Chapter IV

Networks of Evolutionary Processors: Results and Perspectives

Carlos Martín-Vide, Rovira i Virgili University, Spain

Victor Mitrana, University of Bucharest, Romania, and Rovira i Virgili University, Spain

ABSTRACT

The goal of this chapter is to survey, in a systematic and uniform way, the main results regarding different computational aspects of hybrid networks of evolutionary processors viewed both as generating and accepting devices, as well as solving problems with these mechanisms. We first show that generating hybrid networks of evolutionary processors are computationally complete. The same computational power is reached by accepting hybrid networks of evolutionary processors. Then, we define a computational complexity class of accepting these networks and prove that this class equals the traditional class \(NP\). In another section, we present a few \(NP\)-complete problems and recall how they can be solved in linear time by accepting networks of evolutionary processors with linearly bounded resources (nodes, rules, symbols). Finally, we discuss some possible directions for further research.
INTRODUCTION

A rather well-known architecture for parallel and distributed symbolic processing related to the Connection Machine (Hillis, 1985) and the logic flow paradigm (Errico & Jesshope, 1994) consists of several processors, each of them being placed in a node of a virtual complete graph, which is able to handle data associated with the respective node. Each node processor acts on the local data in accordance with some predefined rules, and then local data becomes a mobile agent that can navigate in the network following a given protocol. Only those data that can pass a filtering process can be communicated to the other processors. This filtering process may require satisfying some conditions imposed by the sending processor, by the receiving processor, or by both of them. All the nodes send their data simultaneously, and the receiving nodes also simultaneously handle all of the arriving messages, according to some strategies (see, e.g., Fahlman et al., 1983; Hillis, 1985).

Starting from the premise that data can be given in the form of words, Csuhaj-Varjú and Salomaa (1997) have introduced a concept called networks of parallel language processors with the aim of investigating this concept in terms of formal grammars and languages. Networks of parallel language processors are closely related to grammar systems (Csuhaj-Varjú et al., 1993), and more specifically to parallel communicating grammar systems (Păun & Sântean, 1989). The main idea is that one can place a language-generating device (grammar, Lindenmayer system, etc.) in each node of an underlying graph. Each device rewrites the words existing in the corresponding node, and the words are then communicated to the other nodes. Words can be successfully communicated if they pass some output and input filters. More recently, Csuhaj-Varjú and Salomaa (2003) have introduced networks whose nodes are standard Watson-Crick D0L systems, which communicate to each other either the correct or the corrected words.

In Castellanos et al. (2001), this concept was modified in a way inspired from cell biology. Each processor placed in the nodes of the network is a very simple processor, an evolutionary processor. By an evolutionary processor, we mean a processor that is able to perform very simple operations, namely formal language theoretic operations, that mimic the point mutations in a DNA sequence (insertion, deletion or substitution of a pair of nucleotides). More generally, each node may be viewed as a cell having genetic information encoded in DNA sequences that may evolve by local evolutionary events, namely point mutations. Each node is specialized just for one of these evolutionary operations. Furthermore, the data in each node is organized in the
Artificial Organs
[www.igi-global.com/article/artificial-organs/104503?camid=4v1a](www.igi-global.com/article/artificial-organs/104503?camid=4v1a)

Organization-Oriented Chemical Programming of Distributed Artifacts
[www.igi-global.com/article/organization-oriented-chemical-programming-distributed/40362?camid=4v1a](www.igi-global.com/article/organization-oriented-chemical-programming-distributed/40362?camid=4v1a)