

Genet Program Evolvable Mach (2009) 10:339–341
DOI 10.1007/s10710-009-9094-1

EDITORIAL

Introduction: special issue on parallel and distributed evolutionary algorithms, part I

Marco Tomassini · Leonardo Vanneschi

Published online: 22 October 2009
© Springer Science+Business Media, LLC 2009

Parallel and distributed approaches are natural in evolutionary computation and they have been used intensively since the early years of this research field. Evolutionary algorithms, in fact, have often been described as *intrinsically parallel* computational methods. The reason for this is that many of the main computational tasks characterizing this family of heuristics are independent of each other; thus it is straightforward to perform them at the same time. This is the case, for instance, for the evaluation of the fitness of the individuals in a population. Furthermore, by attributing a non-panmictic structure to the population, something that also finds its inspiration in nature, genetic operations can be performed independently of each other and thus can also potentially be parallelized. These approaches can be useful even when there is no actual parallel or distributed implementation, thanks to the particular information diffusion given by the more local population structures. But of course parallel and distributed approaches are at their best when the structures of the models are reflected in the actual algorithm implementations. In fact, when compared with other heuristics, evolutionary algorithms are relatively costly and slow. But parallel and distributed implementations can boost performance and thereby allow practitioners to solve, exactly or approximately, larger and more interesting problem instances thanks to the time savings afforded.

As mentioned above, these advantages have been known and appreciated for at least three decades; so, what justifies a new presentation of the state of the art in the field? The answer is that significant progress has been made in recent years and these methodologies have become even more important today for at least two reasons. First,

M. Tomassini (✉)

Information Systems Department, HEC, University of Lausanne, Lausanne, Switzerland
e-mail: Marco.Tomassini@unil.ch

L. Vanneschi

Department of Informatics, Systems and Communication (D.I.S.Co.),
University of Milano-Bicocca, Milan, Italy
e-mail: vanneschi@disco.unimib.it

engineers and scientists wish to tackle problems of ever increasing complexity. The resulting increases in computational costs have been compensated for, in part, by the steady improvements in single-CPU performance, but they can obviously be decreased in an even more substantial manner by harnessing the implicit or explicit parallelism present in evolutionary heuristics. Second, while really fast parallel hardware was until recently extremely expensive, difficult to program, and available only to few groups, this has changed in a radical manner during the last decade. Nowadays, almost every scientist has easy access at least to networks of underexploited workstations, if not to dedicated parallel/distributed hardware in the form of clusters of powerful CPUs. Often these collective computational structures are part of hierarchical, geographically extended grid systems that offer unprecedented computational power at a very low cost. But the fast hardware that is now available would be much less exploitable if it was not accompanied by corresponding advances in software, giving rise to emerging standards and portable libraries of routines that are easy to use, such as MPI and the numerous existing programming environments for exploiting the computational power of computer grids. There is also another remarkable emerging trend, which is well represented in the papers that appear in the present issue: the use of graphical processing units (GPUs), which are inexpensive, can be easily harnessed, and provide excellent speedup in the case of evolutionary algorithms.

The present special issue was conceived for presenting high quality work on parallel and distributed approaches in evolutionary computation, targeting a wide set of researchers ranging from specialists in parallel computing for evolutionary computation to researchers interested in using the methodology for solving difficult practical problems.

We received 24 submissions in response to our call for papers, nine of which were accepted for publication after extensive review. Four of these papers appear in the present issue, while the other five will be published in a second forthcoming issue. All of the papers were subject to the same review and editorial standards as any regular paper in this journal.

One additional paper (by Robilliard et al.) also appears in the present issue. This paper was already under review by the journal when our call for papers was distributed, and although it was independently reviewed, the Editor-in-Chief felt that it fit the theme of the special issue well enough that it might also appear here, rather than in a subsequent regular issue. We read the paper with this suggestion in mind and we are pleased to include it.

In the first paper, Veber et al. describe a new distributed differential evolution model based on communicating subpopulations of constant and variable sizes. The model has been tested on a large set of benchmark problems with excellent results when compared with previous approaches.

The second paper, by Asouti et al. is about a difficult real-world engineering optimization problem in aerodynamics. They use an asynchronous approach implemented on a grid computing multi-layer structure. This is a typical example of the degree of sophistication and of the high computational needs of today's real-world problems.

The third paper, by Munawar et al. deals with a classical hard combinatorial optimization problem: MAX-SAT. Their metaheuristic approach combining an evolutionary algorithm with local search is particularly interesting as their implementation makes use of GPUs, which are relatively inexpensive and increasingly popular hardware devices.

The fourth paper, by Walker et al. presents a novel parallelization technique in Cartesian Genetic Programming called the multi-chromosome approach. The method is particularly useful for problems in which the solution can be broken down into simpler problems each of which has its own chromosome in the genotype. The approach has been successfully tested on a circuit-design problem and on the classification of mammograms.

The final paper in this issue, by Robilliard et al. continues the exploration of the use of graphical processing units for evolutionary computation. Working with the widely-used ECJ system they produced a novel system for genetic programming on Nvidia G80 GPUs and demonstrated significant new speed improvements.

The guest editors thank all the authors who submitted their work for consideration. We are also much indebted to the many colleagues who kindly made their time available to assist in the reviewing process. Special thanks go to Lee Spector, Editor-in-Chief, for his continuing assistance and encouragement during all the phases of the project. Finally, the issue could not have been possible without the dedication and professional work of the people at Springer, in particular Melissa Fearon, Emma Kalb, and Vijay Kumar.