In this research, a new model of computing is presented, within the framework of natural computing and computer models inspired by biology: Networks of Genetic Processors (NGP). This new model is based on the one hand, by the family of networks of bio-inspired processors, specifically Networks of Evolutionary Processors (NEP) and Networks of Splicing Processors (NSP), and on the other hand by Genetic Algorithms. We can define the new model as a network of biologically-inspired processors where operations used by processors are crossover and mutation. One of the major interests studying the NGP is the computational power of the operations of crossover and mutation acting together. The NEP is a complete model that uses operations of symbol mutation: insertion, substitution and deletion, the NSP is a complete model that uses splicing operations, but the NEP is no longer a complete model using only substitution operations, as it happens to the NSP if we restrict the context of its splicing rules to the empty context. The study of the new model presented here responds to what happens when we put together in a single model the substitution rules from the NEP (called mutation rules) and the splicing rules with empty context from the NSP (called crossover rules).

When we work with networks of biologically-inspired processors there are two basic types of networks, accepting networks and generating networks. These types of networks are mainly used to work at a theoretical level or to solve decision problems. To work on a more practical level such as solving optimization problems, we propose a new type of network, Networks of Genetic processors as Parallel Genetic Algorithms, inspired by the Parallel Genetic Algorithms.

In this work we prove the computational completeness of our new model by showing that it is equivalent to the Turing machine. We prove the computational completeness of Parallel Genetic Algorithms by using this result and the similarity between the NGP and Parallel Genetic Algorithms. Moreover, we propose a characterization of the Chomsky hierarchy by using the NGP. Here, we simulate every grammar in the language classes of the Chomsky's hierarchy by using a NGP with an small number of processors required for each simulation. Hence, it gives an appreciable idea of the descriptional complexity of the different families of languages.

Finally, in this work there is an experimental study of the behavior of the model for the resolution of some practical problems. First, we design and implement a simulator that allows the execution of networks of Genetic Processors in any of its three defined types: accepting networks, generating networks or as Parallel Genetic Algorithms. This allows us to test the model with different optimization problems. Then, we make a study to see if the new model could solve NP problems in polynomial time. We use the decision problem of Hamiltonian cycle in a graph. Finally, we test the simulator with two optimization problems showing a good computational behavior. The problems are the Multidimensional Knapsack problem and the Traveling Salesman problem.