Visualizing for Success: How Can we Make the User more Efficient in Interactive Evolutionary Algorithms?

Juan-J. Merelo[*]
Dept. of Computer
Architecture and Technology
and CITIC
University of Granada,
Granada, Spain
jmerelo@ugr.es

Mario García-Valdez
Dept. of Graduate Studies
Instituto Tecnológico de
Tijuana, Tijuana, Mexico
mario@tectijuana.edu.mx

Carlos Cotta
Dept. LCC, ETSI Informática
Universidad de Málaga,
Málaga, Spain
ccottap@lcc.uma.es

1. INTRODUCTION

The main objective in volunteer computing systems is to get users to contribute to the experiment with as much time as it is possible. In the case of systems such as NodIO [1], a web-service based system which provides a minimal infrastructure to support pool-based distributed evolutionary computing experiments, there is another issue with the user: the fact that there is some interaction that can be done besides simply loading the web page once.

NodIO has been used for volunteer computing experiments by embedding an island-based evolutionary algorithm inside a web page and making them interchange information via the NodIO server, in a spontaneous and eventually, since all islands communicate via the server, panmictic distributed evolutionary algorithm. In the experiments we have carried out so far we have realized that the user is spontaneously making it an *interactive* algorithm by applying *hypermutation* operators or even *distributing* the population themselves if the performance of their machine allows them to do so. However, it will be impossible for the user to perceive what is happening if the visualization does not convey that information in a way that (1) can be first understood by anyone without using any technical terms and (2) does not add too much overhead to the algorithm itself. As far as we know, the problem of visualizing an evolutionary algorithm so that the non-technical and spontaneous user can help it find the solution faster has not been fully solved. We have made several attempts that we will present in this paper, but this paper is basically a request for comments during the workshop and later on so that the scientific community understands the basis of this problem and can also design their own solutions for it. To reach this goal other aspects of the design must also be addressed in order to improve the overall usabilty of the system.

The client displays three panels, showing three different visualizations of the evolutionary algorithm. Two of them have been kept fixed for all experiments, and they are shown in Figure 1. The top panels show, to the right, the fitness in the last 5000 generations at 100 generations interval. The bottom panel shows a visualization of the best individual, using blocks that represent the number of ones in 4-bit blocks with different shades that represent their block fitness, from 0 (transparent) to 2 (black). 1 is the intermediate fitness, achieved with all 0s, and it is represented by dark gray.

From the point of view of the user, this is probably one of the best indicators of the stagnation of the simulation. It could help the user to restart the simulation in that case, but it is almost impossible for the user to forecast when that will be happening to avoid it, either by restarting or by starting new islands in other browser tabs.

This information might be redundant with that shown in the fitness panel. However, it helps when you have several windows, side by side. If the best chromosome is the same in both, not only a single island is stuck, but the whole number of islands contributing are. That happens usually when there is a single island running for some time in that situation and all of a sudden several other volunteers come in. As soon as they start to receive random individuals from the cache, they are bound to get that *super* individual, which will eventually *invade* the population, crashing global diversity. The best situation is when several individuals arrive at short intervals, contributing diversity but also individuals that are not so different; by the *intermediate disturbance hypothesis* that situation will boost the performance of the whole system. However, the user in general will run a single tab, so they will not know that is happening; this is a kind of information that we would need to convey to the user by way of visualization. But that is why we use the third panel.

Initially we used as third panel the number of IPs that had participated so far. This was mainly for the benefit of us, not really the user; it was a way of knowing the impact of some particular announcement and also to see how many users were there at a particular time. However, the user could gather no information either about his particular involvement or about the progress of the simulation. Besides, it was the *total* number of IPs from the beginning of the batch of experiments, it did not show the instantaneous, or in the last minutes, number of participating IPs. That is why we opted to change that visualization to a new one that at least included some information on it.

That is not exactly what is shown in the graph, that shows

GECCO'16 Companion, July 20 - 24, 2016, Denver, CO, USA
© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ISBN 978-1-4503-4323-7/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2908961.2931742

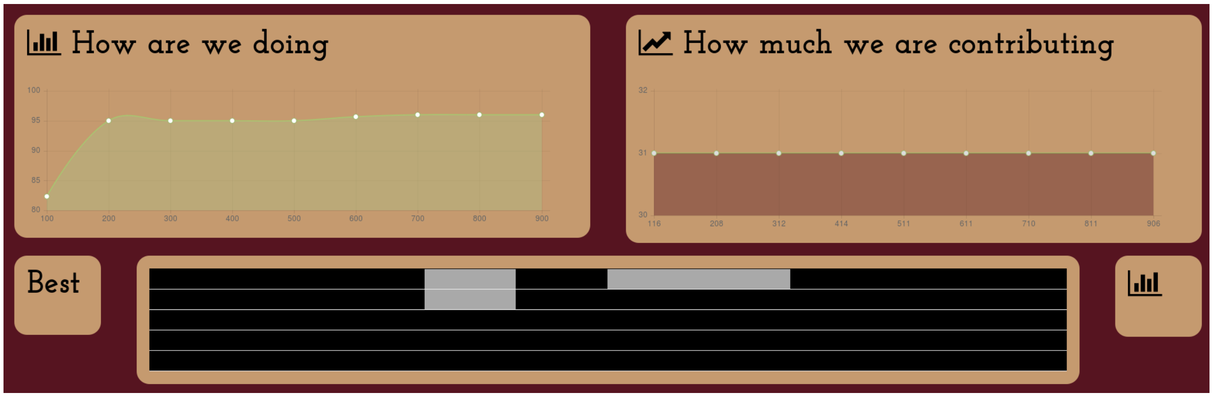

Figure 1: The three panels showing the latest fitness, the cache size and the best chromosome using 4 shades from white to black to represent the number of ones in every block in the Trap problem at the top; at bottom, the panel showing 5 black bars indicating that all but 4 traps have been found.

the cache size. However, if you combine this panel with the other, and the fitness panel and the chromosome panel shows some change, you will see if that change is incorporated to the cache if that changes size. If the cache remains the same, no new individuals are being generated and again you can have a perspective of the whole experiment being stagnated in a single view; the user can then decide to restart or also to create new populations. It is still a poor way of conveying global information and it requires a certain amount of information given to the user so that he or she can understand what is going on.

2. WHAT CAN BE VISUALIZED AND HOW IT COULD HELP

For the time being, the only possible interactions the user has is to start or restart new islands. However, this does not need to be the only way the user can help the simulation. Even keeping the user interface as simple as possible, other possibilities could include: (i) *killing* the best individual, thus immediately creating diversity by forcing the rest to compete; (ii) increasing or decreasing the population size. A bigger population will have higher initial diversity; (iii) increasing or decreasing the mutation or crossover rate. For the time being, bitflip mutation is used on every new generated individual.

However, even if whatever the user does will contribute to diversity, if it is done blindly it will explore more than exploit and thus be a waste of computational resources. That is why there are several possible ways of showing progress and diversity: (i) represent the convergence of the population towards a single individual by showing, using grayscale, the uniformity of values for all bits; (ii) represent global diversity by having a measure of compression entropy plotted, instead or alongside the cache size.

This can be complemented by other visualization techniques, such as (i) use Twitter or a Telegram channel to show the progress of the current experiment, or making calls when a new experiment has started to have an early infusion of diversity, (ii) represent the progress of the particular user against the rest of the users, showing the fitness of every one, for instance, or the time every one has spent represented by the number of PUTs that have been made, (iii) keep past history and represent the current experiment comparing its progress to the progress of previous ones, (iv) show global fitness alongside local fitness,

More important maybe that the data itself is the way of representing it. A visualization that shows how well or badly the global algorithm is doing would help to encourage the users to take measures to improve the situation. That is why we finish this paper with a series of research questions

3. RESEARCH QUESTIONS

With this paper we would like to raise a series of research questions on the good use of visualization of evolutionary algorithms in distributed voluntary evolutionary computing experiments: (i) how can we represent in a meaningful but at the same time simple way the global progress of the simulation so that the user can take action if he needs to? (ii) how can we also represent diversity so that it conveys to the user the fact that he or she can help by taking whatever measures the client allows, even a simple restart? (iii) should we use some kind of reinforcement, indicating whether the individuals created locally by the user have helped find the global solution? (iv) how could we gamify participation so that, at the same time, cheating is discouraged? (v) how can we implement this ideas with a minimum overhead to the system? (vi) Finally, how can any of these measures be taken so that it is not a impossible overhead for the algorithm itself?

Acknowledgments

This work has been supported in part by TIN2014-56494-C4-{1,3}-P (Spanish Ministry of Economy and Competitivity), PROY-PP2015-06 (Plan Propio 2015 UGR). Additional support was received by Projects 5622.15-P (ITM) DNEMESIS P10-TIC-6083 (Junta de Andalucía) and PROINNOVA 2016: 230630 (CONACYT).

4. REFERENCES

[1] J.-J. Merelo-Guervós and P. García-Sánchez. Designing and modeling a browser-based distributed evolutionary computation system. In J. L. J. Laredo, S. Silva, and A. I. Esparcia-Alcázar, editors, *Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, Companion Material Proceedings*, pages 1117–1124. ACM, 2015.