Membrane Computing and Its Applications

Kriti Bhargava, Nisheeth Saxena

Abstract: ***There are two types of computing; one is conventional and another is unconventional. Turing machine is a hypothetical device and a conventional model, model of computation; it is not a random access machine, so to simulate current day's basis systems; in a exact manner register machine had been defined, so they are conventional model of computing. Paradigms like DNA computing, Quantum computing, Membrane Computing, they are model of unconventional computing. These have comes an existence in their resent times and we have found a certain problems can tackle in a very efficient manner using this. The practical use of this is still a question and it's being exploded. Membrane computing which biologically inspired and mathematically motivated. Membrane structure is an essential ingredient of a membrane computing, having hierarchical arrangement of membranes, like a cell or tree. This mainly deals with distributed and parallel computing models. Parallelism is restricted in classical computers bounded by 'n' (16/64) number of processors. But when size of problems increases, then the number of processors can't increases. Unbounded parallelism is exists in DNA, Membrane computing. Because of unbounded parallelism, you can have DNA strands and they can exist within one cubic centimeter of a solution; millions of DNA strands can exists. Because of that the action state place parallel and you can have a number of operations performed simultaneously. The basic features are evolution rules and evolving objects encapsulated in compartments of membranes. These evolution rules are applied on multisets. It is a framework to reflecting a limb of a system model. Inspiration of membrane computing is the process which takes place in a cell, the reaction which develops in cell region. The processing of substances, energy and information in these regions, through the membranes which delimit them are computational processes. We try to simulate these processes and see that anything we can define in a formal manner using this model of computation. In this paper, briefly discuss about membrane computing applications like global clock, L system, solving optimization problems, solving SAT, HPP etc.***

Index Terms: ***Membrane Computing, P system, HPP, SAT, L system.***

I. INTRODUCTION

Membrane is a pliable sheet, like a structure, plays a role as a boundary, partition of an organism, or lining. Membrane computing is springing unorthodox computational approach that endues a platform for modeling individual systems. Membrane computing is a new computability technique which is inspired from biotic. The system model used for any computation here resembles a membrane structure and it is distributed, highly parallel computing model.Branches of Natural computing, under this comes evolutionary computing, molecular computing, neural computing, membrane computing. These models are inspired by a nature. Neural computing, nowadays neural network play a very important role in their field of computer science and they have come and take existence because of their simulation of brain in neural computing. It tries to simulate the way the brain works and the neural networks are implemented with the electronic silicon computer; the implementation is through electronic media.

Evolutionary computing try to solve an algorithm making use of representing an instance using a bit string object, and then they perform the operations of cross over insertion, selection etc. which happens in the actually evolution of humans or any other organism in nature. Because they simulate the evolution, that's why it is called evolutionary computing and genetic algorithms command this. Genetic algorithms are mostly heuristical algorithms and they give very good suggestions to some problems because they converge. Even the conversions can't be properly proved and most the examples where it is considered. It is found that by taking the model in a suitable manner and the presenting this string in a suitable manner, the conversion can be achieved. For network algorithms like routing etc. evolutionary computing has been used. It is a very good research area nowadays.

Molecule computing is computing with natural objects. The actual things is molecules and how they behave and interact with each other, giving solutions were problems, that is called molecular computing, DNA computing etc. they all are commanded this. The implementation is really using the DNA strands molecules; so implementation is through by bio-media.

New branch of natural computing is a membrane computing, which introduced by G. Păun [1] in 1998. That's why; it is also called as 'P System'. It tries to simulate the behavior of cell, what happens in a cell the proteins, the membrane structure; how they develop; how they pass through membranes, changing from one to another etc. Initial theoretical model was developed and the first paper in this area was published in 2000. At the time it was started that there was two possibilities, one it could be implemented using the silicon computers like evolutionary algorithms etc. The other one is, it may actually use membranes are cells and try to simulate the algorithms. There was a question in 2000 and in 2014 now; this looks that is not possible at all how to make their algorithms be simulated by actual cells. Whereas this looks more promising i.e. this may develop as a bio-inspired model of computing like evolutionary computing or neural computing and it will be implemented in electronic media.

Behindhand the concept of a membrane is from biology, but generalized concept is that interpreting a membrane like a separator of two compartments. There are many compartments in a membrane structure, each compartment contain its own objects, which discretely evolved by specified evolution rules. The evolution rules are germane in a non-deterministic and maximally parallel way, which means all the objects that can be evolve, must evolve.

Manuscript Received on April, 2014.

Kriti Bhargava, Department of Computer Science & Engineering, Faculty of Engineering & Technology, Mody University of Science and Technology, Lakshmangarh, Sikar, Rajasthan, India.

Nisheeth Saxena, Department of Computer Science & Engineering, Faculty of Engineering & Technology, Mody University of Science and Technology, Lakshmangarh, Sikar, Rajasthan, India.

Motivation behind this new paradigm of computing is that a cell has complex structure, with several compartments delimited inside the main membrane by several inner membranes: the nucleus, the Golgi apparatus, several vesicles etc.

In principal, all these membranes are similar, so we consider the plasma membrane and consider its structure and functions.

Most common problems are in engineering design and scientific research, to solving optimization problem. Particle swarm optimization [2] was proposed, which based on swarm intelligence, as the framework of computing model. For multi-particle swarms a new hybrid model Membrane Computing Based Particle Swarm Optimization was introduced to improve performance [3]. Multi-objective optimization problem are solved by multi-objective bio-inspired algorithm with high accuracy and cost efficient [4].

Changing or variation in an evolution rules and evolution objects can make a new design of computing system model. Many various models of membrane computing have been introduced, such as cell-like MS (membrane structure), tissue-like P Systems Neural-like P system [1].

II. P SYSTEM

P system [5] is a class of distributed parallel computing devices of bio chemical type. Because whatever happens within the membranes occurs in parallel and then messages pass from inside to outside of each membrane and crossing the membrane, they carry some information. P system used distributed and parallel computing models where basic objects or data structures are numerical variables, strings, symbols, array, graph or multisets. Objects inside the membrane are atomic in nature and they can have structure.

A P system is defined as a series of membranes containing substances (ions, small molecules, chemicals (in finite quantity), numerical variables [6], biological networks [7] and macromolecules), catalysts objects and evolution rules which determine possible ways in which chemicals may react with one another to form products [8, 9]. A P system continues until it reaches a state where no further reactions are feasible. At this point the result of the computation is all those chemicals that have been passed outside of the skin membrane, or otherwise those passed into a designated 'result' membrane [1].

❖Theoretical :-

1) The parts of a biological system are well delimited by various types of membrane.
2) In broad sense of the term, membranes starts delimit the various intra-cell components, going to the cell membrane and then to skin of organisms.
3) Ending with more or less virtual membranes which delimit parts of ecosystem.

❖ Practical :-

1) In biology and chemistry, one knows membranes which keep together certain chemicals and allow other chemicals to pass in a selective manner or sometimes only in one direction.

A. How defined a Membrane

Membrane structure will represent in a Venn diagram. Membranes delimited regions; they have the objects or data structure. These are evolved by evolution rules. Through evolution, the elevated new object or data structure may stay in the same membrane or may relocate in another compartment [10, 11]. Starting with a certain number of objects in certain membranes let the system evolve. If it halts, then the computation is finished. The result can be viewed either in some membrane region or outside skin membrane. If the development of the system goes forever, then the computation fails to produce an output.

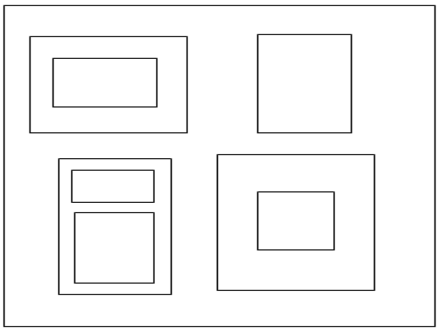


Figure 1: A Venn diagram.

B. Computation Method

In formal definition [12], the regions delimited by membranes, objects are placed. Hence, the object of evolution may be strings, multisets, etc. These evolve based on rules placed in the regions. By varying the evolving objects and evolution rules, several variants of P systems or membrane systems have been developed. In any variant, the objects are assumed to evolve; each object can be transformed into other objects, or can dissolve the membrane in which it is placed. The evolution is done in parallel for all objects which can evolve.

C. Operations

a.) Operations over Membrane

For two multisets M_1, M_2 over the same set X, we say that M_1 is included in M_2:

$$M_1\ M_2, if\ M_1(a) \leq\ M_2(a); for\ all\ a \in X$$

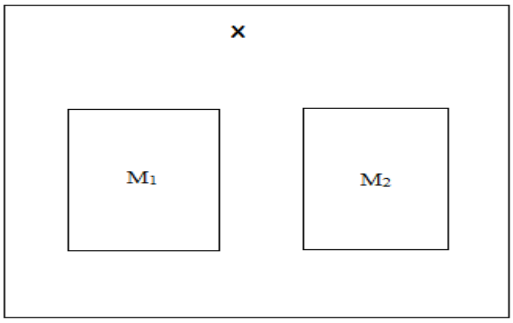


Figure 2: Two multisets M_1, M_2 over same set X.

- **Union (∪)**

$$M_1 \cup\ M_2; (M_1 \cup\ M_2)(a) =\ M_1(a) +\ M_2(a)$$

- **Intersection (∩)**

$$M_2 \cap\ M_1; (M_2 \cap M_1)(a) =\ M_2(a) -\ M_1(a)$$

(only if $M_1\ \subseteq\ M_2$)

- **Complementation (c)**

$$({M_1}^c)^c\ =\ M_1\ or\ ({M_2}^c)^c\ =\ M_2$$

b.) Operations with Strings or data objects

We now introduce the basic structural ingredient of the computing devices. Let us consider first the language MS over

the alphabet {[,]}, whose strings are defined as follows:
1. [ ] $\in$ MS
2. If $\mu_1 \ldots\ldots \mu_n \in MS, n \geq 1$, then $[\mu_1 \ldots\ldots \mu_n] \in MS$
3. Nothing else is in MS.
Relation over the elements of MS: $x \sim y$ if and only if we can write the two strings in the form $x = \mu_1\mu_2\mu_3\mu_4, y = \mu_1\mu_2\mu_3\mu_4, for\ \mu_1\mu_2 \in MS(x)$ and $\mu_3\mu_4 \in MS(y)$. We also denote by ~ the reflexive and transitive closure of the relation. This is an equivalence relation. We denoted by $\overline{MS}$ the set of equivalence classes of MS with respect to this relation. The elements of $\overline{MS}$ are called 'membrane structure' [12].

D.Components

a.) The environment

The environment is the surroundings of the P system. Initial state of a P system it contains only the blank-membrane with a boundary like a vessel, and while the environment can never keep rules; it may have objects passed into it during the computation. The objects cast within the environment at the end of the computation constitute all or part of its result.

b.) Membranes

Membranes are the main 'structure' within a P system. It is a discrete unit which can contain a set of objects (symbols/ catalysts), a set of rules, a set of other membranes contained within. The outer most membrane, held within the environment, is often referred to as the 'skin membrane'. As informed by their namesake, membranes are vulnerable and symbols resulting from a rule may cross them. A membrane (but not the container membrane) may also "dissolve", in which case it's content, except for rules (which are missing), migrate into the membrane in which it was contained.

c.) Symbols

Symbols represent chemicals which may react with other chemicals. In a P system each type of symbol is typically represented by a different letter. It (symbol) content of a membrane is therefore represented by a string of letters or data. Multisets are commonly used to represent the symbol content of a region.

d.) Catalysts Objects

Catalysts are similar to their namesakes in chemistry. They are used in the same way as symbols, but are never consumed during a 'reaction'; they are simply a requirement for it to occur.

e.) Rules

Rules represent a possible chemical reaction within a membrane, because of it to evolve to a new state. A rule has an expected set of input objects (symbols or catalysts) which must be present in order for it to be applied. If the expected objects are present, it consumes them and generates a set of output objects.

E. Parameters[13]

a.) Membrane

Each matching pair of parentheses [,] appearing in a membrane structure is called a "membrane".

b.) Degree of membrane

The number of membranes in a membrane structure μ is called the degree of μ and is denoted by deg(μ).

c.) Skin Membrane

The outer most membrane of a membrane structure μ is called the skin membrane.

d.) Elementary membrane

A membrane, which appears in $\mu \in (\overline{MS})$ in the form [ ] (no other membrane appears inside the two parentheses) is called an elementary membrane.

e.) Depth of membrane

The depth of a membrane structure μ, denoted by dep(μ) is defined as follows [10]:
1.) If $\mu = [\,]$, then dep(μ)=1
2.) If $\mu = [\mu_1 \ldots\ldots \mu_n]$, for some $\mu_1 \ldots\ldots \mu_n \in MS$
3.) than dep(μ)=max{dep(μ$_i$) | 1≤ i ≤ n}+1

f.) Region

The Venn diagram representation of a membrane structure (μ): any closed space delimited by membrane is called a region of μ membrane structure. A membrane structure of degree 'n' contains 'n' internal regions, one associated with each membrane.

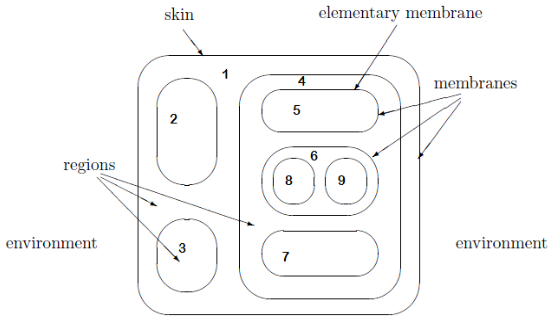


Figure 3: A Membrane Structure [13].

g.) Super-cell

Let U be a finite set whose elements are called objects. Consider a membrane structure μ of degree 'n', with the membranes labeled in a one-to-one manner, with the numbers from 1 to n. This way, the regions of μ are also identified by the numbers from 1 to n. If a multiset Mi: U→ S associated with each region i of μ, i from 1 to n, then we say that we have a super-cell.

III. TREE REPRESENTATION

As figure 3, a membrane structure is a hierarchically arranged set of membranes. Several membranes can be placed inside the skin membrane. The membranes are identified by labels from a given set of labels. In figure 3, we use numbers, starting with number 1 assigned to skin membrane and other labels are assigned in a one-to-one manner to membranes.
A hierarchical structure of membranes can be represented by a rooted tree, which describes the membrane structure from figure 3. The root of the tree is associated with the skin membrane and the leaves with the elementary membranes.

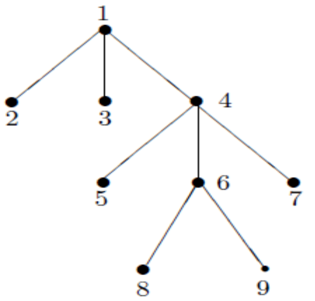


Figure 4: The tree describing the membrane structure from figure 3 [13].

The tree representation is the symbolic representation of a membrane structure, by string of labeled matching parentheses. For example, a string corresponding to the structure from figure 3 is following expression:

$[_1 [_4 [_5]_5 [_6 [_8]_8 [_9]_9]_6 [_7]_7]_4 [_2]_2 [_3]_3]_1$

Or

$[_1 [_3]_3 [_4 [_6 [_8]_8 [_9]_9]_6 [_7]_7 [_5]_5]_4 [_2]_2]_1$

All membranes are labeled; here we have used the numbers from 1 to 9. Membranes 2, 3, 5, 7, 8 and 9 are elementary. We say that the number of membranes is the degree of the membrane structure [13].

IV. EXAMPLE

The image shown depicts the initial state of a P system with three membranes [1, 10]. The outermost membrane 1 is the container membrane for this P system and contains a single last out rule. Membrane 2 contains four rules here, with two in a priority relationship: $cc \rightarrow c$ will always be applied in preference to $c \rightarrow \delta$. The delta symbol represents the special “decompose” symbol. The inlying membrane, 3, contains a set of symbols (“ac”) and here three set of rules. In this initial state no rules outside of membrane 3 are applicable: there are no symbols outside of that membrane. In spite of, during process of the system, as symbols are passed between membranes, in other membranes the rules will become active.

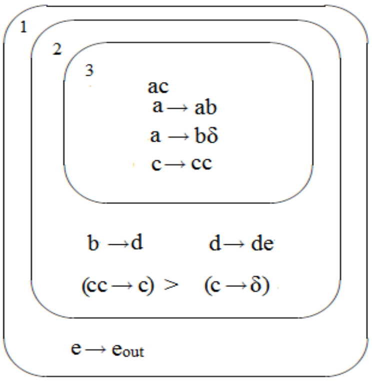


Figure 5: A graphical representation of P system.

Computation:

Because of the non-deterministic nature of P systems; there are many various paths of computation a single P system is competent of, leading to various results. The following is one possible path of computation for the P system depicted.

Step 1:

By the initial configuration, only membrane 3 has any object content: "ac"

"c" is assigned to $c \rightarrow cc$

"a" is assigned to $a \rightarrow ab$

Step 2:

Membrane 3 now contains: "abcc"

"a" is assigned to $a \rightarrow b\delta$

"c" is assigned to $c \rightarrow cc$

"c" is assigned to $c \rightarrow cc$

The application of the second rule ($a \rightarrow b\delta$) as opposed to the first ($a \rightarrow ab$) is non-deterministic and can be presumptive random. As well, the system could continued applying the first rule (and at the same time doubling the c particles) indefinitely.

Step 3:

Membrane 2 now contains: "bbcccc"

"b" is assigned to $b \rightarrow d$

"b" is assigned to $b \rightarrow d$

"cc" is assigned to $cc \rightarrow c$

"cc" is assigned to $cc \rightarrow c$

Step 4:

Membrane 2 now contains: "ddcc"

"d" is assigned to d → de

"d" is assigned to $d \rightarrow de$

"cc" is assigned to $cc \rightarrow c$

Step 5:

Membrane 2 now contains: "dedec"

"d" is assigned to $d \rightarrow de$

"d" is assigned to $d \rightarrow de$

"c" is assigned to $c \rightarrow \delta$

Notice that the priority over $c \rightarrow \delta$ has been lifted now the required inputs for $cc \rightarrow c$ further not exist. Now, membrane 2 decomposes, and all object content passes to membrane 1.

Step 6:

Membrane 1 now contains: "deedee"

"e" is assigned to $e \rightarrow e_{out}$

"e" is assigned to $e \rightarrow e_{out}$

"e" is assigned to $e \rightarrow e_{out}$

"e" is assigned to $e \rightarrow e_{out}$

Computation Halts:

Membrane 1 now contains: "dd" and, due to the out rule $e \rightarrow e_{out}$, the environment contains: "eeee." The computation halts at this point, as no further assignments of objects to rules is possible. The result of the computation is four "e" symbols.

The only non-deterministic choices occurred during steps 1 and 2 in starting, when choosing where to assign the solitary "a" symbol. Consider the case where "a" is assigned to $a \rightarrow b\delta$ during step 1: upon membrane 3 dissolving only a single "b" and two "c", both objects would exist, further to the creation of only a single "e" object to eventually be passed out as the computation’s result.

V. APPLICATION

As a model for computation, P systems offer the attractive possibility of solving NP-complete problems in less-than exponential time. Some P system variants are known to be capable of solving the SAT (Boolean Satisfiability) problem in linear time. As there is no current method of directly implementing a P system, their functionality is instead imitated and therefore solving NP-complete problems in linear time remains theoretical. Yet, it has also been proven that any deterministic P system may be simulated on a Turing Machine in polynomial time.

a. SAT and HPP

The Satisfiability of propositional formulas in the conjunctive normal form (SAT) problem and the Hamiltonian Path Problem (HPP) can be solved in a linear time using replicated rewriting P system. The time is estimated here as the number of steps the system works.

The SAT is probably the most known NP- complete problem. It asks whether or not for a given formula in the conjunctive normal form there is a truth-assignment of the variables for which the formula assumes the value true. A formula as above is of the form [14]:

$$\gamma = C_1 \wedge C_2 \wedge C_3 \wedge \ldots \wedge C_m,$$

Where each $C_j, 1 \leq j \leq m$, is a clause of the form of a disjunction

$$C_j = y_1 \vee y_2 \vee \ldots \vee y_r ,$$

With each yr being a propositional variable, xs, or its negation, $\neg\, xs$. Thus, we use the variables $x1, x2, \ldots$ and the three connectives ∨, ∧, ¬ : or, and, negation.

For example, let us consider the propositional formula:

$$\beta = (x_1 \vee x_2) \wedge (\neg\, x_1 \vee \neg\, x_2)$$

We have two variables x_1, x_2 and two clauses. It is easy to see that it is satisfiable: any of the following truth-assignments makes the formula true:

($x1$ = true, $x2$ = false), (1 = false, $x2$ = true).

The SAT problem can be solved by using replicated rewriting P system in a time linear in the number of variables and number of clauses.

The Hamiltonian Path Problem (HPP) can be solved by P systems with replicated rewriting in a time linear in the number of nodes of the given graph.

b. Solving SAT in linear time

The usefulness of P systems with 2- bounded division by solving the SAT problem in linear time. The time is estimated here as the number of steps the system works. This means, we have a parallel time where each unit is the time of a 'biological' step in the system, the time of using any rule, supposing that all rules take the same time to be applied.

The SAT problem can be solved by a P system with active membranes in a time which is linear in the number of variables and the number of clauses [12, 14].

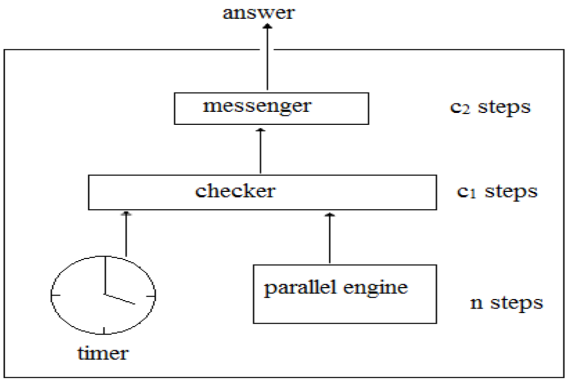


Figure 6: The shape of P systems solving NP- complete problems [10].

c. Solving NP-Complete

To solve NP-complete problems, this can be solved deterministically in linear time. The following step can be used:

1. Start with a linear membrane structure. The central membrane is always divided into an exponential number of copies.
2. In central "parallel engine" get, texture use of the membrane division, a "data pool" of exponential size. In parallel within this process, a "timer" is simultaneously ticking for synchronization.
3. After finishing the generation of the "data pool", one checks whether or not any solution exist.
4. A message is sent out of the system at a precise moment telling whether or not the problem has a solution.

d. L System

An L-system or Lindenmayer system is a parallel rewriting system. An L-system consists of an alphabet of symbols that can be used to make strings, an accumulation of production rules that expand each symbol into some larger string of symbols. Plant models and natural-looking organic forms are simple to define, as by increasing the recursion level the form slowly 'grows' and becomes more complex. It is popular in the generation of artificial life [15].

Example: - A variant of the Koch curve which uses only right angles.

Variable: F

Constants: + −

Start: F

Rules: (F → F+F−F−F+F)

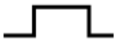
Here, F means "draw forward", + means "turn left 90°", and − means "turn right 90°" (Turtle graphical [13]).

n = 0:

F

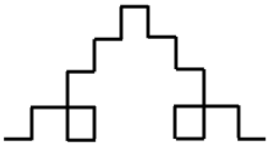
n = 1:

F+F−F−F+F



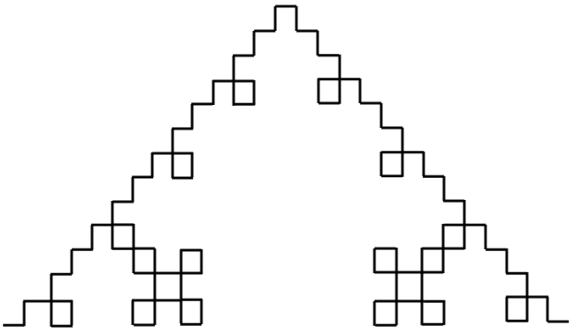
n = 2:

F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+
F+F−F−F+F



n = 3:

F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+F+F
−F−F+F+F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−
F+F+F+F−F−F+F−F+F−F−F+F+F+F−F−F+F−F+F−F−F+F
−F+F−F−F+F+F+F−F−F+F−F+F−F−F+F+F+F−F−F+F−F+
F−F−F+F−F+F−F−F+F+F+F−F−F+F+F+F−F−F+F+F+F−F
−F+F−F+F−F−F+F−F+F−F−F+F+F+F−F−F+F



Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. PROS AND CONS

From a theoretical point of view, it is a powerful branch of natural computing with a continuous afflux of new ideas, notions, problems, having a series of applications, especially in modeling biological phenomena. At the theoretical level, polynomial solutions to computationally hard (presumably intractable) problems can be devised in terms of P systems. Easy scalability, program-ability, and understand-ability are other attractive features of membrane computing as a modeling framework for the use of biology [16].

No lab implementation was intended, and no such implementation is known to be planned for the near future. There is no way to implement these solutions on the existing computers, sequential or with a limited parallelism as they are [16]. It is only deal with distributed and parallel computing. No “real” application in practical computer science was reported, at least not in the direction which is more plausible.

VII. CONCLUSION

Mention here some directions of research and some results on complexity classes and polynomial solutions to hard problems, dynamical systems approaches, population P systems (in general, systems dealing with populations of cells, as in tissue-like or neural-like systems) are of a strong current interest which will probably lead to significant theoretical and practical results. To these trends we can add another general, and not very structured yet, topic: using non-crisp mathematics, handling uncertainty by means of probabilistic, fuzzy sets, rough sets theories.

The computational power is of interest for theoretical computer science, computational efficiency is of interest for practical computer science, if a biologist is interested in simulating a cell and this seems to become a major concern of today biology and other sources then the generality of the model (its comparison with Turing machines and its restrictions) is directly linked to the possibility of solving algorithmic-ally questions about the model.

Membrane system is a computing model with the features of distribution, great parallelism and uncertainty. This P system is very useful to solve hard problems like NP-complete, SAT, HPP, which have exponential time complexity, in a linear time using their traditional algorithm. It will implement in real application of computer science to solve hard complexity problems linearly. Although it is not developed in computer science but there are many chance that it is being developed with emerging new technologies.

REFERENCES

1. Gheorghe Păun. "Introduction to membrane computing." Applications of Membrane Computing. Springer Berlin Heidelberg, 2006. 1-42.
2. Eberhart RC and Kennedy J, "A new optimizer using particle swarm theory," in Proceedings of the sixth international symposium on micro machine and human science, 1995, p. 39-43.
3. Yang Sun, Lingbo Zhang, and Xingsheng Gu. "Membrane computing based particle swarm optimization algorithm and its application." Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on. IEEE, 2010.
4. Jin-Hui Zhao, Ning Wang, and Ping Zhou. "Multiobjective bio-inspired algorithm based on membrane computing." Computer Science and Information Processing (CSIP), 2012 International Conference on. IEEE, 2012.
5. Gheorghe Marian and Florentin Ipate. "A Kernel P Systems Survey."Membrane Computing. Springer Berlin Heidelberg, 2014. 1-9.
6. Ana Pavel, Octavian Arsene, Catalin Buiu, “Enzymatic numerical P systems-a new class of membrane computing systems.” Bio-Inspired Computing: Theories and Applications (BIC-TA), Fifth International Conference on. IEEE, 2010.
7. Vincenzo Manca and Luca Bianco. "Biological networks in metabolic P systems." BioSystems 91.3 (2008): 489-498.
8. Francesco Bernardini and Marian Gheorghe. "Population P Systems." J. UCS10.5 (2004): 509-539.
9. Zhengwei Qi, Jinyuan You, Hongyan Mao, “P systems and Petri nets.” International Workshop, WMC 2003, Tarragona, Spain, July, 2003.
10. Gheorghe Păun, “Computing with Membranes”, Journal of Computer and System Sciences, volume 61, issue 1, (Science Direct), August 2000.(http://www.sciencedirect.com/science/article/pii/S0022000099916938).
11. Gheorghe Paun, "Membrane computing [and graph transformation]." Electronic Communications of the EASST 6 (2007).
12. Kamala Krithivasan. Introduction to formal languages, automata theory and computation. Pearson Education India, 2009.
13. Jie Xue and Xiyu Liu. "Applied membrane computation in creative design."Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on. IEEE, 2012.
14. Claudio Zandron, Claudio Ferretti, and Giancarlo Mauri. "Solving NP-complete problems using P systems with active membranes." Unconventional Models of Computation, UMC’2K. Springer London, 2001. 289-301.
15. (https://researchspace.auckland.ac.nz/bitstream/handle/2292/3611/102paun.pdf?___&origin=publication_detail)
16. Prusinkiewicz, Przemyslaw, and Aristid Lindenmayer. Graphical modeling using L-systems. Springer New York, 1990.(http://algorithmicbotany.org/papers/abop/abop-ch1.pdf)
17. Gheorghe Păun and Mario J. Pérez-Jiménez. “Membrane computing: Brief introduction, recent results and applications.” BioSystems, vol 85, issue 1, july 2006.
18. (http://www.sciencedirect.com/science/article/pii/S0303264706000219).

AUTHOR PROFILE

Kriti Bhargava received her B.Tech. Degree in Information Technology from Rajasthan Technical University and is currently pursuing her M.Tech. in Computer Science from Mody University of Science and Technology. Her areas of interest are public key cryptography and Digital Signatures.

Nisheeth Saxena is currently working as an Assistant Professor (Computer Science) in Mody University of Science and Technology. He has done M.Sc.(Math), MCA and M.Tech. in Computer Science. Currently he is pursuing his Ph.D. in MANET security. His area of interest is Cryptography and Network security. He is a life member of ISTE.