Mobile Membranes

BOGDAN AMAN AND GABRIEL CIOBANU
Faculty of Computer Science, Alexandru Ioan Cuza University, 700506 Iaşi, Romania
Corresponding author: Gabriel Ciobanu (gabriel@info.uaic.ro)

This work was supported by the project funded by the Ministry of Research and Innovation within Program 1 - Development of the National Research and Development System, Subprogram 1.2 - Institutional Performance - RDI excellence funding projects, under Contract 34PFE/19.10.2018.

ABSTRACT Mobile membranes represent a model of computation inspired from the biological movement provided by endocytosis and exocytosis in the living cells. This paper presents a survey of the results treating the computational power of the mobile membranes, their efficiency in solving NP-complete problems, and connections with other formal approaches able to handle mobility.

INDEX TERMS Membrane computing, mobility, process calculi.

I. INTRODUCTION
Taking inspiration from the communication and mobility in cells, new computer architectures performing parallel computations are defined and studied. Bio-inspired computing has the possibility to use features as massive parallelism, reversible computations and non-determinism. The architecture and the behaviour of living cells motivate the membrane systems (also called P systems) which are defined in [57] and [59] by using formal languages theory, computability theory and complexity theory. Membrane systems can appear as a tree (cell-like [59]) or as a graph (tissue-like [53] and neural-like [41]). The membrane systems with a cell-like structure have the following characteristics: (i) an hierarchical structure of membranes (nested or disjoint) with a skin provided by the unique outermost membrane; an elementary membrane does not contain other membranes, while a composite one does; (ii) each membrane may contain multisets of objects and some rules that manipulate objects and membranes. The main research directions considered for membrane systems deal with the computational power with respect to the Turing machine, and with the efficient algorithms to solve NP-complete problems in polynomial time (and exponential space). Another research direction is the connections with other approaches (e.g., process calculi) in which the concept of mobility is central.

The fluid-mosaic model [65] is the currently accepted model for the structure of a cell membrane. The increased fluidity of the outer membrane allowed the development of two mechanisms, called endocytosis and exocytosis. Endocytosis and exocytosis are complementary operation that allow substances to enter (endocytosis) or exit (exocytosis) the cell through membrane-bounded vesicles. P systems with mobile membranes [51] are the first class of membrane systems to consider the operations of exocytosis and endocytosis.

After presenting some notions used throughout the paper (Section II) and existing variants of mobile membranes (Section III), the paper proceeds by presenting a survey of the results treating their efficiency in solving NP-complete problems (Section IV), computational power of the mobile membranes (Section V), and connections with other formal approaches able to handle mobility (Section VI).

II. PRELIMINARIES
A. FORMAL LANGUAGES
Some notions from the theory of formal languages [35], [64] are presented in what follows. Consider $O = \{a_1, \ldots, a_n\}$ an alphabet, and $\lambda$ an empty string. The set of all strings over the alphabet $O$ is denoted by $O^*$. The set $O^*$ with concatenation and unit $\lambda$ is a monoid. Given $x \in O^*$, the number of occurrences of symbol $a$ in $x$ is denoted by $|x|_a$. A string over $O$ (modulo permutation) identifies a multiset. The Parikh vector over $O$ is defined as $\psi_O : O^* \rightarrow N^n$ where $\psi_O(x) = (|x|_{a_1}, \ldots, |x|_{a_n})$ for all $x \in O^*$. In a similar manner, $\psi_V(L) = \{\psi_O(x) \mid x \in L\}$ and $PsFL = \{\psi_O(L) \mid L \in FL\}$ denote the Parikh vector over a language $L$ and a family $FL$ of languages, respectively. $RE$ and $PsRE$ denote the family of recursively enumerable languages and its Parikh image (Turing computable sets of vectors of natural numbers), respectively.

An $EOL$ system is a tuple $G = (O, T, \omega, R)$ such that $O$ and $T \subseteq O$ are the alphabet and terminal alphabet, respectively,
$\omega \in O^*$ represents the initial axiom, and $R = \{ a \rightarrow \nu \mid a \in O, \nu \in O^* \}$ is a finite set of rules. The set of rules is constructed such that for all $a \in O$ it exists $a \rightarrow \nu \in R$. Given $w_1, w_2 \in O^*$ with $w_1 = a_1 \ldots a_n$ and $w_2 = v_1 \ldots v_n$, then $w_2$ can be obtained from $w_1$ (denoted by $w_1 \Rightarrow w_2$) if exists $a_i \rightarrow v_i \in R$, $1 \leq i \leq n$. The set $L(G) = \{ x \in T^* \mid \omega \Rightarrow^* x \}$ represents the generated language.

An ETOL system is a tuple $G = (O, T, \omega, R_1, \ldots, R_n)$ if for all $i$ such that $1 \leq i \leq n$ it holds that $O, T, \omega$ and $R_i$ form an ETOL system. Each set $R_i$ with $1 \leq i \leq n$ is called a table. The set $L(G) = \{ x \in T^* \mid \omega \Rightarrow_{R_i} \ldots \Rightarrow_{R_m} w_m = x \}$ with $m \geq 0$, $1 \leq j_i \leq n$ and $1 \leq i \leq m$ represents the generated language. PsETOL denotes the Parikh image of the ETOL systems.

A context-free matrix grammar without appearance checking is denoted by $G = (N, T, S, M)$, where $N$ and $T$ with $N \cap T = \emptyset$ are non-terminals and terminals alphabets (respectively), $S \in N$ is the initial axiom, $M = \{ (A_1 \rightarrow x_1, \ldots, A_n \rightarrow x_n) \mid A_i \in N, x_i \in (N \cup T)^* \}$ is a finite set of matrices formed from context-free rules. Given a string $w \in (N \cup T)^*$, a matrix $m = (r_1, \ldots, r_n)$ is executed by applying the rules $r_1, \ldots, r_n$ in the given order. An execution denoted by $w \rightarrow_m z$ happens if it exists $m = (A_1 \rightarrow x_1, \ldots, A_n \rightarrow x_n) \in M, w_1, \ldots, w_{n+1} \in (N \cup T)^*$ such that $w = w_1, w_{n+1} = z$, and for all $i$ such that $1 \leq i \leq n$ it holds that $w_i = w_1 A_i w_i''$ and $w_{i+1} = w_i' x_i w_i''$. $L(G) = \{ x \in T^* \mid S \Rightarrow^* x \}$ is the language generated by $G$, and MAT is the family of languages generated by the context-free matrix grammars.

Other useful notations: $N$ denotes the set of natural numbers, $NRE$ the family of recursively enumerable sets, $NCS$ the family of context-sensitive sets, $NO^k$ the family of ordered sets, $NMAT$ the family of sets of matrix grammars without appearance checking, and $NFIN$ the family of finite sets of natural numbers. According to [35], we have:

$$NMAT^A, \quad NMAT_{ac}^A, \quad NCS NO^k \subset NRE.$$ 

### B. BRANE CALCULI

The biological operations of exocytosis, endocytosis and mitosis inspired the creation of two brane calculi called PEP (standing for Pinol/Exo/Phago) and MBDA (standing for Mate/Bud/Drip). As shown in [27], PEP can simulate MBDA.

An empty membrane with patch $\rho$ can be created by using the action $pinol(\rho)$. The complementary exo actions $n^\\backslash$ and $\overline{\pi}^\\backslash$ are used to merge two nested membranes. The complementary phago actions $n^\backslash$ and $\overline{\pi}^\backslash(\rho)$ are used to model a membrane “eating” another membrane.

A PEP system consists of composed nested membranes made from patches $\sigma$. The composition of two system is denoted by $P \circ Q$, while a system composed from a patch $\sigma$ and a system $P$ is denoted by $\sigma(P)$. The term $a, \sigma$ denotes a brane able to execute $a$, and then behave as $\sigma$. We use shorthand notions: $a, (P)$ and $\sigma()$, for $a, 0, 0(P)$ and $\sigma(\emptyset)$, respectively. The set of systems presented in Table 1 is denoted by $P$. The evolution of PEP systems takes place by using the rules given in Table 2.

### TABLE 1. Syntax of PEP calculus [27].

| Actions | $a, b ::= n^\\backslash \mid \overline{\pi}^\\backslash(\sigma)$ |
|---------|-------------------------------------------------|
| $\sigma, \tau ::= 0 \mid \sigma\tau \mid a, \sigma$ |
| Systems | $P, Q ::= \emptyset \mid P \circ Q \mid \sigma(P)$ |

| Phago combinations of actions |
|-----------------------------|
| $\text{Exo}_{\text{ex}0^k}$, $\text{Pino}_{\text{exo}^k}$, $\text{Phago}_{\text{exo}^k}$ |

| Branes |
|--------|
| $\sigma_{\text{F}}, \tau_{\text{F}} ::= 0 \mid \sigma_{\text{F}} \tau_{\text{F}}$ |

### TABLE 2. Reduction rules for PEP calculus [27].

| Pino | $\sigma(\sigma_0(P)) \rightarrow_b \sigma_0(\sigma(\sigma_0(P)) \circ P)$ |
| Exo | $\overline{\pi}^\\backslash(\sigma_0(P) \circ Q) \rightarrow_b P \circ \sigma(\sigma_0(P) \circ Q)$ |
| Phago | $\sigma^\\backslash(\sigma_0(P) \circ Q) \rightarrow_b \sigma_0(\sigma(\sigma_0(P) \circ Q))$ |

| System | $P \rightarrow_b Q$ implies $P \circ R \rightarrow_b Q \circ R$ |
|--------|-------------------------------------------------|
| Mem | $P \rightarrow_b Q$ implies $a, P \rightarrow_b \sigma(P) \rightarrow_b Q$ |
| Struct | $P \equiv_b P' \mid P'' \rightarrow_b Q' \mid Q'' \equiv_b Q$ implies $P \rightarrow_b Q$ |

Used to rearrange some parts of a PEP system, the structural congruence $\equiv_b$ is presented in Table 3.

### TABLE 3. Structural congruence of PEP calculus [27].

| $\sigma \equiv_b \sigma$ |
|-------------------------|
| $\sigma_0 \equiv_b \sigma$ |
| $\sigma \equiv_b \sigma_0$ |
| $\sigma \equiv_b \sigma_0 \rightarrow_b \rho \equiv_b \sigma \rightarrow_b \rho \equiv_b \sigma_0 \rightarrow_b \rho$ |
| $\sigma \equiv_b \sigma_0 \rightarrow_b \rho \equiv \sigma_0 \rightarrow_b \rho \equiv \sigma \equiv_b \sigma_0 \rightarrow_b \rho \equiv \sigma_0 \rightarrow_b \rho \equiv \sigma$ |
| $\sigma \equiv_b \sigma_0 \rightarrow_b \rho \equiv \sigma_0 \rightarrow_b \rho \equiv \sigma \equiv_b \sigma_0 \rightarrow_b \rho \equiv \sigma_0 \rightarrow_b \rho \equiv \sigma$ |
| $0(\sigma) \equiv_b 0, \sigma \equiv_b 0, P \equiv_b Q, P \circ Q \equiv_b P \circ Q$ |
| $P \equiv_b Q \circ R \equiv_b (P \circ Q) \circ R$ |
| $P \equiv_b Q \circ R \equiv_b (P \circ Q) \circ R$ |

### C. SAFE AMBIENTS

A variant of mobile ambients [28] is represented by safe ambients [52] in which the mobility is provided by pairs of capabilities. Given an infinite set of names $\mathcal{N} = \{ m, n, \ldots \}$, the capabilities of the set $C = \{ C_1, C_2, \ldots \}$ and the safe ambients of the set $A = \{ A_1, A_2, B_1, B_2, \ldots \}$ are defined by:

$$C ::= in \ n \mid \overline{in} \ n \mid out \ n \mid \overline{out} \ n$$
| $\in_0 n \mid C.A \mid n[ A ]$ |

### Process 0 does nothing. The process C.A provided a movement by consuming the capability C, and then continues by the execution of A. An ambient $n[ A ]$ denotes a process $A$ that can be executed inside the bounded place with name $n$. $A \mid B$ is a parallel composition. Processes can be rearranged by using the structural congruence $\equiv_{amb}$ constructed such that $(A, \emptyset, 0)$ is a commutative monoid.

The evolution of the safe ambients takes place by using the axioms and rules presented in what follows:

| Axioms |
|---------|
| $\text{(In)} \quad m[ in \ n.A] \mid A_1 ] \mid n[ \overline{in}.n.B] \mid B_1 ]$ |
| $\equiv_{amb} m[n[ A ] \mid A \mid B_1 ] \mid B_1 ]$ |
| $\text{(Out)} \quad m[ out \ n.A] \mid A_1 ] \mid \overline{out}.n.B] \mid B_1 ]$ |
| $\equiv_{amb} m[A \mid A_1 ] \mid n[ B ] \mid B_1 ]$ |
| $\text{(Open)} \quad m.A \mid n[ n.A] \mid B_1 ]$ |
| $\equiv_{amb} A \mid B \mid B_1$ |
Rules:

(Comp) \[ A \Rightarrow_{\text{amb}} A_1 \]

(Amb) \[ A \mid B \Rightarrow_{\text{amb}} A_1 \mid B \]

(Struc) \[ A \equiv A_1, A \Rightarrow_{\text{amb}} A_1 \Rightarrow_{\text{amb}} B_1, B_1 \equiv B \]

The transitive and reflexive closure of the relation \( \Rightarrow_{\text{amb}} \) is denoted by \( \Rightarrow^*_{\text{amb}} \).

D. COLOURED PETRI NETS

Coloured Petri nets [42] are useful in studying various properties of distributed systems. Given a variable \( e \) from a set \( \text{EXPR} \) of expressions and a variable \( x \) from the set \( X \), \( \text{Var}[e] \) denotes the set of free variables, while \( \text{Type}[e] \) and \( \text{Type}[x] \) denote the type of an expression and of a variable, respectively. The set \( \text{EXPR}_X \) of expressions contains the expressions \( e \in \text{EXPR} \) such that \( \text{Var}[e] \subseteq X \).

**Definition 1** [42]: A non-hierarchical Coloured Petri Net is a tuple \( \text{CPN} = (P, T, A, \Sigma, X, C, G, E, I) \), where:

- \( P \) and \( T \), with \( P \cap T = \emptyset \), are finite sets of places and transitions;
- \( A \subseteq (P \times T) \cup (T \times P) \) is a finite set of directed arcs;
- \( \Sigma \) is a non-empty finite set of colour sets;
- \( X \) is a finite set of typed variables with \( \text{Type}[x] \in \Sigma \) for all \( x \in X \);
- \( C : P \rightarrow \Sigma \) is a colour set function assigning to each place a colour set;
- \( G : T \rightarrow \text{EXPR}_X \) is a guard function assigning to each transition \( t \) a guard with \( \text{Type}[G(t)] = \text{Bool} \);
- \( E : A \rightarrow \text{EXPR}_X \) is an arc expression function assigning to each arc \( a \) connected to a place \( p \) a guard with \( \text{Type}[E(a)] = C(p) \cup \Sigma \);
- \( I : P \rightarrow \text{EXPR}_R \) is an initialization function assigning to each place \( p \) an initialization expression \( \text{Type}[I(p)] = C(p) \cup \Sigma \).

A marking is given by a distribution of tokens over the places of a net. Given two markings \( m \) and \( m' \) and a set of transitions \( U \), the notation \( m[U]m' \) means that \( m \) leads to \( m' \) by applying \( U \).

III. MOBILE MEMBRANES

Inspired by the biological movements of cell membranes, the following variants of mobile membrane systems are defined: simple, enhanced and mutual mobile membranes, as well as mutual mobile membranes with objects on surface.

A. SIMPLE MOBILE MEMBRANES

Inspired also by the process calculus of mobile ambients [28], mobile P systems were introduced in [60]. Several years later, a variant of P systems with active membranes [57] called P system with mobile membranes was defined in [51]. We use this variant as simple mobile membrane.

**Definition 2** [51]: A simple mobile membrane is a tuple

\[ \Pi = (O, H, \mu, w_1, \ldots, w_n, R) \]

such that:

- \( n \geq 1 \) is the degree of the system;
- \( O \) is a finite alphabet of objects;
- \( H \) is a finite set of labels for membranes;
- \( \mu \subset H \times H \) is the membrane structure; a pair \((i, j) \in \mu\) with \( j \neq i \) marks the fact that the membrane with label \( i \) contains a membrane with label \( j \);
- \( w_i \in O^* \), with \( 1 \leq i \leq n \), describes the multiset of objects placed in the membrane with label \( i \);
- \( R \) is a finite set of rules, where \( h, m \in H, a \in O \) and \( v \in O^* \):

- \([a \rightarrow v]_m \) (global evolution (gevol))

- \([a]_h \mid m \rightarrow [w]_h\mid m \) (endocytosis (endo))

- \([a]_h \mid m \rightarrow [w]_h\mid m \) (exocytosis (exo))

In [50], it is added the restriction \( |w| = 1 \) in the rules (b) and (c), and so the operations endo and exo are replaced by rendo and rexo, where \( r \) stands for restricted. Moreover, two rules are added for a higher control of the mobility:

- \([a]_h \mid m \rightarrow S \rightarrow [a]_h \mid m \) (inhibitory endocytosis (endo))
- \([a]_h \mid m \rightarrow S \rightarrow [a]_h \mid m \) (inhibitory exocytosis (exo))

If a membrane with label \( m \) does not contain any object from a set \( S \) of inhibitors, then a membrane with label \( h \) and an object \( a \) inside can enter the membrane with label \( m \) without any change of the membrane labels or their objects.

The rules are applied by using the principles listed below:

- A maximal parallel manner is used, namely no more rules can be added to the applicable multiset of rules.
- The moving membrane with label \( h \) is called active, while the membrane with label \( m \) and that does not contain the set of inhibitors \( S \) is called passive. In a maximal multiset of rules to be applied in a computation...
step, the objects and active membranes can be appear
only in one rule, while the passive membrane can appear
in several rules.

- All objects appearing in an elementary active mem-
  branes need to evolve first before the membrane is
  moved by exocytosis or endocytosis.
- Once a membrane exits the skin membrane it will not be
  implied in further application of rules.
- The objects not rewritten in a computational step can be
  used in the subsequent steps.

B. ENHANCED MOBILE MEMBRANES

The simple mobile membranes were extended to enhanced mobile membranes in [5] to be able to describe some bio-
logical processes of the immune system [55]. Inspired by the
approach presented in [5], the article [61] uses concepts from
membrane computing, but also Bio-PEPA and CARMA for
modelling the immune system.

Definition 3 [5]: An enhanced mobile membrane is a
tuple \( \Pi = (O, H, \mu, w_1, \ldots, w_n, R) \), such that:

- \( n, O, H, \mu, w_1, \ldots, w_n \) are similar to those from Definition 2;
- \( R \) is a finite set of rules, where \( h, m \in H, u \in O^+ \) and
  \( v, v', w, w' \in O^* \):

  \[
  \begin{align*}
  \text{– } & [u \rightarrow v]_m^h \quad \text{local evolution (levol)} \\
  \end{align*}
  \]

  The local character of the rule is given by the fact
  that a multiset of objects \( u \) is rewritten into a multi-
  set of object \( v \) only if it is place in a membrane with
  label \( m \) having as a parent a membrane with label \( h \).
  If membrane \( h \) is not required in applying the rule, 
  then this is a global evolution rule as in Definition 2.

  \[
  \begin{align*}
  \text{– } & [uv]_h[v']_m \rightarrow [w]_m[w']_m \quad \text{endocytosis (endo)} \\
  \end{align*}
  \]

  The multisets of objects \( uv \) and \( v' \) placed in the
  membranes with labels \( h \) and \( m \), respectively, are
  rewritten into the multisets of objects \( w \) and \( w' \)
  placed in the membranes with labels \( h \) and \( m \),
  respectively. At the same time, the elementary
  membrane with label \( h \) exits the membrane with
  label \( m \) without any change of the membrane labels.

  \[
  \begin{align*}
  \text{– } & [v]_h[uv']_m \rightarrow [w]_h[w']_m \quad \text{forced endocytosis (fendo)} \\
  \end{align*}
  \]

  Similar form as rule (b) except multiset \( u \) is not
  placed in the moving membrane with label \( h \).

  \[
  \begin{align*}
  \text{– } & [v]_h[uv']_m \rightarrow [w]_m[w']_m \quad \text{forced exocytosis (fexo)} \\
  \end{align*}
  \]

  Similar form as rule (c) except multiset \( u \) is not
  placed in the moving membrane with label \( h \).

The rules of enhanced mobile membranes are applied using
the principles given in Subsection III-A for simple mobile
membranes.

In [23] it is added the restriction \(|w + w'| = 1\) in the rules
\((d)\) and \((e)\), and so the operations fendo and fexo are replaced by rfendo and rfexo, where \( r \) stands for restricted.

C. MUTUAL MOBILE MEMBRANES

It is worth noting that the rules of the enhanced mobile
membranes allow a membrane to move without any permission
from the other involved membrane. In contrast with such an
approach, we introduce the mutual mobile membranes [6]
where the movement is performed only if there is a mutual
agreement between membranes. This means to use multisets
of objects \( u \) and \( \overline{u} \) where the multiset \( u \) marks the
membrane that initializes the move, while a multiset of \( \overline{u} \) marks
the membrane that accepts the movement. Due to the equality
\( \overline{u} = u \), fewer rules are needed (than for enhanced mobile
membranes).

Definition 4 [6]: A mutual mobile membrane is a tuple
\( \Pi = (O, H, \mu, w_1, \ldots, w_n, R) \), such that :

- \( n, O, H, \mu, w_1, \ldots, w_n \) are similar to those from Definition 2;
- \( R \) is a finite set of rules, where \( h, m \in H, u, \overline{u} \in
  V^+, v, v', w, w' \in V^* \):

  \[
  \begin{align*}
  \text{– } & [u \rightarrow v]_m^h \quad \text{local evolution (levol)} \\
  \end{align*}
  \]

  Similar to that in Definition 3.

  \[
  \begin{align*}
  \text{– } & [uv]_h[\overline{v}]_m \rightarrow [w]_m[w']_m \quad \text{mutual endocytosis (mendo)} \\
  \end{align*}
  \]

  The multisets of objects \( uv \) and \( \overline{v} \) placed in the
  membranes with labels \( h \) and \( m \), respectively, are
  rewritten into the multisets of objects \( w \) and \( w' \)
  placed in the membranes with labels \( h \) and \( m \),
  respectively. At the same time, the elementary
  membrane with label \( h \) enters the adjacent mem-
  brane with label \( m \) without any change of the mem-
  brane labels.
The rules of mutual mobile membranes with objects on surface are applied using the principles given in Subsection III-A for simple mobile membranes.

In [29], the authors consider slightly different versions of the above rules together with the rules inspired by the operations \( \text{mate} \) and \( \text{drip} \) from the brane calculus [27]:

\[
\begin{align*}
\text{pino}_a & \rightarrow [l_{aav} \rightarrow [l_{acv} v]] \\
\text{pino}_b & \rightarrow [l_{aav} \rightarrow [l_{aav}]] \\
\text{expo}_a & \rightarrow [l_{aav} \rightarrow [l_{aav} v]] \\
\text{expo}_b & \rightarrow [l_{aav} \rightarrow [l_{aav}]] \\
\text{mate} & \rightarrow [l_{aav} \rightarrow [l_{aav} v]] \\
\text{drip} & \rightarrow [l_{aav} \rightarrow [l_{aav}]]
\end{align*}
\]

These operations are used in [40] to describe mitochondrial fusion in the membrane automata starting from a similar approach done in BioAmbients [1]. Other extensions of these operations are defined in [48].

IV. EFFICIENCY IN SOLVING NP-COMPLETE PROBLEMS

Based on the size of their input, the NP-complete problems can be classified into two categories: \textit{weak} (e.g., Subset Sum, Knapsack, Partition) and \textit{strong} (e.g., SAT, Bin Packing, Clique, Common Algorithmic Problem) [39]. A comprehensive survey presenting solutions to NP-complete problems obtained in polynomial-time is presented in [62]. The solutions are achieved by using P systems with active membranes with electrical charges together with rules modelling membrane creation and membrane division.

To obtain such solutions, the membrane systems need to respect the conditions: (i) all computations reach a halting configuration; (ii) beside the working alphabet also two objects \( \text{yes} \) and \( \text{no} \) need to be added to mark the response given by the algorithm; only one of these two objects can appear in the halting configuration; (iii) in the halting configuration the presence of the object \( \text{yes} \) marks a successful computation), while the presence of the object \( \text{no} \) marks an unsuccessful computation.

In addition to using mobility rules, also elementary division rules are added for generating the working space needed to tackle NP-complete problems:

\[
\begin{align*}
\text{ediv} & \rightarrow [l_{aw} \rightarrow [l_{aw} w']] \\
\text{elementary division (ediv)} & \rightarrow [l_{aw} \rightarrow [l_{aw} w']] \\
\end{align*}
\]

An object \( a \) is rewritten to the multisets of objects \( w \) and \( w' \). The surrounding elementary membrane with label \( h \) is divided into two copies with the same label \( h \), such that each new membrane with label \( h \) contains a copy of each object from the initial membrane with label \( h \) and one of the multisets of objects \( w \) and \( w' \).

In what follows we present some NP-complete problems and the results claiming that they can be solved efficiently by using mobile membrane systems.

The \( k \)-Partition problem asks to decide whether or not there exists a partition of given a finite set \( A \) into \( k \) subsets such that they have the same weight according to a given weight function \( g : A \rightarrow \mathbb{N} \).
Theorem 1 [14]: By using mobile membrane systems with the rules mendo, mexo and ediv, the 2-Partition problem is solvable in a polynomial number of steps.

The Subset Sum problem asks to determine whether or not there exists a non-empty subset $B$ of a given finite set $A$ such that $g(B) = s$ for a given a weight function $g : A \to \mathbb{N}$ and constant $s$.

Theorem 2 [16]: By using mobile membrane systems with the rules mendo, mexo and ediv, the Subset Sum problem can be solved in a polynomial number of steps.

The Bin Packing problem asks to decide whether or not there exists a partition of a given finite set $A = \{a_1, \ldots, a_n\}$ into $b$ subsets such that their weights do not exceed $c$ for a given weight function $g : A \to \mathbb{N}$ and two constants $b, c \in \mathbb{N}$.

Theorem 3 [22]: By using mobile membrane systems with the rules mendo, mexo and ediv, the Bin Packing problem can be solved in a polynomial number of steps.

The Knapsack problem asks to decide whether or not there exists a subset of a given finite set $A = \{a_1, \ldots, a_n\}$ such that its weight does not exceed $k$ and its value is greater than or equal to $l$ for a given weight function $g : A \to \mathbb{N}$, value function $r : A \to \mathbb{N}$ and constant $l \in \mathbb{N}$.

Theorem 4 [16]: By using mobile membrane systems with the rules mendo, mexo and ediv, SAT can be solved in a polynomial number of steps.

The SAT problem ask if a propositional logic formula written in conjunctive normal form (CNF) is satisfiable. A formula $\varphi = C_1 \land C_2 \land \cdots \land C_m$ is in CNF if $C_i = \psi_1 \lor \psi_2 \lor \cdots \lor \psi_r$, for $1 \leq i \leq m$ and $r \leq n$, and either $\psi_j = x_k$ or $\psi_j = \neg x_k$, for a given set of propositional variables $X = \{x_1, x_2, \ldots, x_n\}$.

Theorem 5 [19]: By using mobile membrane systems with the rules mendo, mexo and ediv, SAT can be solved in a polynomial number of steps.

Theorem 6 [50]: By using mobile membrane systems with the rules rendo, rexo and ediv, SAT can be solved in a polynomial number of steps.

The kQBF problem asks if a quantified Boolean formulas with $k$ alternations of quantifiers is satisfiable. This means to check if there exists an assignment for a set of propositional variables $X = X_1 \cup \cdots \cup X_k$ to obtain the satisfiability of the formula $\varphi = \exists x_1 \forall x_2 \exists x_3 \cdots \psi$, where $\psi$ is in CNF and the set of variables $X$ is partitioned into $k$ sets $X_1, \ldots, X_k$.

Theorem 7 [16]: By using mobile membrane systems with the rules mendo, mexo and ediv, 2QBF can be solved in a polynomial number of steps.

Theorem 8 [50]: By using mobile membrane systems with the rules rendo, rexo, iendo and ediv, 2QBF can be solved in a polynomial number of steps.

Theorem 9 [23]: By using mobile membrane systems with the rules rendo, rexo, rfendo, rfexo, iendo and ediv, 4QBF can be solved in a polynomial number of steps.

V. COMPUTABILITY POWER

In this section we present some existing results related to the computational power of the different variants of mobile mem-

branes: simple, enhanced, mutual and mutual with objects on surface.

A. SIMPLE MOBILE MEMBRANES

$\text{PsSMM}_n(\alpha)$ denotes the family of sets $\text{Ps}(\Pi)$ generated by systems with at most $n$ membranes and evolving by rules of the type $\alpha \subseteq \{\text{gevol}, \text{levol}, \text{exo}, \text{endo}\}$. The subscript $n$ can be replaced by $\ast$ if the number of membranes is not fixed but finite. The rules of type $\alpha$ do not lead to an increase in the number of membranes during the computation, but as membranes can exit the skin membrane the number of membranes can decrease during computation.

Theorem 10 [51]:

\[
\text{PsSMM}_n(\text{exo}, \text{endo}) = \text{PsRE}.
\]

Corollary 1 [51]:

\[
\text{PsSMM}_n(\text{gevol}, \text{exo}, \text{endo}) = \text{PsSMM}_n(\text{levol}, \text{exo}, \text{endo}) = \text{PsSMM}_n(\text{exo}, \text{endo}) = \text{PsRE}, \quad \text{for all } n \geq 9.
\]

When the global evolution rules are used, the computational universality is obtained with only four membranes.

Theorem 11 [46]:

\[
\text{PsSMM}_4(\text{gevol}, \text{exo}, \text{endo}) = \text{PsRE}.
\]

By using the local evolution rules, an improvement is obtained as the number of membranes decreases to three (the minimal number for mobility).

Theorem 12 [9]:

\[
\text{PsSMM}_3(\text{levol}, \text{exo}, \text{endo}) = \text{PsRE}.
\]

B. ENHANCED MOBILE MEMBRANES

$\text{PsSMM}_n(\alpha)$ denotes the family of all sets $\text{Ps}(\Pi)$ generated by systems with at most $n$ membranes and evolving by rules of the type $\alpha \subseteq \{\text{gevol}, \text{exo}, \text{iendo}, \text{exo}, \text{fbedo}, \text{fexo}, \text{rfexo}, \text{endo}, \text{rendo}, \text{iendo}, \text{fendo}, \text{rfendo}, \text{fexo}, \text{ediv}, \text{ediv}\}$. The computational universality is obtained with twelve membranes by using the rules exo, endo, fendo and fexo.

Theorem 13 [32]:

\[
\text{PsEMM}_{12}(\text{exo}, \text{fexo}, \text{endo}, \text{fendo}) = \text{PsRE}.
\]

The result is improved by obtaining the same power with nine membranes.

Theorem 14 [9]:

\[
\text{PsEMM}_9(\text{exo}, \text{fexo}, \text{endo}, \text{fendo}) = \text{PsRE}.
\]

When using only the global evolution rules without any mobility rules, only three membranes are enough to obtain the computational universality.

Theorem 15 [32]:

\[
\text{PsEMM}_3(\text{gevol}) = \text{PsRE}.
\]
An interesting aspect is that the systems with three membranes using either the pairs \((exo, endo)\) or \((fexo, fendo)\) have the same computational power.

Theorem 16 [32]:

\[
PsEMM_3(exo, endo) = PsEMM_3(fexo, fendo).
\]

Systems with eight membranes subsume \(PsET0L\), while those with seven membranes subsume \(PsE0L\).

Theorem 17 [32]:

\[
PsET0L \subseteq PsEMM_8(exo, fexo, endo, fendo).
\]

Corollary 2 [32]:

\[
PsE0L \subseteq PsEMM_7(exo, fexo, endo, fendo).
\]

On the other hand, the system with three membranes are contained in \(PsMAT\).

Theorem 18 [32]:

\[
PsEMM_3(exo, fexo, endo, fendo) \subseteq PsMAT.
\]

When controlled and restricted rules are added, some interesting results are obtained with respect to the computed sets of natural numbers.

Theorem 19 [49]:

\[
NEMM_4(exo, fexo, endo, fendo) = NRE.
\]

Theorem 20 [49]:

\[
NEMM_4(exo, iendo) = NRE.
\]

Theorem 21 [49]:

\[
NEMM_4(iexo, endo) = NRE.
\]

Theorem 22 [23]:

\[
NEMM_4(exo, fexo, endo, fendo) \subseteq NMAT.
\]

Theorem 23 [23]:

\[
NEMM_4(rexo, rfexo, iexo, rendo, rfendo, iendo) \subseteq NFIN.
\]

Theorem 24 [23]:

\[
NEMM_4(rexo, rfexo, iexo, rendo, rfendo, iendo, ediv) \subseteq NCS.
\]

Theorem 25 [23]:

\[
NEMM_4(ediv) \subseteq NCS.
\]

Theorem 26 [49]:

\[
NEMM_4(rexo, rendo, rdiv) - NMAT^k \neq \emptyset.
\]

C. MUTUAL MOBILE MEMBRANES

\(PsMMM_n(\alpha)\) denotes the family of all sets \(Ps(\Pi)\) generated by systems with at most \(n\) membranes and evolving by rules of the type \(\alpha \subseteq \{pino, pino_c, phago, mate\}\). Due to the addition of co-objects, fewer membranes are needed (than for enhanced mobile membranes) in order to obtain computational universality.

Theorem 27 [9]:

\[
PsMMM_7(exo, mendo) = PsRE.
\]

This result was improved significantly such that only three membranes suffice.

Theorem 28 [8]:

\[
PsMMM_3(exo, mendo) = PsRE.
\]

Moreover, systems with three membranes subsume \(PsET0L\).

Proposition 1 [8]:

\[
PsET0L \subseteq PsMMM_3(mexo, mendo).
\]

In [7] was proven also that adding timers to mutual mobile membranes does not lead to an increase of the computational power.

D. MUTUAL MOBILE MEMBRANES WITH OBJECTS ON SURFACE

\(PsM^3OS_n(r_1(s_1), r_2(s_2))\) denotes the family of all sets \(Ps(\Pi)\) generated by systems with at most \(n\) membranes and evolving by rules of the type \(r_1 \in \{pino, pino_c, phago, mate\}\) and \(r_2 \in \{exo, exo_i, exo_c, drip\}\) of weight at most \(s_1\) and \(s_2\), respectively. The weight of a rule is given by the number of objects of its right-hand side. The computational universality of the pair of operations \((pino, exo)\) where the weight of \(pino\) is at most four and the weight of \(exo\) is at most three is obtained by using at most eight membranes.

Theorem 29 [47]:

\[
PsRE = PsM^3OS_n(pino(s_1), exo(s_2)),
\]

for all \(m \geq 8, s_1 \geq 4, s_2 \geq 3\).

By allowing an increase of the weight in the \(pino\) rules from four to five, and an increase of the weight in the \(exo\) rules from three to four, the number of membranes is reduced to only three.

Theorem 30 [15]:

\[
PsRE = PsM^3OS_n(pino(s_1), exo(s_2)),
\]

for all \(m \geq 3, s_1 \geq 5, s_2 \geq 4\).

By allowing a decrease of the weight in the \(pino\) rules from five to four, the number of membranes remains the same (namely, three).

Theorem 31 [13]:

\[
PsRE = PsM^3OS_n(pino(s_1), exo(s_2)),
\]

for all \(m \geq 3, s_1 \geq 4, s_2 \geq 4\).
The computational universality of the pair of operations (phago, exo) where the weight of phago is at most five and the weight of exo is at most two is obtained by using at most nine membranes.

Theorem 32 [47]:

\[ \text{PsRE} = \text{PsM}^{2} \text{OS}_{m}(\text{phago}(s_1), \text{exo}(s_2)), \]

for all \( m \geq 9, s_1 \geq 5, s_2 \geq 2. \)

A decrease by one of the weight for phago leads to an increase by one of the weight for exo in order to preserve the number of membranes.

Theorem 33 [47]:

\[ \text{PsRE} = \text{PsM}^{3} \text{OS}_{m}(\text{phago}(s_1), \text{exo}(s_2)), \]

for all \( m \geq 9, s_1 \geq 4, s_2 \geq 3. \)

An important reduction on the number of membranes is obtained by preserving the weight of exo operations and increasing the weight of the phago operation from four to six.

Theorem 34 [15]:

\[ \text{PsRE} = \text{PsM}^{4} \text{OS}_{m}(\text{phago}(s_1), \text{exo}(s_2)), \]

for all \( m \geq 4, s_1 \geq 6, s_2 \geq 3. \)

The same number of membranes is obtained while the weights of both phago and exo operations are decreased by one to five and two, respectively.

Theorem 35 [13]:

\[ \text{PsRE} = \text{PsM}^{5} \text{OS}_{m}(\text{phago}(s_1), \text{exo}(s_2)), \]

for all \( m \geq 4, s_1 \geq 5, s_2 \geq 2. \)

When using the operations mate and drop and the rules presented in [29], the following results are obtained.

Lemma 1 [29]:

\[ \text{PsM}^{6} \text{OS}_{m}(\text{mate}(s_1), \text{drop}(s_2)) \subseteq \text{PsM}^{6} \text{OS}_{m}(\text{mate}(s_1), \text{drop}(s_2)), \]

for all \( m \leq m', s_1 \leq s'_1 \) and \( s_2 \leq s'_2. \)

Theorem 36 [29]:

\[ \text{PsM}^{7} \text{OS}_{m}(\text{mate}(s_1); \text{drop}(s_2)) \subseteq \text{PsRE}, \]

for all \( m \geq 11, s_1 \geq 5 \) and \( s_2 \geq 5. \)

Theorem 37 [58]:

\[ \text{NRE} = \text{NM}^{2} \text{OS}_{m}(\text{mate}(s_1), \text{drop}(s_2)) \]

for all \( m \geq 5, s_1 \geq 4 \) and \( s_2 \geq 4. \)

Some related results are presented in [48] by using some variants of the mobility rules in which the objects are placed inside membranes.

VI. RELATIONSHIPS WITH OTHER APPROACHES

One of the research line in membrane systems is to connect mobile membranes with process calculi; e.g., [31], Petri nets [2], [45], cellular automata [34] and communicating X-machines [44]. Several studies focused on the relationships between these models in order to use their techniques and results: e.g., model checking of process calculi, invariants of Petri nets, testing methods of X-Machines. In what follows we present some connections following this line of research.

A. BRANE CALCULI AS MUTUAL MOBILE MEMBRANES WITH OBJECTS ON SURFACE

There exist articles treating similar aspects existing in both membrane systems and brane calculus: [24]–[26], [29], [30], [47]. Related to this research line, we present how to use mutual mobile membranes with objects on surface to encode the PEP fragment of brane calculus.

When studying the evolution of mobile membranes, the notion of membrane configuration is central. The membrane configurations of the set \( M = \{ M, N, \ldots \} \) are defined using the finite alphabet \( O = \{ a, \pi, b, \ldots \} \) of objects:

\[ M, N ::= [ ]_u | [ M ]_u | M N. \]

A configuration \([ M ]_u\) describes a membrane \( M \) with objects \( u \) on its surface. The configuration \( M N \) models two (possibly composite) sibling membranes.

Considering membranes with objects from \( O \) and rules from \( R \), by \( M \rightarrow_m N \) is denoted the reduction of a configuration \( M \) to a configuration \( N \) using a rule from the set \( R \). The evolution is given by the following rules:

\[
\begin{align*}
\text{(Mem)} & : & M & \rightarrow_m M_1 \\
& & [ M ]_u & \rightarrow_m [ M_1 ]_u \\
& & M & \rightarrow_m M_1 & N & \rightarrow_m N_1 \\
\text{(Comp)} & : & M N & \rightarrow_m M_1 N_1 \\
& & M & \equiv_m M_1 & M_1 & \rightarrow_m N_1 & N_1 & \equiv_m N \\
& & M N & \rightarrow_m M_1 \\
\text{(Struct)} & : & M N & \rightarrow_m M N \\
\end{align*}
\]

where the membranes can be rearranged by using the structural congruence \( \equiv_m \) presented in Table 4.

| TABLE 4. Structural congruence over mutual mobile membranes. |
|-----------------|-----------------|-----------------|-----------------|
| \( u v \equiv_m u u v \) & \( u (u w) \equiv_m (u w) w \) & \( u v \equiv_m \) implies \( u w \equiv_m v w \) \& \( u \equiv_m u \) & \( M N \equiv_m M N \) \& \( M (N P) \equiv_m (M N) P \) & \( M \equiv_m N \) implies \( M P \equiv_m N P \) & \( M \equiv_m N \) & \( u \equiv_m v \) implies \( [M]_u \equiv_m [N]_v \) |

Using the above membrane configurations and the evolution rules for mobile membranes, an encoding of the PEP fragment into the mutual mobile membranes with objects on surface can be achieved. For this purpose, we define a translation function that transforms the brane processes of the set \( P \) into membrane configurations from the set \( M \).

Definition 6 [11]: A translation function \( T : P \rightarrow M \) is defined as

\[
T(P) = \begin{cases} 
[ ]_S(\sigma) & \text{if } P = [ ]_S(\sigma) \\
(T(R))_S(\sigma) & \text{if } P = (T(R))_S(\sigma) \\
(T(Q) T(R)) & \text{if } P = Q \mid R, 
\end{cases}
\]
where $S : \mathcal{P} \to O$ is defined by

$$
S(\sigma) = \begin{cases}
\lambda & \text{if } \sigma = 0 \\
\sigma & \text{if } \sigma = n \setminus \sigma = n \setminus \sigma = \pi \setminus \\
\pi \setminus S(\rho) & \text{if } \sigma = \pi \setminus S(\rho) \\
p \setminus S(\rho) & \text{if } \sigma = p \setminus S(\rho) \\
S(\alpha) S(\rho) & \text{if } \sigma = \alpha, \rho \\
S(\tau) S(\rho) & \text{if } \sigma = \tau \setminus \rho 
\end{cases}
$$

The translation function $T$ can be used to transform the first three rules of Table 2 into the following rules:

$$
\begin{align*}
\mathcal{I}_{pino} \pi S(\rho) S(\sigma|\tau_0) & \rightarrow_m \mathcal{I}_{pino} \pi S(\sigma|\tau_0), \\
\mathcal{I}_{n} \setminus S(\sigma|\tau_0) S(\tau|\tau_0) & \rightarrow_m \mathcal{I} S(\sigma|\tau_0), \\
\mathcal{I}_{\pi \setminus} \setminus S(\sigma|\tau_0) S(\tau|\tau_0) & \rightarrow_m \mathcal{I} S(\sigma|\tau_0) S(\rho) S(\tau|\tau_0).
\end{align*}
$$

It holds that two structurally congruent PEP systems are translated into two congruent configurations of mutual mobile membranes with objects on surface.

**Proposition 2** [11]: If $M = T(P)$, there is $N$ such that $M \equiv_m N$ and $N = T(Q)$ whenever $P \equiv_b Q$.

**Proposition 3** [11]: If $M = T(P)$, there is $Q$ such that $N = T(Q)$ whenever $M \equiv_m N$.

It holds also that if a PEP system reduces to another one by applying a rule, the membrane configurations obtained by translation can evolve one into another by using the translated rule.

**Proposition 4** [11]: If $M = T(P)$, there is $N$ such that $M \rightarrow_m N$ and $N = T(Q)$ whenever $P \rightarrow_b Q$.

**Proposition 5** [11]: If $M = T(P)$, there is $Q$ such that $N = T(Q)$ whenever $M \rightarrow_m N$.

### B. MOBILE AMBIENTS AS MOBILE MEMBRANES

The relationship between mobile membranes and mobile ambients is studied in [4], [31]. This relationship is achieved by translating the pure safe ambients into mobile membranes.

**Definition 7** [11]: A translation $T : A \rightarrow M$ is defined as $T(A) = \text{lock} \, T_1(A)$, where $T_1 : A \rightarrow M$ is

$$
T_1(A) = \begin{cases}
\mathcal{I}_{n} & \text{if } A = n[ ] \\
\mathcal{T}_1(A_1) \mathcal{I}_{n} & \text{if } A = n[A_1] \\
\mathcal{C} \mathcal{P} \mathcal{A} n \mathcal{C} \mathcal{P} \mathcal{A} n & \text{if } A = \mathcal{C} \mathcal{P} n \\
\mathcal{C} \mathcal{P} n \mathcal{C} \mathcal{P} \mathcal{A} n \mathcal{C} \mathcal{P} \mathcal{A} n & \text{if } A = \mathcal{C} \mathcal{P} n.A_1 \\
\mathcal{T}_1(A_1), \mathcal{T}_1(A_2) & \text{if } A = A_1 | A_2.
\end{cases}
$$

The unique object $\text{lock}$ is used to control the evolution by blocking the application of rules corresponding to a mobile ambient that cannot evolve.

The structural congruence is preserved by this translation.

**Proposition 6** [31]:

$A \equiv_{amb} B$ if and only if $T(A) \equiv_{mem} T(B)$.

If $M = \text{lock} \, M_1$ and $N = \text{lock} \, N_1$ are two membrane configurations such that $M_1$ and $N_1$ do not contain the $\text{lock}$ object, then $M \Rightarrow_{mem} N$ denotes the application of certain rules $r_1, \ldots, r_n$ to obtain the membrane configuration $N$ starting from the membrane configuration $M$.

**Proposition 7** [11]:

If $A \Rightarrow_{amb} B$ and $M = T(A)$, then there is $N$ such that $M \Rightarrow_{mem} N$ and $N = T(B)$.

**Proposition 8** [11]:

If $M = T(A)$ and $M \Rightarrow_{mem} N$, then there is $B$ such that $A \Rightarrow_{amb} B$ and $N = T(B)$.

**Theorem 38** [31]: *(Operational Correspondence)*

1. If $A \Rightarrow_{amb} B$, then $T(A) \Rightarrow_{mem} T(B)$.
2. If $T(A) \Rightarrow_{mem} M$, then exists $B$ such that $A \Rightarrow_{amb} B$ and $M = T(B)$.

These results facilitate the use for membrane systems of some existing tools for mobile ambients. As a consequence, checking if a mobile membrane configuration can be obtained from another membrane configuration (the reachability problem) was treated in [3]. It was proven that the problem is decidable (by using the decidability achieved in a variant of the ambient calculus).

**Theorem 39** [3]: It is decidable whether an arbitrary mobile membrane $M_1$ reduces to another mobile membrane $M_2$.

Reachability aspects were also studied in [36] in the framework of BioAmbients [63], a biological inspired version of mobile ambients. We consider that a uniform analysis of BioAmbients and membrane systems having now different notions of reachability represents and interesting topic to approach.

Inspired by the relationship between process calculi and membrane systems, some observational equivalences were defined and studied in the context of mobile membranes with lifetimes [20], [21]. These observational equivalences correspond to the ability of observing combinations of mobility, timing and positions of the membranes. Similar equivalences were studied in [37], [38] for the bio-inspired formalism Bio-PEPA presented in [33].

### C. ENHANCED MOBILE MEMBRANES AS COLOURED PETRI NETS

In what follows we use $\text{lhs}(r)$ and $\text{rhs}(r)$ to represent the left-hand side and right-hand side of a rule $r$, respectively. A coloured Petri net $\text{CPN}_{\Pi} = (P, T, A, \Sigma, \mathcal{X}, \mathcal{C}, \mathcal{G}, E, I)$ can be constructed from a given system of enhanced mobile membranes $\Pi = (O, H, \mu, w_1, \ldots, w_n, R, i)$ as described in [10], [17]:

- $P = \{\text{structure} \} \cup \{1, \ldots, n\}$, where $\text{structure}$ contains $\{(i, j) \mid (i, j) \in \mu\}$
- $T = \bigcup_{1 \leq k \leq |\mathcal{R}|} t_k$, where for each $r_k \in \mathcal{R}$ a transition $t_k$ is constructed;
° the set A of directed arcs contains:
   - input arcs \((P \times T)\) from the place \(structure\) and from places corresponding to the membranes from \(lhs(r_k)\) to the transition \(t_k\),
   - output arcs \((T \times P)\) from the transition \(t_k\) to the place \(structure\) and to the places corresponding to the membranes from \(rhs(r_k)\);
° \(\Sigma = O \cup \mu;\)
° \(X = \{x, y, z, \ldots\}\), a finite set of variables used for checking the structure \(\mu;\)
° \(C(p) = \begin{cases} \mu & \text{if } p = \text{structure}, \\ O & \text{if } p \in \{1, \ldots, n\}; \\ \{ x = y \} & \text{if } t \text{ simulates an } \text{endo}, \text{fendo} \\ \text{or } \text{level rule,} & \text{true otherwise;} \end{cases}\)
° \(G(t) = \begin{cases} \text{true} & \text{if } \phi, \\ \text{false otherwise;} \end{cases}\)
° The set \(E\) is constructed as follows:
   - on the arc from the place \(structure\) to a transition \(t_k\), add the pairs \((i, j) \in \mu\) describing the membrane structure appearing in \(lhs(r_k)\);
   - on the arc from a place that represents a membrane containing an object \(a\) in \(lhs(r_k)\) to the transition \(t_k\), add the object \(a\);
   - on the arc from a transition \(t_k\) to the place \(structure\), add all the pairs \((i, j)\) describing the membrane structure appearing in \(rhs(r_k)\);
   - on the arc from a transition \(t_k\) to a place representing a membrane containing a multiset of objects \(w\) appearing in \(rhs(r_k)\), add the multiset of objects \(w\);}
[16] B. Aman and G. Ciobanu, “Mobility in process calculi and natural computing,” in Natural Computing Series, vol. 194, Berlin, Germany: Springer, 2011.

[17] B. Aman and G. Ciobanu, “Properties of enhanced mobile membranes via coloured Petri nets,” Inf. Process. Lett., vol. 112, no. 6, pp. 243–248, Mar. 2012.

[18] B. Aman and G. Ciobanu, “Mobile membranes with objects on surface as colored Petri nets,” in Proc. Int. Conf. Membrane Comput. in Lecture Notes in Computer Science, vol. 7762, E. Csete-Várúj, M. Gheorghe, G. Rozenberg, A. Salomaa, G. Vaszil, Eds. Budapest, Hungary, 2013, pp. 128–144.

[19] B. Aman and G. Ciobanu, “Mobile membranes: Computability and complexity,” in Proc. 10th Int. Colloq. Theor. Aspects Comput. (ICTAC) in Lecture Notes in Computer Science, vol. 8049, Z. Liu, J. Woodcock, and H. Zhu, Eds. Shanghai, China, 2013, pp. 59–75.

[20] B. Aman and G. Ciobanu, “Observational equivalences in a bio-inspired formalism involving mobility and lifetimes,” in Proc. 8th Int. Conf. Bio-Inspired Inf. Commun. Technol. (Formerly BIONETICS), 2015, pp. 173–176.

[21] B. Aman and G. Ciobanu, “Behavioural observations of cell movements with timing aspects,” Nano Commun. Netw., vol. 6, no. 3, pp. 96–102, Sep. 2015.

[22] B. Aman and G. Ciobanu, “Efficiently solving the bin packing problem through bio-inspired mobility,” Acta Inf., vol. 54, no. 4, pp. 435–445, Jun. 2017.

[23] B. Aman, G. Ciobanu, and S. N. Krishna, “Solving the QBPF problem in polynomial time by using the biological-inspired mobility,” in Proc. 9th Int. Conf. Distrib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[24] R. Brüder, M. Cavaliere, A. Riscos-Núñez, G. Rozenberg, and D. Sburlan, “Membrane systems with proteins embedded in membranes,” Theor. Comput. Sci., vol. 404, nos. 1–2, pp. 26–39, Sep. 2008.

[25] N. Busi, “On the computational power of the mate/bud/drip brane calculus: Interleaving vs. Maximal parallelism,” in Membrane Computing (Lecture Notes in Computer Science), vol. 3850, 2005, pp. 144–158.

[26] N. Busi and R. Gorrieri, “On the computational power of brane calculi,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[27] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[28] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[29] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[30] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[31] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[32] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[33] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[34] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[35] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[36] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[37] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.

[38] L. Cardelli, “Brane calculi. Interactions of biological membranes,” in Interleaving vs. Maximal parallelism,” in Proc. 9th Int. Conf. Distib. Comput. Internet Technol. (ICDCIT) in Lecture Notes in Computer Science, vol. 7753, C. Hota and P. K. Srivani, Eds. 2013, pp. 432–443.
BOGDAN AMAN graduated from the Faculty of Mathematics, Alexandru Ioan Cuza University, Iaşi, in 2007. He received the Ph.D. degree from the Romanian Academy (Iaşi Branch) under the supervision of Prof. G. Ciobanu, in 2009. He received a public recognition for his research with the 2013 Grigore Moisil Award of the Romanian Academy of Sciences and the 2019 International Membrane Computer Society (IMCS) Prize for the Theoretical Result of the Year. His main research interests are membrane computing, natural computing, process algebra, type systems, and other theoretical aspects of computer science.

GABRIEL CIOBANU is currently a Researcher at the Romanian Academy of Sciences (Iaşi Branch) and at Alexandru Ioan Cuza University of Iaşi. His research interests include distributed systems (process calculi), formal methods (semantics, logics), and natural computing (membrane systems). For his scientific contributions, he received awards from the Romanian Academy, in 2000, 2004, and 2013; Ad Astra Association, in 2018; and International Membrane Computer Society, in 2019. He is the Editor-in-Chief of the Scientific Annals of Computer Science. He is a member of Academia Europaea (the Academy of Europe).

* * *

[64] A. Salomaa, *Formal Languages*. New York, NY, USA: Academic, 1973.
[65] S. J. Singer and G. L. Nicolson, “The fluid mosaic model of the structure of cell membranes,” *Science*, vol. 175, no. 4023, pp. 720–731, Feb. 1972.