Chapter XII
Parallelizing Genetic Algorithms:
A Case Study

Iker Gondra
St. Francis Xavier University, Canada

ABSTRACT

Genetic Algorithms (GA), which are based on the idea of optimizing by simulating the natural processes of evolution, have proven successful in solving complex problems that are not easily solved through conventional methods. This chapter introduces their major steps, operators, theoretical foundations, and problems. A parallel GA is an extension of the classical GA that takes advantage of a GA's inherent parallelism to improve its time performance and reduce the likelihood of premature convergence. An overview of different models for parallelizing GAs is presented along with a discussion of their main advantages and disadvantages. A case study: A parallel GA for finding Ramsey Numbers is then presented. According to Ramsey Theory, a sufficiently large system (no matter how random) will always contain highly organized subsystems. The role of Ramsey numbers is to quantify some of these existential theorems. Finding Ramsey numbers has proven to be a very difficult task that has led researchers to experiment with different methods of accomplishing this task. The objective of the case study is both to illustrate the typical process of GA development and to verify the superior performance of parallel GAs in solving some of the problems (e.g., premature convergence) of traditional GAs.

INTRODUCTION

Evolutionary Algorithms

Nature has always served as a source of inspiration in engineering and science. When looking for the best problem solvers known in nature, two candidates stand out: the human brain and the evolutionary mechanism that created the human brain. Attempting to design artificial models of those two problem solvers leads to the fields of neurocomputing and evolutionary computing, respectively. The fundamental metaphor of evolutionary algorithms, or evolutionary computing, relates natural evolution to problem solving in a trial-and-error approach (Eiben & Smith, 2003). In natural evolution, a population of individuals exists in an environment with limited resources. Competition for those resources results in the selection of fitter individuals (i.e., individuals that are better adapted to the environment). Then, those individuals act as seeds for the new generation of individuals through the processes of recombination and mutation. The fitness of the new individuals is evaluated and they compete (possibly also with their parents) for survival. Over time, this natural selection process results in a rise in the fitness of the population (Eiben & Smith, 2003).

There are many different variations of evolutionary algorithms but the underlying idea is the same: given a population of individuals, environmental pressure results in natural selection, which results in a rise in the overall fitness of the entire population. It is easy to think of such process as optimization. That is, given an objective function that is to be maximized, an initial population of randomly generated solutions is obtained. Then, a fitness (evaluation) function that represents the requirements that the population should adapt to is used to assign a single real-valued fitness to each individual in the population. The selection mechanism, which is based on the fitness value, is stochastic. That is, the probability that a particular individual is selected to act as a seed for the next generation is based on the individual's fitness. Thus, high-quality solutions are more likely to be selected than low-quality solutions but this is not guaranteed and even the worst solution in the population usually has a nonzero probability of being selected. This stochastic nature can aid in escaping from local optima. Recombination is a binary variation operator that merges information from parents (i.e., selected solutions) into offspring (i.e., new solutions). The choice of what information to merge is also usually stochastic. The hope is that, by combining good solutions, better solutions may be obtained. This principle has been used for millennia by breeders of plants and livestock. Mutation is a unary variation operator that randomly modifies one solution to deliver another. Thus, recombination allows us to perform exploitation by optimizing promising areas of the search space whereas, by creating random small diversions, mutation is explorative. The variation operators (i.e., recombination and mutation) generate the necessary diversity whereas selection acts as a force pushing quality. It is this combination of variation and selection that generally leads to improvements in the overall fitness value of successive generations. The whole field of evolutionary computing, or evolutionary algorithms, includes evolutionary programming, evolution strategies, genetic algorithms, and genetic programming as subareas (Eiben & Smith, 2003).

Genetic Algorithms

Genetic algorithms (GA) are adaptive methods that can be used to solve search and optimization problems. They are based on the mechanics of natural selection and genetic processes of living organisms. From one generation to another, populations evolve according to the principles of natural selection and the survival of the fittest individuals (Darwin, 1859). By imitation of the natural process, GAs are capable of developing solutions to real problems.

The basic principles of GAs were established by Holland (1975). Holland's insight was to be able to represent the fundamental biological mechanisms that permit system adaptation into an abstract form that could be simulated on a computer for a wide range of problems. He introduced bit strings to represent feasible solutions (or individuals) in some problem space. GAs are analogous to the natural behavior of living organisms. Individuals in a population compete for resources. Those individuals that are better adapted survive and have a higher probability of mating and generating descendants; therefore, the genes of stronger individuals will increase in successive generations.

A GA works with a population of individuals, each representing a feasible solution to a given problem. During each iteration step, called a generation, the individuals in the current population

Related Content