Introduction to the Special Issue: Scalable Evolutionary Computation

Hillol Kargupta
School of Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752, USA
hillol@eecs.wsu.edu

Scalability is an important issue in algorithm design that studies the characteristics of an algorithm with respect to the changes in the factors that influence its performance. Such factors typically include the "size" of the problem, different facets of problem-difficulties, desired accuracy and reliability of the algorithm performance, and the model of computation. It is important to a researcher because it characterizes the performance of an algorithm and identifies the suitability of the algorithm for solving large, difficult problems using different computing environments. It is also important to a practitioner because it points out the cost of running an algorithm for the expected level of success. Therefore, it makes sense to study scalability on both theoretical and experimental grounds.

From the theoretical perspective, *computational complexity* analysis offers a useful way to quantify the scalability of an algorithm. This perspective of scalability is a widely accepted norm in almost every matured discipline that deals with computation. However, the presence of computational complexity-based analysis in the Evolutionary Computation (EC) literature has been minimal in the past. This could be because enumerative search takes exponential time at the worst case for almost every interesting application domain.

We know that evolutionary search, optimization, and machine learning are hard, in general. However, such worst case results should not prohibit us from quantifying the performance of evolutionary algorithms (EAs) for different classes of problems, performance qualities, and models of computation. We need to study the scalability of EAs for different scenarios and compare their performance with that of other existing algorithms.

The scalability issue in EC shows up in many different facets. Typically, EAs are applied to search, optimization, and machine learning problems. The factors controlling the difficulty level of such problems include the number of search variables, epistatic interaction among the search variables, local optima, misleadingness, and other structural properties of the search space. Scalability of EAs with respect to growing degrees of difficulty along these dimensions is an important issue. We need to quantify the performance characteristics of our favorite EAs as we increase our demand for better reliability and accuracy. Moreover, the speedup in problem solving by EAs with increasing number of processors in different models of computation needs to be quantified. We should also explore the practitioner's perspective toward scalability. We need to study successful, large-scale EC applications and identify the main issues behind their scalability. Undoubtedly, this will play a critical role in the future of EC for the next millennium.

Fortunately, an increasing number of EC researchers and practitioners are investigating the scalability of popular evolutionary algorithms. They are developing EAs that can be rigorously claimed to be efficient on both theoretical and experimental grounds. This special issue offers a flavor of this growing body of work by putting together several interesting papers that address different facets of scalability in EAs.

The first paper by Dirk Thierens discusses the scalability of simple genetic algorithms (GAs). Typically in simple GAs, the search space structure is supposed to be implicitly detected using schemata (similarity-based equivalence classes). This paper considers the scalability of the process of schema detection and its subsequent exploitation in simple GAs. It essentially shows that, unless the user hand-picks the representation in such a way that the information about the features defining good schemata are available to the crossover operator, simple GAs face some serious scalability problems.

The second paper by Heinz Mühlenbein and Thilo Mahnig describes the Factorized Distribution Algorithm (FDA) that estimates the underlying search space distribution using an a-priori chosen structure of the model. The FDA algorithm is designed for the class of additively decomposable functions. It runs in $O(n\sqrt{n})$ time. The FDA assumes the knowledge about the decomposition structure of the objective function. However, the authors propose to eliminate this problem using an FDA that learns a factorization—a Bayesian algorithm called the LFDA. The LFDA algorithm inductively estimates the structure of the search space by detecting the mutually interacting variables.

The third paper by Masaharu Munetomo and David Goldberg develops the linkage identification by non-monotonicity detection (LIMD) algorithm that notes the dependencies among the search variables by a non-monotonicity detection procedure. The LIMD algorithm works by checking second order dependencies in $O(n^2)$ objective function evaluations. This algorithm works for problems where all the search variables can be grouped into a set of equivalence classes based on their mutual interaction. However, this paper also extends the LIMD approach for detecting the variable interaction for problems where the different classes of mutually interacting variables overlap.

The paper by William Langdon focuses on properties of the distribution of program fitness in genetic programming. It investigates the characteristics of such distributions with respect to the changes in the size of the programs. It makes use of the enumerative and Monte Carlo techniques to study these characteristics. It also reports that, once a minimum size threshold is reached, the fitness distribution appears to be almost independent of the program length. This work offers a systematic study of the convergence property of the fitness distributions for a large class of problems and an analytical proof of this for linear programs.

Erick Cantú-Paz and David Goldberg explore the performance speed-up of different models of parallel GAs as the number of processors is increased. They investigate two important classes of parallel GAs: (1) a single population-based parallel GA with master-slave architecture where different processors are used for parallel evaluation of the objective function and (2) a multi-population-based parallel GA where separate instances of the algorithm are executed in parallel along with migration-based exchange of population members.

This special issue presents an exposure to some important issues in scalable EC. Hopefully, this will take a small step toward crystallizing the growing effort to analyze and develop scalable evolutionary algorithms that come with a performance guarantee.