EXTENDED NETWORKS OF EVOLUTIONARY PROCESSORS

Luis Fernando de Mingo, Nuria Gómez Blas, Francisco Gisbert, Miguel A. Peña

Abstract: This paper presents an extended behavior of networks of evolutionary processors. Usually, such nets are able to solve NP-complete problems working with symbolic information. Information can evolve applying rules and can be communicated though the net provided some constraints are verified. These nets are based on biological behavior of membrane systems, but transformed into a suitable computational model. Only symbolic information is communicated. This paper proposes to communicate evolution rules as well as symbolic information. This idea arises from the DNA structure in living cells, such DNA codes information and operations and it can be sent to other cells. Extended nets could be considered as a superset of networks of evolutionary processors since permitting and forbidden constraints can be written in order to deny rules communication.

Keywords: Networks of Evolutionary Processors, Membrane Systems, and Natural Computation.

ACM Classification Keywords: F.1.2 Modes of Computation, I.6.1 Simulation Theory, H.1.1 Systems and Information Theory

Conference: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008

Introduction

Natural sciences, and especially biology, represent a rich source of modeling paradigms. Well-defined areas of artificial intelligence (genetic algorithms, neural networks), mathematics, and theoretical computer science (L systems, DNA computing) are massively influenced by the behavior of various biological entities and phenomena. In the last decades or so, new emerging fields of so-called "natural computing" [1,2,3] identify new (unconventional) computational paradigms in different forms. There are attempts to define and investigate new mathematical or theoretical models inspired by nature [8], as well as investigations into defining programming paradigms that implement computational approaches suggested by biochemical phenomena. Especially since Adleman's experiment [4] these investigations received a new perspective. One hopes that global system-level behavior may be translated into interactions of a myriad of components with simple behavior and limited computing and communication capabilities that are able to express and solve, via various optimizations, complex problems otherwise hard to approach.

The origin of networks of evolutionary processors is a basic architecture for parallel and distributed symbolic processing, related to the Connection Machine [7] as well as the Logic Flow paradigm [5], which consists of several processors, each of them being placed in a node of a virtual complete graph, which are able to handle data associated with the respective node. All the nodes send simultaneously their data and the receiving nodes handle also simultaneously all the arriving messages, according to some strategies, see, e.g., [6,7].

Networks of evolutionary processors (NEP) [9,11] are language-generating device, if we look at the strings collected in the output node. We can also look at them as doing some computation. If we consider these networks with nodes having filters defined by random context conditions, which seems to be closer to the recent possibilities of biological implementation, then using these simple mechanisms we can solve NP-complete problems in linear time. Such solutions are presented for the *Bounded Post Correspondence Problem* in [10] for the *3-Colorability Problem* in [9] and for the {\it Common Algorithmic Problem} in [12] As a further step, in [12] the so-called hybrid networks of evolutionary processors are considered. Here deletion node or insertion node has its own working mode (performs the operation at any position, in the left-hand end or in the right-hand end of the word) and different nodes are allowed to use different ways of filtering. Thus, the same network may have nodes where the deletion operation can be performed at arbitrary position and nodes where the deletion can be done only at right-end of the word.

In this paper, we present some results regarding a network of evolutionary processor based on an extended behavior from the biological point of view. This is a preliminary work in which rules pass through the net, they are not associated to a fix processor.

Networks of Evolutionary Processors

Connectionist models consists of several processors which are located in the nodes of a virtual graph and are able to perform operations in that node, according to some predefined rules. Information is passed through connections in order to obtain a collaborative solution to a given problem. All processors work in a parallel way and they only perform simple operations.

A network of evolutionary processors is a tuple

$$\Gamma = (V, U, G, \mathcal{N}, \alpha, x_I, x_O)$$

- V, U are the net alphabet and input alphabet respectively.
- G is an undirected graph in which each node is a processor. Processors have a set of objects/strings and a set of evolution rules.
- N is a mapping that associates each processor with a set of filters.
- α is a mapping that defines the behavior of filters (weak or strong conditions).
- x_I, x_O are the input and output processors.

Objects in processors can evolve and communicated to other connected processors. That is, rules can be applied (evolution) or objects can pass filter conditions (communication). These two steps could be sequential (evolution and then communication) or parallel (evolution and communication at same time).

Evolution

A given string *x* can evolve provided there is some rule to apply it. Taking into account that there are an arbitrary large number of copies of string *x* in processor, several rules can be applied in parallel to different copies in one evolutionary step.

Therefore, the set of objects in a processor *i* after an evolution step, denoted by L'_i, are those before the evolution (L_i) adding objects obtained after rules in *i* are applied. That is,

$$L'_i = L_i \cup r_k(x),\ x \in A_i, r_k \in R_i$$

Communication

A given string/object *x* can pass filters in processor *i*, iff the following constraint is satisfied:

$$\varphi_i^{(\beta)}(x; \mathcal{N}(i)), \beta = \{s, w\}$$

Where, $\mathcal{N}(i)$ is the filter set associated to a given processor *i*. This set can contain just input and output filters, or forbidden context filters as well. That is, $\mathcal{N}(i) = \{PI, PO\}$, or $\mathcal{N}(i) = \{FI, PI, FO, PO\}$. Constraints are defined as follows:

- Constraints conditions with permitting filters (*PI, PO*); where *P* is either input filter or output filter, it depends if the string is sent out or received with strong conditions (s) or weak condition (w):

$$\varphi_i^{(s)}(x; \mathcal{N}(i)) \equiv P \subseteq alph(x)$$
$$\varphi_i^{(w)}(x; \mathcal{N}(i)) \equiv P \cap alph(x) \neq \emptyset$$

- Constraints conditions with permitting filters (*PI, PO*) and forbidden context (*FI, FO*); where *P, F* is either input filter or output filter, it depends if the string is sent out or received with strong conditions (s) or weak conditions (w):

$$\varphi_i^{(s)}(x;\mathcal{N}(i)) \equiv P \subseteq alph(x) \wedge F \cap alph(x) = \emptyset$$
$$\varphi_i^{(w)}(x;\mathcal{N}(i)) \equiv P \cap alph(x) \neq \emptyset \wedge F \cap alph(x) = \emptyset$$

Therefore, the set of objects in a processor *i* after a communication step, denoted by *L'ᵢ*, are those before the communication (L_i) removing objects sent out and adding objects from other processors connected to *i*. That is,

$$L_i' = L_i - \{w|\varphi^k(x;\mathcal{N}(N_i))\} \bigcup_{\{N_i,N_j\}\in E} \{x|\varphi^k(x;\mathcal{N}(N_j)) \wedge \varphi^k(x;\mathcal{N}(N_i))\}$$

The most important result of such networks of evolutionary processors is that they can solve NP-complete problems in linear time and linear resources.

Extended Networks of Evolutionary Processors

The communication step is only applied to objects in traditional nets. An extended version is proposed in order to be able to send rules from one processor to other ones. This property provides a more realistic behavior since operations are not fixed in processors, they can pass through the net in the same way objects do.

A rule can pass filter conditions provided:

$$\varphi_i^{(\beta)}(r_j : x \to y;\mathcal{N}(i)) \equiv \varphi_i^{(\beta)}(x;\mathcal{N}(i)) \wedge \varphi_i^{(\beta)}(y;\mathcal{N}(i))$$

That is, given rule r_j belonging to processor *i* can be sent out if both antecedent and consequent strings can pass filters in processor *i*, and can be received by other processors if the rule pass their input filters (weak or strong conditions).

A network of evolutionary processors can be transformed into an equivalent extended network of evolutionary processors just choosing the right filters. For example, antecent belonging to rules can be added to forbidden filter in order to avoid rules communication.

Following theorems regarding computational power of non-extended networks of evolutionary processors can be also applied to extended networks of evolutionary processors.

Theorem 1. A complete NEP of size 5 can generate each recursively enumerable language. [9]

Theorem 2. A star NEP of size 5 can generate each recursively enumerable language. [9]

Theorem 3. The bounded PCP can be solved by an NEP in size and time linearly bounded by the product of K and the length of the longest string of the two Post lists. [12]

Theorem 4. The families of regular and context-free languages are incomparable with the family of languages generated by simple NEPs. [10]

Theorem 5. The 3-colorability problem can be solved in O(m + n) time by a complete simple NEP of size 7m+2, where n is the number of vertices and m is the number of edges of the input graph. [10]

Conclusions and Future Work

This paper presents an extended behavior in networks of evolutionary processors. Now, rules can pass from one processor to another one provided filter constraints are satisfied. This mechanism allows operations and data to pass through the net, not only data like in networks of evolutionary processors. This idea tries to model DNA behavior in which information combines data and operations for living cells, the information is a whole, does not matther if it is data or operations. Rules can pass filters as well as objects do, according to filter specifications.

It is clear that this model is a superset of networks of evolutionary processor and therefore it can solve NP-complete problems in linear time and linear resources. Main advantage of such extended model is that the time to solve a problem is lower than non-extended models since rules can travel to transform objects at different processors.

There are some other possibilities when defining conditions of rules in order to pass filters,

$$\varphi_i^{(\beta)}(r_j : x \to y; \mathcal{N}(i)) \equiv \varphi_i^{(\beta)}(x \cup y; \mathcal{N}(i))$$
$$\varphi_i^{(\beta)}(r_j : x \to y; \mathcal{N}(i)) \equiv \varphi_i^{(\beta)}(x \cap y; \mathcal{N}(i))$$
$$\varphi_i^{(\beta)}(r_j : x \to y; \mathcal{N}(i)) \equiv \varphi_i^{(\beta)}(x \setminus y; \mathcal{N}(i))$$

A lot of open problems that can be taken into account to probe computational power of extended networks of evolutionary processors. First step will consist on solving same problems than non-extended net in order to compare time and resources.

Bibliography

[1] Zandron, C. (2002). A Model for Molecular Computing: Membrane Systems, Universita degli Studi di Milano, Italy.

[2] Paun, G. (2002). Membrane Computing. An Introduction. Springer-Verlag, Berlin.

[3] Ciobanu, G., Paun, G., and Perez-Jimenez, M. (2005). Applications of Membrane Computing. Springer-Verlag, Berlin.

[4] Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. Science, 226:1021–1024.

[5] Errico, L. and Jesshope, C. (1994). Towards a new architecture for symbolic processing. Artificial Intelligence and Information Control Systems of Robots, pages 31–40.

[6] Fahlman, S., Hinton, G., and Seijnowski, T. (1983). Massively parallel architectures for ai: Massively parallel architectures for ai: Netl, thistle and boltzmann machines. In AAAI National Conference on Artificial Intelligence, pages 109–113.

[7] Hillis, W. (1985). The Connection Machine. MIT Press, Cambridge.

[8] Robinson, D. A. (1992). Implications of neural networks for how we think about brain function. Behavioral and Brain Sciences, 15(4): 644–655.

[9] J. Castellanos, C. Martın-Vide, V. Mitrana, and J. Sempere (2003). Networks of evolutionary processors. Acta Informatica, 39:517–529,

[10] J. Castellanos, C. Martın-Vide, V. Mitrana, and J. Sempere (2001). Solving np-complete problems with networks of evolutionary processors. Lecture Notes in Computer Science, 2084:621–628.

[11] E. C. Varju and V. Mitrana (2000). Evolutionary systems, a language generating device inpired by evolving communities of cells. Acta Informatica, 36:913–926,

[12] C. Martin-Vide, V. Mitrana, M. Perez-Jimenez, and F. S. Caparrini (2003). Hybrid networks of evolutionary processors. Lecture Notes in Computer Science, 2723:401–412.

Authors' Information

Luis Fernando de Mingo López – Dept. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain;
e-mail: lfmingo@eui.upm.es

Nuria Gómez Blas – Dept. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain;
e-mail: ngomez@dalum.eui.upm.es

Francisco Gisbert – Dept. Lenguajes, Sistemas Informáticos e Ingeniería del Software, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain; e-mail: fgisbert@fi.upm.es

Miguel Angel Peña – Dept. Lenguajes, Sistemas Informáticos e Ingeniería del Software, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain;
e-mail: fgisbert@fi.upm.es