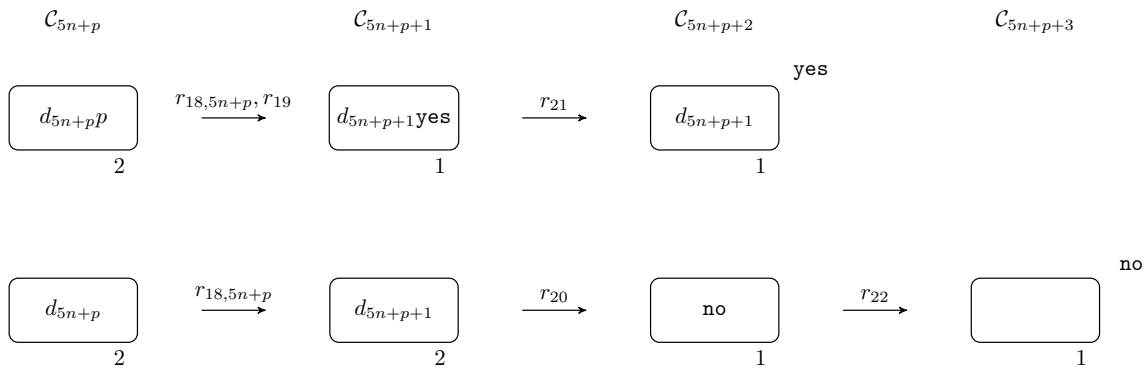


Fig. 2 Above, in the positive case, object p dissolves the membrane 2 in the first step, stopping the evolution of the object d_{5n+p+1} . Below, this object resides in the membrane 2 in the configuration C_{5n+p+1} , dissolving that membrane and transforming into an object no

$$\Pi^{(2)}((n_2, p_2)) = (\Gamma^{(2)}, \Sigma^{(2)}, \mu^{(2)}, H^{(1)}, \mathcal{M}_{2n_1+p_1+4}^{(2)}, \dots, \mathcal{M}_{2n_1+p_1+2n_2+p_2+6}^{(2)}, \mathcal{R}^{(2)}, i_{in}^{(2)}, i_{out}^{(2)})$$

that solves the instances of UNSAT of n_2 variables and p_2 clauses as follows: Since $\Gamma^{(i)}(t_i)$ should be disjoint, $\Gamma^{(2)}(t_2)$ will consist of the symbols from $\Gamma^{(1)}(t_1)$ in capital letters. All the rules are changed according to this. To make $H^{(1)}$ and $H^{(2)}$ disjoint, given $n_1, n_2, p_1, p_2 > 0$, we change the label i

of $H^{(2)}$ to $2n_1 + p + 3 + i$. In $\Pi^{(i)}$ we change objects yes and no by yes_i and no_i , respectively.

Let $t_i = \langle n_i, p_i \rangle$, and $t = \langle t_1, t_2 \rangle$. Let $q_i = t_i$ and $d_i = 2n_i + p_i + 3$. Let $p_i(|u_i|) = 5n_i + p_i + 3$. Let $m_i = q_i + d_i + 2 = t_i + 2n_i + p_i + 5$ and $n = \max\{d_i + q_i + p_i(|u_i|)\} = \max\{t_1 + 2n_1 + p_1 + 5, t_2 + 2n_2 + p_2 + 5\}$.

Now we can define a recognizer P system that solves the problem SAT \otimes UNSAT, defined as follows: An instance of the problem SAT \otimes UNSAT is a pair of formulas (φ_1, φ_2) and

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

We define a recognizer polarizationless P system with dissolution rules

$$\Pi(t) = (\Gamma, \Sigma, \mu, \mathcal{M}_1, \dots, \mathcal{M}_{2n_1+p_1+2n_2+p_2+6}, \mathcal{M}_{r_i}, \mathcal{M}_{r'_i}, \mathcal{M}_{r''_i}, \mathcal{M}_{h^{(1)}(t_1)}, \mathcal{M}_{h^{(2)}(t_2)}, \mathcal{R}, i_{in}, i_{out})$$

where:

– $\Gamma = \Gamma^{(1*)}(t) \cup \Gamma^{(2*)}(t) \cup \Gamma^{(3)}(t) \cup \Gamma^{(4)}(t) \cup \Gamma^{(5)}(t) \cup \Gamma^{(6)}(t)$, where:

$$\begin{aligned} \Gamma^{(1*)}(t) &= \Sigma^{(1)}(t_1) \cup \{\text{yes}_1, \text{no}_1, p, z, z^*, \#\} \cup \{d_k \mid 0 \leq k \leq 5n_1 + p_1 + 1\} \cup \\ &\quad \{c_j \mid 1 \leq j \leq p_1\} \cup \{a_{ij} \mid 1 \leq i \leq n_1, 0 \leq j \leq 2i - 1\} \cup \\ &\quad \{t_i, f_i, f_i^* \mid 1 \leq i \leq n_1\} \cup \{x_{ij}^*, \bar{x}_{ij}^* \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1\} \\ \Gamma^{(2*)}(t) &= \Sigma^{(2)}(t_2) \cup \{\text{yes}_2, \text{no}_2, P, Z, z^*, \#\} \cup \{D_k \mid 0 \leq k \leq 5n_2 + p_2 + 1\} \cup \\ &\quad \{C_j \mid 1 \leq j \leq p_2\} \cup \{A_{ij} \mid 1 \leq i \leq n_2, 0 \leq j \leq 2i - 1\} \cup \\ &\quad \{T_i, F_i, F_i^* \mid 1 \leq i \leq n_2\} \cup \{X_{ij}^*, \bar{X}_{ij}^* \mid 1 \leq i \leq n_2, 1 \leq j \leq p_2\} \\ \Gamma^{(3)}(t) &= \{\#, \text{YES}, \text{NO}\} \cup \{x'_{ij}, \bar{x}'_{ij} \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1\} \cup \\ &\quad \{X'_{ij}, \bar{X}'_{ij} \mid 1 \leq i \leq n_2, 1 \leq j \leq p_2\} \\ \Gamma^{(4)}(t) &= \{\text{yes}_{1k}, \text{no}_{1k}, p_k, z_k, z_k^*, \# \mid 0 \leq k \leq m_1\} \cup \\ &\quad \{d_{k'k} \mid 0 \leq k \leq 5n_1 + p_1 + 1, 0 \leq k' \leq m_1\} \cup \\ &\quad \{c_{jk} \mid 1 \leq j \leq p_1, 1 \leq k \leq m_1\} \cup \\ &\quad \{a_{ijk} \mid 1 \leq i \leq n_1, 0 \leq j \leq 2i - 1, 0 \leq k \leq m_1\} \cup \\ &\quad \{t_k, f_k, f_k^* \mid 1 \leq k \leq n_1, 0 \leq k \leq m_1\} \cup \\ &\quad \{x_{ijk}, \bar{x}_{ijk}, x_{ijk}^*, \bar{x}_{ijk}^* \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1, 0 \leq k \leq m_1\} \cup \\ &\quad \{\text{yes}_{2k}, \text{no}_{2k}, P_k, Z_k, Z_k^*, \# \mid 0 \leq k \leq m_2\} \cup \\ &\quad \{D_{k'k} \mid 0 \leq k \leq 5n_1 + p_1 + 1, 0 \leq k' \leq m_2\} \cup \\ &\quad \{C_{jk} \mid 1 \leq j \leq p_1, 1 \leq k \leq m_2\} \cup \\ &\quad \{A_{ijk} \mid 1 \leq i \leq n_1, 0 \leq j \leq 2i - 1, 0 \leq k \leq m_2\} \cup \\ &\quad \{T_{ik}, F_{ik}, F_{ik}^* \mid 1 \leq i \leq n_1, 0 \leq k \leq m_2\} \cup \\ &\quad \{X_{ijk}, \bar{X}_{ijk}, X_{ijk}^*, \bar{X}_{ijk}^* \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1, 0 \leq k \leq m_2\} \\ \Gamma^{(5)}(t) &= \{\alpha_0, \alpha_1\} \cup \{\beta_j \mid 0 \leq j \leq n\} \\ \Gamma^{(6)}(t) &= \{\gamma_j \mid 0 \leq j \leq \max\{m_1, m_2\}\} \end{aligned}$$

– $\Sigma(t) = \Sigma^{(1)}(t_1) \cup \Sigma^{(2)}(t_2)$, being $\Sigma^{(1)}(t_1) = \{x_{ij}, \bar{x}_{ij} \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1\}$ and $\Sigma^{(2)}(t_2) = \{X_{ij}, \bar{X}_{ij} \mid 1 \leq i \leq n_2, 1 \leq j \leq p_2\}$.

– $H(t) = H^{(1)} \cup H^{(2)} \cup \{r_i, r'_i, r''_i, h^{(1)}(t_1), h^{(2)}(t_2)\}$, where $r_i, r'_i, r''_i, h^{(1)}(t_1), h^{(2)}(t_2)$ are different from each other, and $r_i, r'_i, r''_i, h^{(1)}(t_1), h^{(2)}(t_2) \notin H^{(1)} \cup H^{(2)}$.

– $\mu = [[[[[\dots [[]_{h^{(1)}(t_1)}]_{h_1} \dots]_1 [\dots [[]_{h^{(2)}(t_2)}]_{h_3} \dots]_{h_2}]_{r''_i}]_{r'_i}]_{r_i}]_{t_i}$ where $h_1 = 2n_1 + p_1 + 3$, $h_2 = 2n_1 + p_1 + 4$ and $h_3 = 6 + 2n_1 + 2n_2 + p_1 + p_2$.

– $\mathcal{M}_2 = \{d_{0_0}\}, \mathcal{M}_{2n_1+p_1+3} = \{d_{0_0}\} \cup \{a_{i,0_0} \mid 1 \leq i \leq n_1\}, \mathcal{M}_{2n_1+p_1+5} = \{D_{0_0}\}, \mathcal{M}_{2n_1+p_1+2n_2+p_2+6} = \{D_{0_0}\} \cup \{A_{i,0_0} \mid 1 \leq i \leq n_2\}, \mathcal{M}_{r'_i}(t) = \{\alpha_0, \beta_0\}$ and $\mathcal{M}_{h^{(i)}(t_i)}(t) = \{\gamma_0\}$, for $i = 1, 2$ and $\mathcal{M}_j = \emptyset$ for the rest of membranes.

– The set of rules is $\mathcal{R}_{\Pi^{(1)}(t_1)} \cup \mathcal{R}_{\Pi^{(2)}(t_2)} \cup \mathcal{R}_i^*$, 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 yes and no by objects yes_i and no_i , respectively, in each rule. \mathcal{R}_i^* is the following set of rules:

1 Rules for simultaneously transporting $\text{cod}_i(u_i)$ from r'_i to $h^{(i)}(t_i)$, for $i = 1, 2$ and transforming it into $\text{cod}_i(u'_i)$:

$$\left. \begin{array}{l} a [\]_1 \rightarrow [a']_1 \\ A [\]_{2n_1+p_1+4} \rightarrow [A']_{2n_1+p_1+4} \\ a' [\]_i \rightarrow [a']_i \\ a' [\]_{h^{(1)}(t_1)} \rightarrow [a']_{h^{(1)}(t_1)} \\ A' [\]_j \rightarrow [A']_j \\ A' [\]_{h^{(2)}(t_2)} \rightarrow [A']_{h^{(2)}(t_2)} \end{array} \right\} \text{for } \left\{ \begin{array}{l} a \in \{x_{ij}, \bar{x}_{ij} \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1\}, \\ A \in \{X_{ij}, \bar{X}_{ij} \mid 1 \leq i \leq n_2, 1 \leq j \leq p_2\} \\ a \in \{x_{ij}, \bar{x}_{ij} \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1\}, \\ 3 \leq i \leq 3 + \lfloor n_1/2 \rfloor \\ A \in \{X_{ij}, \bar{X}_{ij} \mid 1 \leq i \leq n_2, 1 \leq j \leq p_2\}, \\ 2n_1 + p_1 + 5 \leq j \leq 2n_1 + p_1 + 2n_2 + p_2 + 6 \end{array} \right.$$

2 Rules for simultaneously transporting $cod_i(u_i)$ from $h^{(i)}(t_i)$ to the input membrane in $\Pi^{(i)}(t_i)$, by transforming it into $cod_i(u_i)$, for $i = 1, 2$:

$$\left. \begin{array}{l} [\gamma_j \rightarrow \gamma_{j+1}]_{h^{(i)}(t_i)} \text{ for } i \in \{1, 2\}, 0 \leq j \leq m_i - 1 \\ [\gamma_{m_i}]_{h^{(i)}(t_i)} \rightarrow \# \text{ for } i \in \{1, 2\} \\ [a \rightarrow a]_{h^{(1)}(t_1)} \\ [A' \rightarrow A]_{h^{(2)}(t_2)} \end{array} \right\} \text{for } \left\{ \begin{array}{l} a \in \{x_{ij}, \bar{x}_{ij} \mid 1 \leq i \leq n_1, 1 \leq j \leq p_1\}, \\ A \in \{X_{ij}, \bar{X}_{ij} \mid 1 \leq i \leq n_2, 1 \leq j \leq p_2\} \end{array} \right.$$

3 Rules for synchronizing the simulation of $\Pi^{(i)}(t_i) + cod_i(u_i)$ in order to start at the $(m_i + 2)$ -th transition step, for $i = 1, 2$:

$$\left. \begin{array}{l} [d_{0_k} \rightarrow d_{0_{k+1}}]_2 \\ [d_{0_k} \rightarrow d_{0_{k+1}}]_{2n_1+p_1+3} \\ [a_{i,0_k} \rightarrow a_{i,0_{k+1}}]_{2n_1+p_1+3} \end{array} \right\} \text{for } 0 \leq k \leq m_1 \\ \left. \begin{array}{l} [D_{0_k} \rightarrow D_{0_{k+1}}]_{2n_1+p_1+5} \\ [D_{0_k} \rightarrow D_{0_{k+1}}]_{2n_1+p_1+2n_2+p_2+6} \\ [A_{i,0_k} \rightarrow A_{i,0_{k+1}}]_{2n_1+p_1+2n_2+p_2+6} \end{array} \right\} \text{for } 0 \leq k \leq m_2 \\ [d_{0_{m_1}} \rightarrow d_0]_2 \\ [d_{0_{m_1}} \rightarrow d_0]_{2n_1+p_1+3} \\ [D_{0_{m_1}} \rightarrow D_0]_{2n_1+p_1+5} \\ [D_{0_{m_1}} \rightarrow D_0]_{2n_1+p_1+2n_2+p_2+6} \\ [a_{i,0_{m_1}} \rightarrow a_{i,0}]_{2n_1+p_1+3} \text{ for } 1 \leq i \leq n_1 \\ [A_{i,0_{m_1}} \rightarrow A_{i,0}]_{2n_1+p_1+2n_2+p_2+6} \text{ for } 1 \leq i \leq n_2$$

4 Rules for synchronizing the instant $n = 2 + \max\{d_i + q_i + p_i(|u_i|) : 1 \leq i \leq 2\}$ in which the answers of $\Pi^{(1)}(t_1) + cod_1(u_1)$ and $\Pi^{(2)}(t_2) + cod_2(u_2)$ reach the membrane r''_i : $[\beta_j \rightarrow \beta_{j+1}]_{r''_i}$ for $0 \leq j \leq n - 1$

5 Rules for the output of the system $\Pi(t) + cod(u_1, u_2)$:

$$\left. \begin{array}{l} [\beta_n]_{r''_i} \rightarrow \# \\ [no_i]_{r''_i} \rightarrow NO \\ [\alpha_0 \rightarrow \alpha_1]_{r''_i} \\ [\alpha_1]_{r''_i} \rightarrow YES [\]_{r''_i} \\ [YES]_{r''_i} \rightarrow yes [\]_{r''_i} \end{array} \right\} \text{for } i \in \{1, 2\}$$

Apart from these rules, rules from $\mathcal{R}_{\Pi^{(1)}(t_1)}$ and $\mathcal{R}_{\Pi^{(2)}(t_2)}$ are present in the set of rules \mathcal{R}

– $i_{in} = r''_i$ and $i_{out} = env$.

Theorem 1 $SAT \otimes UNSAT \in \text{PMC}_{\mathcal{AM}^0(+d,+ne)}$

Proof It can be seen from the previous family of P systems that the previous P system takes an instance of the problem

$SAT \otimes UNSAT$, (φ_1, φ_2) , and it returns an object yes if and only if φ_1 is satisfiable and φ_2 is unsatisfiable. This system will return an object yes or an object no, but not both, only in the last step of the computation. Taking into account the definition of n from above, the computation will take $n + 4$ computational steps in the affirmative case and $n + 3$ steps in the negative case.

Corollary 1 $\text{DP} \cup \text{co-DP} \subseteq \text{PMC}_{\mathcal{AM}^0(+d,+ne)}$

7 Conclusion

In previous works, several solutions to NP-complete problems have been given, but while some of them this solution is non-uniform, in other there is a solution to other problem different than SAT. Since this work was thought as a SAT-UNSAT solver, an efficient solution to SAT has been given in this paper by means of a uniform family of recognizer membrane systems from $\mathcal{AM}^0(+d,+ne)$.

In [18], a methodology for transforming an efficient solution for an NP-complete problem into an efficient solution for a DP-complete problem was given. In this paper, we have applied this methodology to make easier to understand how can it be used to obtain efficient solutions for harder problems. The final result is that, given the fact that $SAT \in \text{PMC}_{\mathcal{AM}^0(+d,+ne)}$, and we can apply the methodology to that solution (since it has dissolution rules, one of the main ingredients of this methodology), then $\text{DP} \cup \text{co-DP} \subseteq \text{PMC}_{\mathcal{AM}^0(+d,+ne)}$. From [1], we know that $\text{PSPACE} \subseteq \text{PMC}_{\mathcal{AM}^0(+d,+ne)}$, so it seems to be a non-surprising result, since $\text{DP} \subseteq \text{PSPACE}$. But the main contribution of this paper is to provide an example of the application of the methodology presented in [18], in such a way that every solution to a NP-complete problem can be transformed into a solution to its corresponding DP-complete problem. It is really interesting as the P vs NP problem is equivalent to the P vs DP problem. As DP-complete problems are supposed to be harder than NP-complete problems, they are better candidates to find a solution to the conjecture. Therefore, it is interesting to find borderlines between non-efficient computing models and models capable of efficiently solving DP-complete problems.

The most interesting research lines that arise from here is to create methodologies for different variants of P systems, such as tissue P systems with symport/antiport rules and other types of cell-like membrane systems. In general, we think that this process can be translated to any type of membrane system that has some minimum requirements. Another interesting research line is to apply the methodology to different solutions of decision problems, since a

natural extension to the parallel resolution of n problems can arise from this protocol.

Acknowledgements This work was supported by “FEDER/Ministerio de Ciencia e Innovación - Agencia Estatal de Investigación/ _Proyecto (TIN2017-89842-P)” - MABICAP.

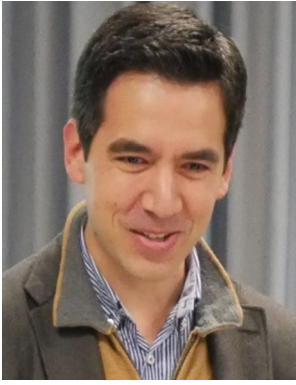
Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alhazov, A., & Pérez-Jiménez, M. J. (2007). Uniform solution of QSAT using polarizationless active membranes. In J. Durand-Lose & M. Margenstern (Eds.), *Machines, Computations, and Universality* (pp. 122–133). Berlin Heidelberg, Berlin, Heidelberg: Springer.
- Cai, J. Y., Gundermann, T., Hartmanis, J., Hemachandra, L. A., Sewelson, V., Wagner, K., & Wechsung, G. (1988). The boolean hierarchy i: Structural properties. *SIAM Journal on Computing*, *17*(6), 1232–1252. <https://doi.org/10.1137/0217078>.
- Garey, M.R., Johnson, D.S.: *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA (1990)
- Gutiérrez-Naranjo, M. A., Pérez-Jiménez, M. J., Riscos-Núñez, A., & Romero-Campero, F. J. (2006). On the power of dissolution in p systems with active membranes. In R. Freund, G. Păun, G. Rozenberg, & A. Salomaa (Eds.), *Membrane Computing* (pp. 224–240). Berlin Heidelberg, Berlin, Heidelberg: Springer.
- Leporati, A., Manzoni, L., Mauri, G., Porreca, A., & Zandron, C. (2020). Subroutines in P systems and closure properties of their complexity classes. *Theoretical Computer Science*, *805*, 193–205.
- Leporati, A., Manzoni, L., Mauri, G., Porreca, A., & Zandron, C. (2020). A Turing machine simulation by P systems without charges. *Journal of Membrane Computing*, *2*, 71–79.
- Macías-Ramos, L.F., Pérez-Jiménez, M.J., Riscos-Núñez, A., Valencia-Cabrera, L.: Membrane fission versus cell division: When membrane proliferation is not enough. *Theoretical Computer Science* **608**, 57 – 65 (2015). doi: 10.1016/j.tcs.2015.06.025. <http://www.sciencedirect.com/science/article/pii/S0304397515005356>
- Orellana-Martín, D., Valencia-Cabrera, L., Riscos-Núñez, A., & Pérez-Jiménez, M. (2019). Membrane creation in polarizationless P systems with active membranes. *Fundamenta Informaticae*, *171*(1–4), 297–311.
- Pan, L., Orellana-Martín, D., Song, B., Pérez-Jiménez, M.J.: Cell-like P systems with polarizations and minimal rules. *Theoretical Computer Science* **816**, 1 – 18 (2020). doi: 10.1016/j.tcs.2019.10.001. <http://www.sciencedirect.com/science/article/pii/S0304397519306231>
- Papadimitriou, C., Yannakakis, M.: The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences* **28**(2), 244 – 259 (1984). doi: 10.1016/0022-0000(84)90068-0. <http://www.sciencedirect.com/science/article/pii/0022000084900680>
- Papadimitriou, C.H.: *Computational complexity*. Addison-Wesley (1994)
- Păun, G.: Computing with membranes. *Journal of Computer and System Sciences* **61**(1), 108 – 143 (2000). doi: 10.1006/jcss.1999.1693. <http://www.sciencedirect.com/science/article/pii/S002200009916938>
- Păun, G. (2001). Computing with membranes: Attacking NP-complete problems. In I. Antoniou, C. S. Calude, & M. J. Dinneen (Eds.), *Unconventional Models of Computation, UMC2K* (pp. 94–115). London, London: Springer.
- Păun, G., Rozenberg, G., & Salomaa, A. (2010). *The Oxford Handbook of Membrane Computing*. USA: Oxford University Press Inc.
- Pérez-Jiménez, M. J., Romero-Jiménez, A., & Sancho-Caparrini, F. (2003). Complexity classes in models of cellular computing with membranes. *Natural Computing*, *2*(3), 265–285. <https://doi.org/10.1023/A:1025449224520>.
- Pérez Jiménez, M. J., Romero Jiménez, Á., & Sancho Caparrini, F. (2003). Decision P systems and the $P \neq NP$ conjecture. In G. Păun, G. Rozenberg, A. Salomaa, & C. Zandron (Eds.), *Membrane Computing* (pp. 388–399). Berlin Heidelberg, Berlin, Heidelberg: Springer.
- Song, B., Li, K., Orellana-Martín, D., Valencia-Cabrera, L., Pérez-Jiménez, M.J.: Cell-like P systems with evolutionary symport/antiport rules and membrane creation. *Information and Computation* p. 104542 (2020). doi: 10.1016/j.ic.2020.104542. <http://www.sciencedirect.com/science/article/pii/S0890540120300304>
- Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J.: From NP-completeness to DP-completeness: A membrane computing perspective. *Complexity* **2020**(6765097), 10 pages (2020)
- Wechsung, G. (1985). On the Boolean closure of NP. In L. Budach (Ed.), *Fundamentals of Computation Theory* (pp. 485–493). Berlin Heidelberg, Berlin, Heidelberg: Springer.

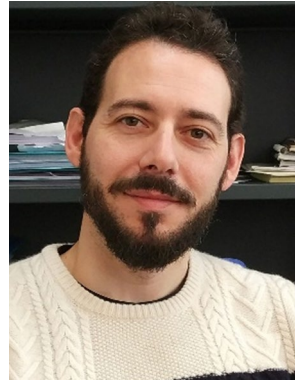
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Agustín Riscos-Núñez is an Associate Professor at Dpt. Computer Science and Artificial Intelligence, guarantor researcher at the Smart Computer systems Research and Engineering Lab (SCORE), head of the Research Group on Natural Computing, founding member and Secretary of the Research Institute of Computer Engineering (I3US) at Universidad de Sevilla, Spain. Founding member of the “International Membrane Computing Society (IMCS)”, IEEE Member.

His main areas of expertise are bio-inspired computing and artificial intelligence. His research interests mainly focus on computational complexity theory and computational modelling of complex systems (specially population dynamics in ecology), as well as other practical applications in the fields of bioinformatics, biomedicine, high performance computing and robotics. He has co-authored around 50 scientific papers published at indexed international journals, and more than 35 conference contributions (some of them as invited speaker). He has also co-authored several book chapters

and one monograph, and has served as editor for more than 15 collective volumes.



Luis Valencia-Cabrera received his BSc in 2005, and was working for an IT company from 2005 to 2010, getting professional certifications SCJP, SCWCD, SCBCD (Oracle, 2008) and CFPS (IFPUG, 2006), plus a professional MSc in 2007. In 2013 he got an academic MSc (2013), and in 2015 he defended his Ph.D. in Logics, computing and Artificial Intelligence, receiving a Ph.D. Extraordinary award. He is currently a researcher at the Smart Computer systems Research and

Engineering Lab (SCORE) and an Associate Professor at Universidad de Sevilla (Spain), where he has supervised many MSc/BSc, 2 doctoral theses, and has more than a hundred scientific contributions, including 43 papers in ISI-JCR indexed journals.