Nat Comput (2013) 12:133–134
DOI 10.1007/s11047-013-9371-z

Preface

Juan J. Merelo · Maribel García Arenas · David Corne · Juan Luis Jiménez Laredo · Francisco Fernández de Vega

Published online: 3 May 2013

1 Preface

The broad area of 'informal computing' refers to ways of creating computing systems which are not fixed or bound to any strict organization, in terms of either the hardware resources on which the computation executes, and/or in terms of the software and its parameters. Examples of such informal environments include the following:

- Parasitic or stealth computing: using computing resources without explicit authorization from the user, for instance by visiting a web page.
- Volunteer computing: the user submits resources to a pool explicitly, by running a program or visiting a web site.
- Freeriding computing: using computing resources which are free or available, to a certain extent, in the network; for instance, using Google Apps or resources such as Wolfram-Alpha. This is similar to parasitic computing, except that the provider of those resources is aware of the situation but generally unconcerned.
- Ubiquitous computing: using computing power available in user devices such as mobile phones or other appliances.

J. J. Merelo · M. G. Arenas
University of Granada, Granada, Spain

D. Corne (✉)
Heriot-Watt University, Edinburgh, UK
e-mail: dwcorne@macs.hw.ac.uk

J. L. J. Laredo
University of Luxembourg, Luxembourg City, Luxembourg

F. Fernández de Vega
University of Extremadura, Extremadura, Spain

Using these (and similar) kinds of computing environments presents a number of significant challenges, since neither the topology nor the availability of a particular node is typically known in advance. Also, computing nodes (and the connections to them) will have different performance profiles, capabilities, and availability profiles. Nevertheless, informal environments present significant opportunities that can be exploited by those with large-scale and compute-intensive algorithmic tasks, to the extent that the algorithmic structure of those tasks can operate successfully in such environments.

Evolutionary computation is among the few algorithmic frameworks that can be expected to cope well in such informal environmental configurations, while still able to perform successful optimization (or other) tasks. It is well known that the overall performance of a distributed evolutionary algorithm is highly robust to variation in practically all of its algorithmic building blocks, just as evolution in nature thrives in greatly varied and dynamic conditions. Nevertheless we are at the early stages in learning how to exploit informal environments, as well as in understanding how best to design the algorithms that will inhabit them.

It was with this vision that in 2011 the first 'International Workshop on Distributed Evolutionary Computation in Informal Environments' (IWDECIE) was organized. The IWDECIE event took place as a half-day workshop under the auspices of the IEEE Congress on Evolutionary Computation 2011, which was held in New Orleans between June 5th and June 8th 2011. This special issue includes significantly extended versions of a small selection of the work that was presented at IWDECIE, along with additional papers that were solicited via a targeted call for papers.

The result is a collection of five articles that chart a snapshot of ongoing work in the general area of distributed evolutionary computation in informal environments. To some extent echoing the balance of research in this area, four of the articles focus on the particular types of informal distributed environment in which we are starting to see more and more examples of evolutionary computation research, while one has an application focus.

In the first paper, Meri et al. study ‘cloud-based’ evolutionary algorithms, in which the environment is a collection of distinct computers that are connected via wi-fi or Ethernet, and in which advantage is taken of freely available storage services such as Dropbox™ and SugarSync™. This paper pays particular attention to how the distributed evolutionary algorithm is able to take advantages of new computers dynamically joining the cloud. They also evaluate how the performance of the algorithm on different problems is sensitive to the particular choice of storage service used.

In the second paper, Merelo et al. describe the iterated design of SofEA, an architecture for distributing evolutionary algorithms across computer networks in an asynchronous and decentralized way. SoftEA is based on a pool architecture implemented on an object store, allowing the asynchronous interaction with several clients. The autonomy of the clients leads to complex behaviour, making it very difficult to design aspects of the architecture in advance using analytic methods. The paper shows how appropriately chosen experimentation and measurement comprise a good approach to the design of such asynchronous, heterogeneous and distributed systems.

In the third paper, Fernandez de Vega et al. focus on the powerful computation opportunities available in the innumerable desktops and other PCs found at typical institutions, normally barely used and devoted entirely to administrative tasks. Using the BOINC (http://boinc.berkeley.edu/) open-source software platform, they present a new model for running unmodified applications with a web-based centralized management system, making it easy for researchers to exploit these barely used resources to run scientific applications without modifying the application’s source code, and manage thousands of computers from a single web page. They test their approach in the context of applying genetic programming on a real-world problem.

In the fourth paper, the focus shifts towards how the algorithmic structures can gracefully adapt to exploit the varying capabilities of the heterogeneous nodes in the environment. In this paper, Domnguez and Alba describe and explore an algorithm called ‘Ethane’, which adaptively reconfigures its components to match the different capabilities of the compute nodes on which it finds itself running. The overall approach to this adaptive behaviour takes inspiration from chemical bonding in hydrocarbons. Their experiments suggest that this is a robust and promising approach.

Finally, Bodas-Sagi et al. focus on an application, in which they use a multi-objective evolutionary algorithm to optimize the parameters of technical indicators for stock market investment. Their results are promising, suggesting that trading decisions based on their optimized parameters are more profitable. Of particular interest from our viewpoint is that they achieved their results by using informal distributed environments, relying in particular on exploiting the idle-time in a corporate technology infrastructure.

We hope you find this issue, as we do, to be a stimulating and inspiring introduction to an emerging subfield at the interfaces of distributed, natural and social computation. Meanwhile, we are very grateful to the several scientists who helped in the preparation of this special issue by providing timely and helpful anonymous reviews. We are also grateful to the NACO administrators and editors for helping us bring this special issue to press.

Prof. Juan J. Merelo,
Prof. Maribel García Arenas
Prof. David Corne
Prof. Juan Luis Jiménez Laredo
Prof. Francisco Fernández de Vega