Complexity Engineering, Evolution and Optimality of Structures

Sourav Kundu

Center for Applied Evolutionary Intelligence (CAEI), Department of Computer Engineering, Atilim University, Kizilkasar Koyu, 06836 Incek, Golbasi, Ankara,Turkey. Email: sourav@caei-turkey.org

Telephone: +90 (312) 499 13 96 Fax: +90 (312) 499 14 34

Abstract

Complexity Engineering deals with harnessing the power of Cellular Automata (CA) like simple models to solve real life difficult and complex engineering problems, dealing with systems that have very simple components that collectively exhibit complex behaviors. The state transition rules are generally fixed in a CA, but evolution of the rules, is considered here. To evolve the *local rules*, we use the Genetic Algorithm (GA) model. The example application described here will serve to demonstrate that the GAs *can* discover CA rules that give rise to emergent computational strategies by self-organization and exhibit *globally* coordinated tasks for optimization, with simple *local* interactions only.

1 OVERVIEW

Discrete dynamical systems that follow the second law of thermodynamics evolve with time to maximal entropy and complete disorder. Cellular Automata (CA) are examples of such dynamical systems which, on the contrary exhibit *self organizing* behavior with increasing time. CAs are useful in modeling modular systems. An important aspect of modularity in engineering systems is the abstraction it makes possible. Once the construction of a particular module has been completed, the module can be treated as a single object and only its behavior need be considered wherever the module appears. One such application of modularity is described, where a structural plate is considered as composed of smaller structural modules which are cells in a lattice of sites in a CA and have discrete values updated in discrete time steps according to *local rules*. Furthermore, an evolutionary CA model tries to find local rules which can minimize the weight of such a plate structure. The plate is subjected to an external load which may be distributed or at a point. It is 50mm X 50mm square with fixed left edge and loaded on the right edge. This plate is divided into 25 unit square elements with a discrete set of plate thicknesses defined for each element. A CA encodes the configuration of this plate, with the state of its cells representing the discrete set of thicknesses, changing from one to any other as a result of its interaction with neighboring elements (governed by *local rules* of the CA). These rules are subjected to evolutionary improvement. They are encoded in the GA chromosome strings, which evolve to give a rich set of rules that can take any starting random initial configuration (IC) to a desired final configuration. Figure 1 shows the overall system architecture. Rule encoding is very similar to (Mitchell, 1996). Each GA chromosome represents a candidate rule table of the CA. Unit length neighborhood of cells in four cardinal directions is considered, which affect the state of each cell. Considering the cell itself, there are five cells having 2 state each, making it 32 states in all (2^5=32), that is 32 basic rules in all.

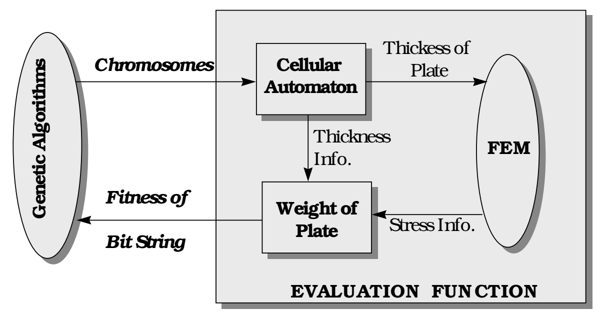

Figure 1: System Architecture

2 RESULTS

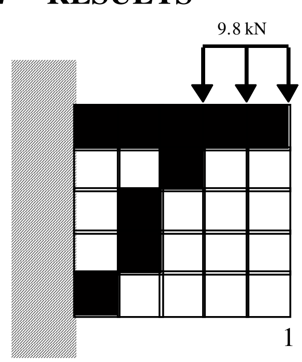

Figure 2 shows one of the results obtained for a 25 by 25 elements plate. For larger plate sizes, CA based optimization gives better performance than direct GA application. Here; no. of rules = 40, GA population = 15, crossover rate = 40%, mutation rate = 1 % and generation = 66^{th} .

Figure 2: One final configuration obtained.

References

Mitchell, M.; Crutchfield, J. P. and Das, R., (1996). Evolving cellular automata with genetic algorithms: A review of recent works. Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96).