University of Bucharest
Faculty of Mathematics and Computer Science

Massively Parallel Computing Models Inspired from Biology
-Summary-

author: Stefan Popescu

supervisor: Prof. Dr. Victor Mitrana

Submitted in part fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science
October 2015

1 Introduction

Twenty years ago, Adleman's experiment proved that biology can be a source of inspiration for designing computational models, but it can also help by providing a new computer architecture, an alternative to traditional silicone based hardware, by successfully solving an instance of the Hamiltonian path problem using DNA bio-engineering tools. In the last two decades a new branch in computer science, generally named Natural Computing, has risen and become very dynamic.

The main difference between these bio-inspired models and traditional models is that the former's operations are inspired from living cells, operations that take place in a massive parallel manner. Bio-inspired computing has the potential to solve hard problems in a reasonable time, by taking advantage of the massive parallelism made possible by the way DNA stores and processes information at a molecular level.

2 Motivation

Networks of Evolutionary Processors (NEPs) as language generating devices and problem solvers have been considered in [6] and [18], respectively. They have been further investigated in a series of subsequent works. NEPs as accepting devices and problem solvers have been considered in [17]; later on, a characterization of the complexity classes **NP**, **P**, and **PSPACE** based on accepting NEPs has been reported in [14]. Universal NEPs and some complexity problems are discussed in [13]. The main results can be found in [16].

Software implementations of NEPs have been reported, see, e.g., [7, 8, 12], most of them in JAVA. They encountered difficulties especially in the implementation of filters. The main idea to simulate the non-deterministic behavior of NEPs has been to consider a safe-thread model of processors, that is to have each rule and filter in a thread, respectively. Clearly the threads corresponding to the filters are much more complicated than those associated with the evolutionary rules. Configuration changes in a NEP are accomplished either by a communication step or by an evolutionary step, but these two steps may be realized in any order. This suggests that evolution or communication may be chosen depending on the thread model of processor [8]. The input and output filters are implemented as threads extending the `Runnable` interface. Therefore a processor is the parent of a set of threads, which use all objects from that processor in a mutual exclusion region. When a processor starts to run, it starts in a cascade way the rule threads and filter threads. As one can see, the filters associated with processors, especially if there are both input and output filters, seem to be hardly implementable. Consequently, it would be of interest to replace the communication based on filters among processors by another protocol. A first attempt was to move filters from each node to the edges between the nodes, see, e.g., [9]. Although this variant seems to be theoretically simpler, the attempts towards an implementation have encountered similar difficulties due to the fact that the filters associated with edges are similar to those associated with nodes.

Work [1] considers a new variant of NEP with the aim of proposing a new type of filtering process and discusses the potential of this variant for solving hard computational problems. The main and completely new feature of this variant is the valuation mapping which assigns to each string an integer value, depending on the values assigned to its symbols. Actually, we are not interested in computing the exact value of a string, but just the sign of this value. By means of this valuation, one may metaphorically say that the strings are electrically polarized. Thus, if the nodes are polarized as well, the strings migration from one node to another through the channel between the two cells seems to be more natural and easier to be implemented.

Regarding picture languages, two variants of networks of evolutionary picture processors were considered in [4] and [5]. These studies include a 2-dimensional generalization for networks of evolutionary processors with filter based communication protocols and compare the resulting classes of picture languages to the class of picture languages that can be locally accepted by a tile set and the one that can be recognized by a tiling system. Although these models seem to be able to accept a wide variety of languages (i.e. the complement of any local language, some languages that are not tile recognizable), certain limitations come to mind, especially regarding the 2-dimensional pattern matching problem: the solution being presented only for the case where the pattern was made up of at most three columns or three rows.

3 Thesis Structure

This thesis is divided into 5 chapters as follows:

1. In the introductory chapter we present the motivation for our study.
2. In the chapter entitled "**Background Theory**" we define the basic notions and principles that shall be used throughout this thesis. This chapter is divided into 5 sections:
   - The first section contains the basic notations and definitions regarding strings.
   - In the second part we define the notion of Turing machines and give some examples of different models of Turing machines.
   - In the third section we briefly describe the notion of time complexity and define some important complexity classes.
   - In the fourth section we discuss the class of **NP**-complete languages
   - In the last section we describe the notion of tag systems - a universal deterministic computational model.
3. In the third chapter, **Networks of Polarized Evolutionary Processors**, we consider the computational power of a new variant of networks of evolutionary processors. Each processor as well as the data navigating throughout the network are now considered to be polarized. While the polarization of every processor is predefined, the data polarization is dynamically computed by means of a valuation mapping. Consequently, the protocol of communication is naturally defined by means of this polarization. This chapter is structured as follows:

- We start with some background information about bio-inspired computing in general, the motivation behind such research and we refer to other computational models based on evolutionary processors.
- In the second section we formally define the notion of a *polarized evolutionary processor* and of networks based on this processor type along with a specific communication protocol.
- In the third section we show that *Networks of Polarized Evolutionary Processors* are universal, meaning that all enumerable functions can be computed by these networks. Firstly we show that tag systems can be simulated by such a network with a constant number of nodes, while Turing machines can be simulated, in a time-efficient way, by these networks with a number of nodes depending linearly on the tape alphabet of the Turing machine [2]. If we want to simulate Turing machines with networks of constant size, this is possible with a considerable increase of the computation time [3]. Finally, we show that every network can be simulated by a Turing machine and discuss the time cost of this simulation [3].
- In the fourth section we try to improve the previous result. We simulate a Turing machine using a $NPEP$ with a constant number of nodes and that has the polarity of each symbol, which was previously defined to be an integer, restricted to the set $\{-1, 0, 1\}$ [19].
- In the last section we give an example by constructing a Network of Polarized Evolutionary Processors that solves the CNF-SAT problem.

4. In the fourth chapter, **2-Dimensional Computing**, we define a new bio-inspired computational model for deciding 2-dimensional languages similar to those presented in [4] and [5]. The novel factor being the communication protocol, which is based on the polarity associated with the symbols on the picture frame. This model can be viewed as a 2-dimensional extension of the one presented in [2]. This chapter is structured as follows:

    - In the first section we state our motivation for the generalization of the concept of evolutionary processors to operate on pictures.

- The second section contains the basic definitions for understanding picture languages along with some simple examples.
- In the third section we formally describe the class of languages that can be locally accepted by tile sets and the class of languages that can be recognized by tiling systems.
- In the last section we introduce a new model for computing two dimensional words. Namely, extend the concept of Networks of Polarized Evolutionary Processors described in Chapter 3 to include working on rectangular pictures. These *Networks of Polarized Evolutionary Picture Processors* have a modified communication protocol that associates to every picture a quadruple with elements from the set $\{-, 0, +\}$. We consider this quadruple as an 2-dimensional generalization of the basic string polarity previously described. We compare the class of languages that can be accepted by our model and the ones that can be locally accepted and the ones that are tile-recognizable, respectively. Finally we propose a partial solution to the 2D-pattern matching problem. We only consider the case where the pattern is a picture which either has at most three rows or three columns. This last section is mostly based on work done in [20].

5. In the last chapter, **Conclusions**, we present a short summary of the results given and we present a list of open problems.

4 Main Results

4.1 Background Information

Networks of Polarized Evolutionary Processors (NEPs) are defined as a graph, with every node containing a cell like processor, capable of basic evolutionary operations (i.e. insertions, deletions and substitutions). Processors placed in adjacent nodes can communicate with each-other according to a predefined protocol.

In this thesis we discuss a more recent variant of NEPs, namely that of Polarized Processors which was considered in [1]. This class of processors differs from the previously mentioned in works such as [16]. In defining the communication protocol we abandon the idea of filters and introduce the concept of node and word polarization.

4.2 Summary of Results

Although the communication protocol based on the polarized processors and the valuation function seems to offer less control, the new variant is still computationally complete. We show that NPEP with a constant number of processors, namely 15, are computationally complete by devising a method for simulating 2-Tag Systems.

Theorem 4.1. *For every* 2*-tag system* $T = (V, \mu)$ *there exists an NPEP* Γ *of size* 15 *such that* $L(\Gamma) = \{w \mid \ T \text{ halts on } w\}$.

As a 2-tag system can efficiently simulate any deterministic Turing machine but not nondeterministic ones, we propose a simulation of nondeterministic Turing machines with NPEPs which maintains the working time of the Turing machine.

Theorem 4.2. *For any recursively enumerable language* L*, accepted in* $\mathcal{O}(f(n))$ *time by a Turing machine with tape alphabet* U*, there exists an NPEP of size* $10 \cdot card(U)$ *accepting* L *in* $\mathcal{O}(f(n))$ *time.*

Unlike the simulation of a 2-tag system, the size of an NPEP simulating an arbitrary Turing machine depends linearly on the number of tape symbols of the Turing machine, but preserves the time complexity. If we want to simulate Turing machines with networks of constant size, this is possible with a considerable increase of the computation time.

Theorem 4.3. *For any recursively enumerable language* L*, accepted in* $\mathcal{O}(f(n))$ *time by a Turing machine with tape alphabet* U*, there exists an NPEP of size* 39 *accepting* L *in* $\mathcal{O}((f(n) \cdot card(U))^2)$ *time.*

We also show that every NPEP, with the input alphabet V and the valuation mapping φ, can be simulated by a Turing machine.

Theorem 4.4. *For every language L accepted by an NPEP, with the network alphabet V and the valuation mapping φ, in $\mathcal{O}(f(n))$ time, there exists a Turing machine that accepts L in $\mathcal{O}(f(n)(Kf(n)+Kn))$ time, where $K = \max\{|\varphi(a)| \mid a \in V\}$.*

We then consider an open problem given in [3]. We construct with a constant number of nodes that directly simulates a Turing machine, but this time we only use symbols with an associated polarity from the set $\{-1, 0, 1\}$ but at a computational time cost.

Theorem 4.5. *For any recursively enumerable language L, accepted by a Turing machine with tape alphabet U, there exists an NPEP of size* 35*, with a polarization of symbols restricted to the set $\{-1, 0, 1\}$, that can accept L.*

In chapter 4 we compare the classes of 2-dimensional languages $L(LOC)$ and $L(REC)$ to the class of languages that can be decided by *Accepting Networks of Polarized Picture Processors* [20]. Firstly we construct an ANPEP that can accept only square pictures over a single letter alphabet - this set of pictures being recognizable by a tile system but not by a tile set.

Theorem 4.6. $L(ANPEPP) \cap (L(REC) \backslash L(LOC)) \neq \emptyset$.

We then show that ANPEPPs can accept some languages that are not in $L(REC)$ - namely the set of pictures with an even number of rows, that have the two middle rows identical.

Theorem 4.7. $L(ANPEPP) \backslash L(REC) \neq \emptyset$.

Our next result regards the complement of any local language.

Theorem 4.8. *The complement of any local language can be accepted by an $ANPEPP$.*

Finally we give a partial solution to the pattern matching problem.

Theorem 4.9. *Given a picture π with $l_1(\pi) \leq 3$ or $l_2(\pi) \leq 3$ and $L = \{\pi' \mid \pi$ is a sub-picture of $\pi'\}$, then $L \in L(ANPEPP)$.*

References

[1] P. Alarcón, F. Arroyo, V. Mitrana
`Networks of polarized evolutionary processors as problem solvers`. In *Advances in Knowledge-Based and Intelligent Information and Engineering Systems, Frontiers in Artificial Intelligence and Applications*, IOS Press, (2012), pg 807–815.

[2] F. Arroyo, S. G. Canaval, V. Mitrana , Şt. Popescu
`Networks of Polarized Evolutionary Processors Are Computationally Complete`. In *Language and Automata Theory and Applications*, Lecture Notes in Computer Science 8370, Springer-Verlag, Berlin, (2014), pg 101–112.

[3] F. Arroyo, S. G. Canaval, V. Mitrana, Şt. Popescu
`On the Computational Power of Networks of Polarized Evolutionary Processors`. Accepted for publication at Information and Computation.

[4] P. Bottoni, A. Labella, V. Mitrana
`Accepting Networks of Evolutionary Picture Processors`. *Journal Fundamenta Informaticae - Formal Models-Computability, Complexity, Applications, Volume 131 Issue 3-4* (July 2014), pg 337–349.

[5] P. Bottoni, A. Labella, F. Manea, V. Mitrana, J. Sempere
`Networks of Evolutionary Picture Processors with Filtered Connections`. In *UC '09 Proceedings of the 8th International Conference on Unconventional Computation* (2009), pg 70–84.

[6] J. Castellanos, C. Martín-Vide, V. Mitrana, J. M. Sempere
`Networks of evolutionary processors`. In *Acta Inf.* 39 (2003), pg 517–529.

[7] M.A. Diaz, L.F. de Mingo, N. Gómez Blas, J. Castellanos
`Implementation of massive parallel networks of evolutionary processors (MPNEP): 3-colorability problem`. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, Studies in Computational Intelligence 129, Springer, (2008), pg 399–408.

[8] M.A. Diaz, L.F. de Mingo, N. Gómez
`Networks of evolutionary processors: Java Implementation of a threaded processor`. In *International Journal of Information Theories & Applications* 15(2008) pg 37–43.

[9] C. Drăgoi, F. Manea, V. Mitrana
`Accepting networks of evolutionary processors with filtered connections.` *J. UCS* 13 (2007), pg 1598–1614.

[10] D. Giammarresi, A. Restivo,
`Recognizable picture languages.` In *Int. Journal Pattern Recognition and Artificial Intelligence.* Vol. 6, No. 2 & 3 (1992), pg 241-–256.

[11] D. Giammarresi, A. Restivo,
`Two-dimensional languages` , in [21], pg 215-267.

[12] N. Gómez, L.F. de Mingo, J. Castellanos
`Networks of evolutionary processors with a self-organizing learning.` In *Proc. ACS International Conference on Computer Systems and Applications*, IEEE Computer Society Washington, DC, USA, (2008), pg 917–918.

[13] F. Manea, C. Martín-Vide, V. Mitrana
`On the size complexity of universal accepting hybrid networks of evolutionary processors.` In *Mathematical Structures in Computer Science* 17 (2007), pg 753–771.

[14] F. Manea, M. Margenstern, V. Mitrana, M. J. Pérez-Jiménez
`A new characterization of NP, P, and PSPACE with accepting hybrid networks of evolutionary processors`. In *Theory Comput. Syst.* 46 (2010), pg 174–192.

[15] F. Manea, C. Martín-Vide, V. Mitrana
`Accepting networks of splicing processors.` In *New Computational Paradigms,* Springer Berlin Heidelberg, (2005) pg 300–309.

[16] F. Manea, C. Martín-Vide, V. Mitrana
`Accepting networks of evolutionary word and picture processors: A survey.` In *Scientific Applications of Language Methods, Mathematics, Computing, Language, and Life: Frontiers in Mathematical Linguistics and Language Theory - Vol. 2*, World Scientific (2010), pg 523–560.

[17] M. Margenstern, V. Mitrana, M. J. Pérez-Jiménez
`Accepting Hybrid Networks of Evolutionary Processors.` In *Proc. 10th International Workshop on DNA Computing, DNA 10*, Lecture Notes in Computer Science 3384, Springer-Verlag, Berlin (2004), pg 235–246.

[18] C. Martín-Vide, V. Mitrana, M. J. Pérez-Jiménez, F. Sancho-Caparrini
`Hybrid networks of evolutionary processors.` In *Proc. Genetic and Evolutionary Computation - GECCO 2003*, Lecture Notes in Computer Science 2723 Springer-Verlag, Berlin (2003), pg 401–412.

[19] Şt. Popescu
`Networks of Polarized Evolutionary Processors with Elementary Polarization of Symbols`, submitted.

[20] Şt. Popescu
`Networks of Polarized Evolutionary Picture Processors`. Accepted to appear in *Romanian Journal of Information Science and Technology.* Vol. 18, No. 1 (2015).

[21] G. Rozenberg, A. Salomaa (eds.)
Handbook of Formal Languages, vol. I-III. Springer-Verlag, Berlin (1997).