SIGEVOlution

newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation

June 2006
Volume 1 Issue 2

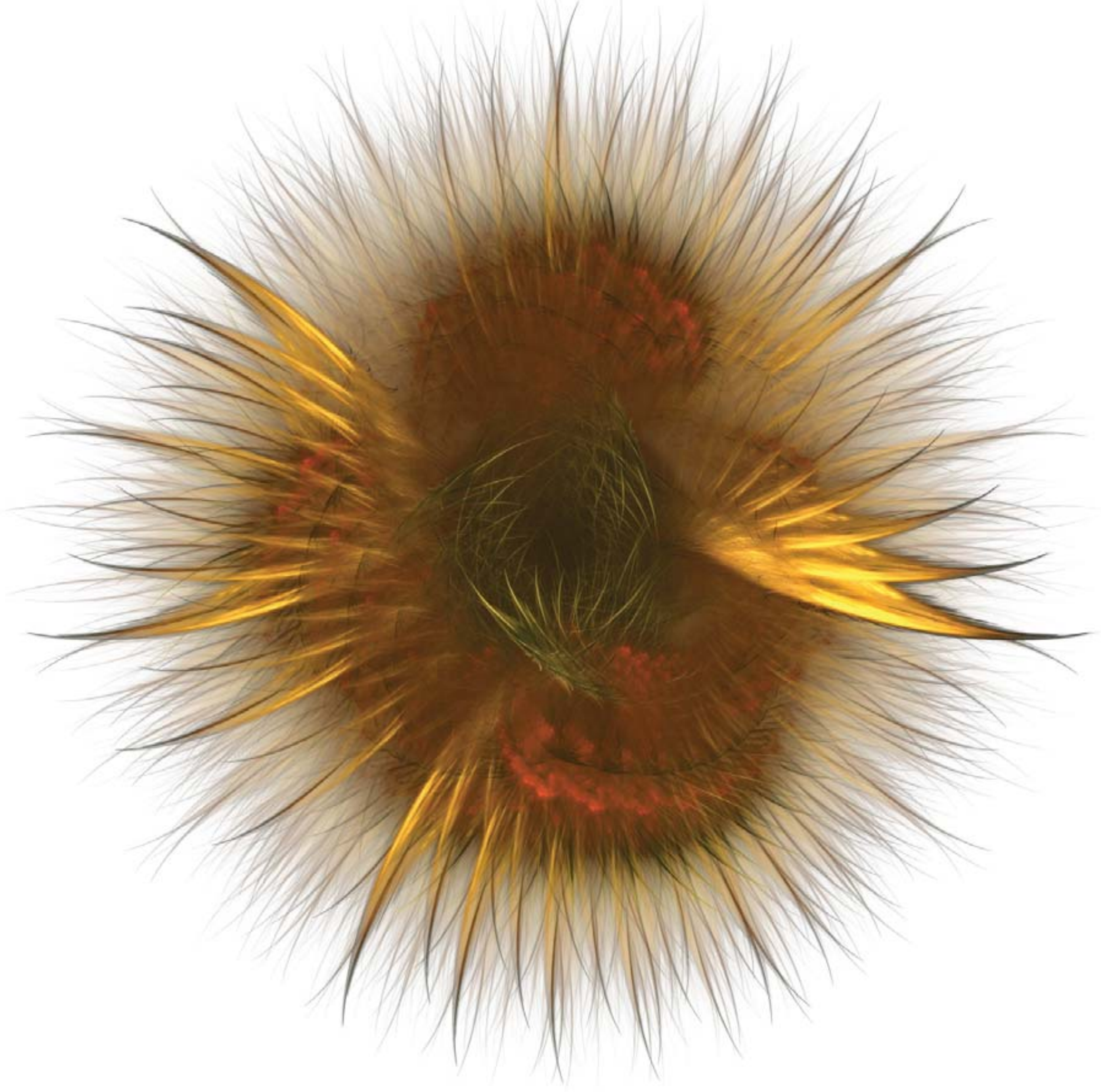
in this issue

The Network of EC Authors
Carlos Cotta & Juan-Julián Merelo

The Electric Sheep
Scott Draves

A Survey of μGP
Massimiliano Schillaci &
Ernesto Sánchez

The Columns
GECCO-2006 highlights
events reports
letters
forthcoming papers
new books
calls & calendar



Editorial

GECCO is just a few days away, it is time to pack things up and leave for Seattle! I am always happy when this time of the year comes. Not only GECCO has always a terrific technical program and gives access to several, exciting, EC related events (nine workshops and tens of tutorials this year!). GECCO is also the time when I am able to meet with long time friends and to catch up with state-of-the-art research in the field. So, just before I leave, let me introduce you to the new issue of SIGEVOlution.

This issue brings you two contributed papers, the software corner with an overview of μGP, and several columns. At first, Carlos Cotta and Juan-Julián Merelo take us inside our community with an analysis of the co-authorship network that arises from a major EC bibliography. Then, Scott Draves (a.k.a. Spot) overviews his famous creature, the *Electric Sheep*, a distributed screen-saver which exploits users' votes to evolve animated artificial "sheep". Spot also presents the most recent physical manifestation of the sheep, *Dreams in High Fidelity*. In the software corner, Massimiliano Schillaci and Ernesto Sánchez introduce us to μGP, an evolutionary framework that, among the others, has been used to evolve assembly programs for the Core Wars competition. In the 2005 edition of the Human-Competitive Results, this application of μGP received a silver award. The next columns provide information about what we will find in Seattle during GECCO, what happened during the NCSA/IlliGAL Gathering on Evolutionary Learning, the forthcoming issues of EC related journals, new books, and the calendar of EC events.

As always, I wish to thank all the people who helped me with the newsletter, the board members, Dave Davis and Martin Pelikan, Pat Cattolico from GECCO-2006, and the authors, Carlos Cotta, Scott Draves, Juan-Julián Merelo, Ernesto Sánchez, and Massimiliano Schillaci.

See you in Seattle!

Pier Luca
June 30th, 2006

SIGEVOlution
June 2006, Volume 1, Issue 2

Newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation.

Contributors to this Issue

Carlos Cotta
Juan-Julián Merelo
Scott Draves
Massimiliano Schillaci
Ernesto Sánchez

Contents

The Complex Network of EC Authors **2**
Carlos Cotta & Juan-Julián Merelo
The Electric Sheep **10**
Scott Draves
A Brief Survey of μGP **17**
Massimiliano Schillaci & Ernesto Sánchez
GECCO-2006 Highlights **22**
Events Reports **26**
Letters **27**
Forthcoming Papers **28**
New Books **29**
Calls and Calendar **32**
About the Newsletter **36**

ISSN: 1931-8499

The Complex Network of EC Authors

Carlos Cotta, Dept. Lenguajes y Ciencias de la Computación, University of Málaga, Spain, ccottap@lcc.uma.es
Juan-Julián Merelo, Dept. Arquitectura y Tecnología de Computadores, University of Granada, Spain, jj@merelo.net

The study of all kind of complex networks has undergone an accelerated expansion in the last few years, after the introduction of models for scale-free/power-law [2] and small-world [12] networks, which, in turn, has induced the study of many different phenomena under this new light. Co-authorship patterns are one of them. Nodes in co-authorship networks are paper authors, joined by edges if they have written at least one paper together. Even as most papers are written by a few authors staying at the same institution, science is a global business nowadays, and lots of papers are co-authored by scientists continents apart from each other. There are several interesting facts that can be computed on these co-authorship networks: first, what kind of macroscopic values they yield, and second, which are the most outstanding *actors* (authors) and edges (co-authorships) within this network. An understanding of the structure of the network and what makes some nodes stand out goes beyond mere curiosity to give us some insight on the deep workings of science, what makes an author popular, or some co-authors preferred over others.

Co-authorship networks are studied within the field of sociometry, and, in the case at hand, scientometry. First studies date back to the second half of the nineties: Kretschmer [5] studied the *invisible colleges* of physics, finding that its behavior was not much different to other collaboration networks, such as co-starring networks in movies. However, it was at the beginning of this century when Newman [8, 9] studied co-authorship networks as complex networks, giving the first estimations of their overall shape and macroscopic properties. In general, these kind of networks are both small worlds [12], that is, there is, on average, a short distance between any two scientists taken at random, and scale free, which means they follow a power law [2] in several node properties (e.g., the number of nodes linking to a particular one). Newman made measurements on networks from several disciplines: physics, medicine and computer science, showing results for clustering coefficients (related to transitivity in co-authorship networks), and mean and maximum distances (which gives an idea of the shape of the network, and, thus, of the mechanism that underlie this social network). Barabási and collaborators [1] later showed that the scale free structure of these co-authorship networks can be attributed to preferential attachment: authors that have been more time in business have generally published more papers on average, thus getting more exposure, and more new links than novel authors. However, even as this model satisfactorily explains the overall structure of the network, there must be much more in the author positions in the network that just having been there for more time.

In this work, we analyze the co-authorship network of evolutionary computation (EC) researchers. Studying this network will give us a better understanding of its cohesiveness as a discipline, and might shed some light on the collaboration patterns of the community. It also provides interesting hints about who are the central actors in the network, and what determines their prominency in the area.

The bibliographical data used for the construction of the scientific-collaboration network in EC has been gathered from the DBLP –*Digital Bibliography & Library Project*– bibliography server, maintained by Michael Ley at the University of Trier. This database provides bibliographic information on major computer science journals and proceedings, comprising more than 610,000 articles and several thousand computer scientists (as of March 2005). We have defined a collection of terms that include the acronyms of EC-specific conferences –such as GECCO,

	EC	Medline	Physics	SPIRES	NCSTRL
total papers	6199	2163923	98502	66652	13169
total authors	5492	1520251	52909	56627	11994
mean papers per author (PA)	2.9	6.4	5.1	11.6	2.6
mean authors per paper (AP)	2.56	3.75	2.53	8.96	2.22
collaborators per author (CA)	4.2	18.1	9.7	173.0	3.6
size of the giant component	3686	1395693	44337	49002	6396
as a percentage	67.1%	92.6%	85.4%	88.7%	57.2%
2nd largest component	36	49	18	69	42
clustering coefficient (CC)	0.808	0.066	0.43	0.726	0.496
mean distance	6.1	4.6	5.9	4.0	9.7
diameter (maximum distance)	18	24	20	19	31

Table 1: Summary of results of the analysis of five scientific collaboration networks (data not corresponding to EC is taken from [8]).

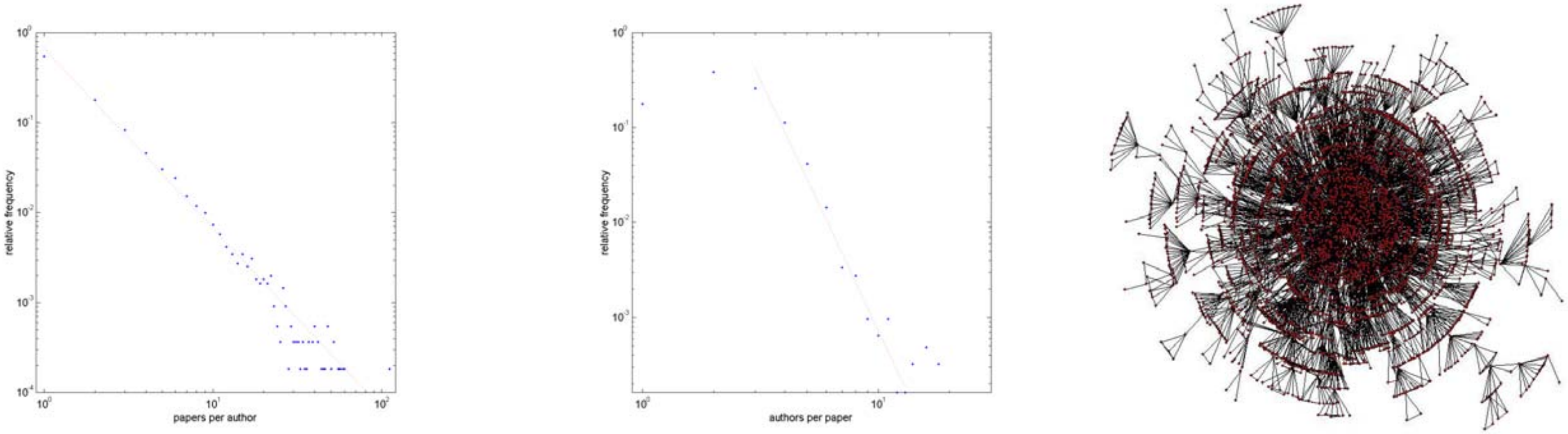
PPSN or EuroGP– or keywords –such as “Evolutionary Computation”, “Genetic Programming”, etc.– that are sought in the title or in the publication forum of papers. Using an initial sample of authors (those that have published at least one paper in the last five years in any of the following large EC conferences: GECCO, PPSN, EuroGP, EvoCOP, and EvoWorkshops), their list of publications is checked for relevance, and the corresponding co-authors are recursively examined. Just as an indication of the breadth of the search, the number of authors used as seed is 2,536 whereas the final number of authors in the network is 5,492, that is, more than twice as many.

1 Macroscopic Network Properties

The overall characteristics of the EC co-authorship network are shown in Table 1 alongside with results obtained by Newman [8, 10]. The latter correspond to co-authorship networks in Medline (biomedical research), the Physics E-print archive and SPIRES (several areas of physics and high-energy physics respectively), and NCSTRL (several areas of computer science). First of all, the number of EC papers and authors is much smaller than those for the communities studied by Newman; however, it must be taken into account that these communities are much more general and comprise different subareas. Notice also that in most aspects, EC data seems closer to the NCSTRL database than to any other. This indicates that despite the interdisciplinary nature of EC (with researchers that come from very diverse fields, e.g., biology, engineering, mathematics, physics, etc.), publication practices of this area are still those of computer science. This way, average scientific productivity per author (2.9) is not so high as in physics (5.1, 11.6) and biomedicine (6.4). It nevertheless follows quite well Lotka’s Law of Scientific Productivity [6], as shown by the power law distribution illustrated in Figure 1a. The most interesting feature is the *long tail*: while most authors appear only once in the database, there are quite a few that have authored dozens of papers.

The average size of collaborations (2.56) is also smaller than in biomedical research (3.75) or high-energy physics (8.96), although similar to those of average physicists (2.53), and slightly superior to average computer scientists (2.22). It also follows a power law (from 3 authors on) as shown in Figure 1b. Notice the peak in the tail of the distribution, caused by the large collaborations implied by proceedings, whose role will be studied in next section.

Relevant considerations can also be done regarding the mean number of collaborators per author (4.2); physics and biomedicine are areas in which new collaborations seem more likely than in EC (9.7, 173.0, and 18.1). However, the figure for NCSTRL (3.6) is clearly lower than for EC, thus suggesting that the EC author is indeed open to new collaborations, as regarded from a computer science perspective. The histogram of number of collaborators per authors (not shown) also fits quite well to a power law with exponent -2.58. In this case, this power law can be attributed to a model of preferential attachment such as the one proposed by Barabási



(a) (b) (c)

Figure 1: (a) Histogram of the number of papers per author. The slope of the dotted line is -2.00. (b) Histogram of the number of authors per paper. The slope of the dotted line is -5.27. The peak in the tail of the distribution is caused by the large collaborations implied by proceedings. Their role will be examined in Section 2. (c) Graphical representation of the giant component of the EC co-authorship network. A dense core with heavily connected authors can be distinguished, with *tendrils* sprouting out of it that include authors with less collaborators.

et al. [1]: *new* authors tend to link (be co-authors) to those that have published extensively before. However, as we pointed out before, that cannot be the whole story. For starters, information on who is the most prolific author is not usually available (although educated guesses can go a long way), and besides, there are strong constraints that avoid free linking: a person can not tutor too many PhD students at the same time, for instance, and not everybody is ready, or able, to move to the university of the professor he/she wants to work with. However, let us point out that actors with many links do not necessarily coincide with the most prolific; they are rather persons that have diverse interests, reflected in their choice or co-authors, participate in transnational projects, or have a certain wanderlust, being visiting professors in many different institutions, which leads them to co-author papers with their sponsors or hosts in those institutions. The fact that the clustering coefficient in the EC co-authorship networks is so high, and the mean degree of separation is so close to the proverbial six degrees, implies that in general all authors in this field are no more than 6 degrees of separation of those *sociometric stars* with a wide variety of interests, projects or visits. These sociometric stars will be analyzed more in depth in the next section.

Another interesting aspect refers to the so-called *giant component*, a connected subset of vertices whose size encompasses the majority of network vertices. The remaining vertices group in connected components of much smaller size (actually, independent of the total size of the network). As pointed out in [10], the existence of this giant component is a healthy sign, since it shows that most of the community is connected via collaboration, and hence, ultimately by person-to-person contact. In the case of the EC network, the giant component comprises more than 2/3 of the network (see Figure 1c; this accounts for the “giant” denomination), again superior to the computer science network, but significantly smaller than for physics or biomedicine. This fact is nevertheless counteracted by the high clustering coefficient (actually the highest of the set). This indicates a much closer contact among actors, since one’s collaborators are very likely to collaborate among themselves too. It is also significant that the mean distance among actors is halfway between the medical/physics communities (around 4) and the computer science community (around 9), while diameter is the second-smallest. In some sense, this indicates that the community is less hierarchical than computer science in general, yet not so decentralized as physics.

	# of co-workers		betweenness		closeness	
1.	K. Deb	98	K. Deb	12.94	K. Deb	3.48
2.	D.E. Goldberg	75	D.E. Goldberg	9.66	W. Banzhaf	3.67
3.	R. Poli	67	D. Corne	7.01	D.E. Goldberg	3.72
4.	M. Schoenauer	62	X. Yao	5.36	R. Poli	3.72
5.	W. Banzhaf	58	W. Banzhaf	5.23	H.-G. Beyer	3.76
6.	D. Corne	56	H. de Garis	4.69	P.L. Lanzi	3.77
7.	X. Yao	56	R. Poli	4.66	D. Corne	3.86
8.	J.A. Foster	54	J.J. Merelo	4.41	M. Schoenauer	3.89
9.	J.J. Merelo	53	H. Iba	4.40	E.K. Burke	3.90
10.	J.F. Miller	51	M. Schoenauer	4.30	D.B. Fogel	3.91

Table 2: Actors with the highest centrality in the giant component of the EC network. Rankings are shown for three quantities: number of coauthors, betweenness and closeness. Betweenness values have been divided by 10^5.

2 Evolutionary Computation Sociometric Stars

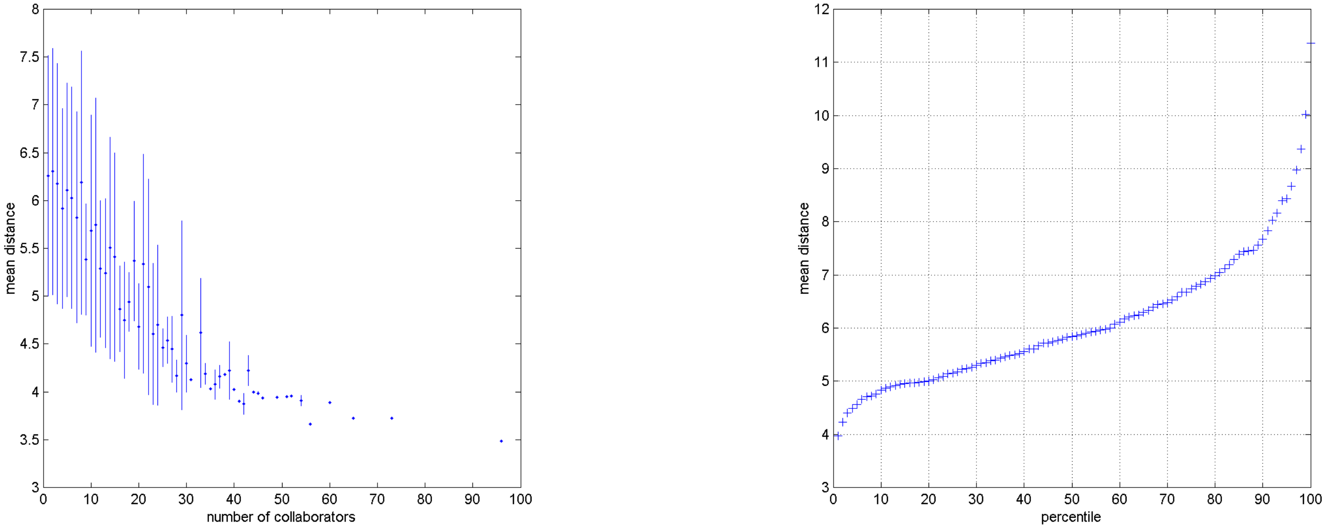
In the previous section we have considered global collaboration patterns that can be inferred from the macroscopic properties of the network. Let us now take a closer look at the fine detail of the network structure. More precisely, we are going to identify which actors play a more prominent role in the network, and analyze why they are important. The term *centrality* is used to denote this prominency status for a certain node.

Centrality can be measured in multiple ways. We are going to focus on metrics based on geodesics, i.e., the shortest paths between actors in the network. These geodesics constitute a very interesting source of information: the shortest path between two actors defines a “referral chain” of intermediate scientists through whom contact may be established – cf. [10]. It also provides a sequence of research topics (recall that common interests exist between adjacent links of this chain, as defined by the co-authored papers) that may suggest future joint works.

The first geodesic-based centrality measure we are going to analyze is *betweenness* [4], i.e., the relative number of geodesics between any two actors j,k passing through a certain i, summed for all j,k. This measure is based on the information flow between actors: when a joint paper is written, the authors exchange lots of information (such as knowledge of certain techniques, research ideas, potential development lines, or unpublished results) which can in turn be transmitted (at least to some extent) to their colleagues in other papers, and so on. Hence, actors with high betweenness are in some sense “hubs” that control this information flow; they are recipients –and emitters– of huge amounts of cutting-edge knowledge; furthermore, their removal from the network results in the increase of geodesic distances among a large number of actors [11].

The second centrality measure we are going to consider is precisely based on this geodesic distance. Intuitively, the length of a shortest path indicates the number of steps that research ideas (and in general, all kind of memes) require to jump from one actor to another. Hence, scientists whose average distance to other scientists is small are likely to be the first to learn new information, and information originating with them will reach others quicker than information originating with other sources. Average distance (i.e., *closeness*) is thus a measure of centrality of an actor in terms of their access to information.

The result of our centrality analysis of the EC network is shown in Table 2. Regarding betweenness, the analysis provides clear winners, with large numerical differences among the top actors. These differences are not so marked for closeness values with all top actors clustered in a short interval. Notice that there are some actors that appear in both top-lists. For example, David E. Goldberg, author of one of the most famous books on EC, figures prominently in all rankings, as well as Kalyanmoy Deb, who is a well known author in theoretical EC and multi-objective optimization. The rest of the authors are well known as leaders of subfields within EC,



(a)

(b)

Figure 2: (a) Mean distance to other authors as a function of the number of collaborators. The error bars indicate standard deviations. (b) Percentile distribution of mean distances in the giant component.

or as having an active role in conference organization. Using Milgram's terminology [7], they are the *sociometric superstars* of the EC field.

Several factors are responsible for the prominent status of these actors. Obviously, scientific excellence is one of them. This excellence is difficult to measure in absolute, objective terms, but the number of co-authored papers –and indirectly, the number of collaborators– is a possible estimator (notice that we are thus measuring the efficiency in knowledge transmission, which is the ultimate goal of scientific publishing). This quantity is shown for the ten highest-degree actors in the network in Table 2. Certainly, some correlation between degree and centrality is evident. This is further illustrated in Figure 2a. As it can be seen, there is a trend of decreasing average distance to other actors as the actor degree increases (the correlation coefficient is -0.72 for actors in the top 25% percentile of distance). By crossing this information with the percentile distribution of distances shown in Figure 2b we can obtain some interesting facts about the collaborative strength of elite scientists. For example, consider the top 5% percentile; it is composed of actors whose average distance to the remaining actors is at most 4.61. If authors are grouped according to their number of collaborators, Figure 2a indicates that 23 collaborators are required at least to have an average group distance below this value. A more sensitive analysis indicates that 33 collaborators are required to have a statistically significant (using a standard t-test) result.

Another important factor influencing the particular ranking shown above is the presence of conference proceedings among authors' publications. These play a central role in the creation and structure of the network, to the point that some features change dramatically if links arising from proceedings co-authorship are removed. To begin with, the visual aspect of the network is different, as is shown in Figure 3a (compare it to the network with proceedings included, shown in Figure 1c). The reader should notice that the core is much more diffuse (actually, it looks like there are several micro-cores, plausibly corresponding to different EC subareas).

Although the degree-based properties of the network are in this case very similar (PA = 2.86, AP = 2.53, CA = 3.9, CC = 0.807) since only

	# of co-workers		betweenness		closeness	
1.	D.E. Goldberg	63	D.E. Goldberg	12.77	Z. Michalewicz	4.95
2.	K. Deb	55	K. Deb	11.29	K. Deb	4.99
3.	M. Schoenauer	52	M. Schoenauer	7.14	M. Schoenauer	5.03
4.	X. Yao	42	H. de Garis	7.11	A.E. Eiben	5.06
5.	H. de Garis	41	Z. Michalewicz	7.09	B. Paechter	5.08
6.	T. Higuchi	40	T. Bäck	5.81	D.E. Goldberg	5.09
7.	Z. Michalewicz	40	R.E. Smith	5.33	T. Bäck	5.35
8.	L.D. Whitley	39	X. Yao	5.11	D.B. Fogel	5.38
9.	M. Dorigo	38	A.E. Eiben	4.85	J.J. Merelo	5.40
10.	J.J. Merelo	38	B. Paechter	4.54	T.C. Fogarty	5.40

Table 3: Most central actors after removing proceedings. Betweenness values have been divided by 10^5.

a few links are removed, geodesic-based properties do change significantly as reflected in Figure 3b: without proceedings, the average and maximum distances increase by 2 units (from 6.1 to 8.0 and from 18 to 20 respectively), and the modal distance increases by 3 units (from 5 to 8). The resulting distribution is also much more symmetric than the original distribution, which was notably skewed towards low values. This can be explained by the very distinctive *authoring* patterns of proceedings: they are usually edited by a larger number of researchers, typically corresponding to the different thematic areas included in the conference or symposium. These are often senior researchers, with a prominent position in their subareas (thus, centrality and proceeding editorship reinforce each other). Furthermore, the fact that editors come from different areas contribute to the creation of long-distance links, resulting in a dramatic overall decrease of inter-actor distances.

If we exclude proceedings from the network, we obtain an image of the community in pure technical research terms. We have done this, obtaining the results shown in Table 2. As it can be seen, there is now a higher agreement between the two centrality measures (7/10 are the same in the top 10, and 39/100 in the top 100, vs. 6/10 and 19/100 respectively before). Furthermore, researchers of unquestionable scientific excellence who were not in the previous ranking do appear now. For example, Zbigniew Michalewicz, author of several excellent EC books, is now the author with the highest closeness, the fifth-highest centrality, and the seventh highest number of collaborators.

3 Discussion and Conclusions

In this paper, we have made a study of the co-authorship network in the field of evolutionary computation, paving the way to study the impact of certain measures, such as grants, the establishment of scientific societies or new conferences, has on the subject. The general features of the network suggest that it is quite similar to the field it can be better placed, computer science, but, at the same time, authors are much more closely related to each other. We have also taken into account the impact coeditorship of proceedings have on the overall aspect of the network and most centrality measures. This issue had not been considered in previous related works, and we believe that it plays an important role in distorting some network properties. We suggest to exclude them in the future in this kind of studies.

In connection to this latter issue, we believe that co-authorship networks created by different kind of papers (technical reports, conference papers, journal papers) might be different owing to the different kind of collaboration they imply. Consider that while technical reports may be written in a hurry and present very preliminary results, conference papers are usually somewhat more long term, and journal papers really indicate a committed scientific relationship (due to the long time they take to be published and the several iterations of the revision process). The authors suggest to approach them separately and analyze the features of the networks they yield.

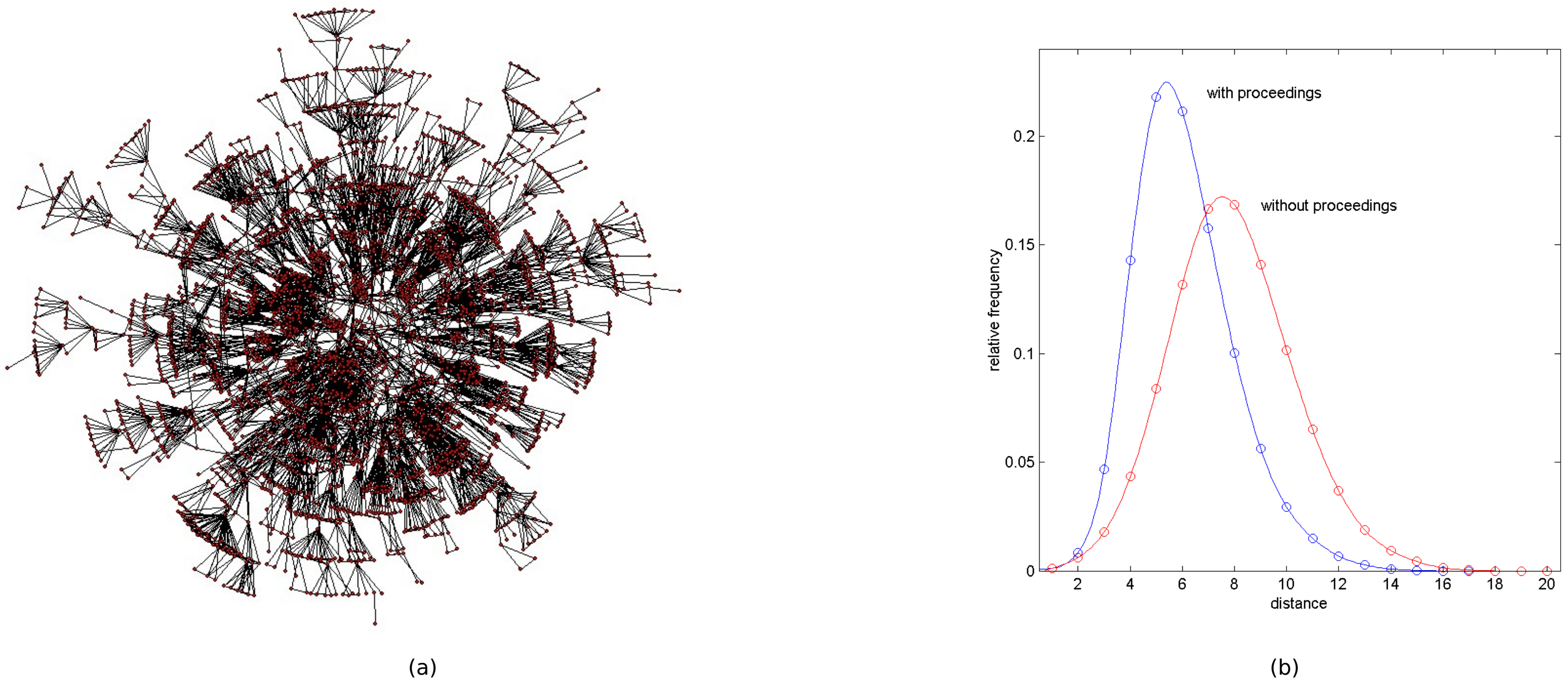


Figure 3: (a)Graphical representation of the network after removing proceedings (cf. Figure 1c; notice the network core is clearly less compact). (b) Comparison of the distribution of author distances with and without proceedings. The solid lines are eye-guides.

With respect to using these rankings as a proxy for overall contribution to the field, several considerations should be made. Firstly, a person that has been in many institutions might be very well linked, while not having done significant contributions to the field. On the other hand, persons with highly relevant papers or books but without a significant number of collaborators will not become sociometric stars, as might be the case with Dr. Koza. If anything, these rankings reflect the status of connectedness of the persons in the field, and their ability for team work. It certainly may be the case that social connections tend to inflate people's perception of scientific contributions. Then again, social connections do not just take the form of published papers, but can also substantiate in other scientific activities. In this sense, if the relevance of a researcher in the field can be somehow decomposed into objective and subjective components; objectively quantifiable measures such as those used in this work can be helpful to estimate this former part.

Our future lines of work along this topic will include the analysis of the network evolution through time, as well as the impact funded scientific networks and transnational grants (such as EU grants) have had on it. We also plan to study the existence of *invisible colleges* or communities within the EC field, and analyze which their axes of development are, e.g., topical or regional.

Acknowledgements

This work has been funded in part by projects TIC2003-09481-C04-01 and TIN2005-08818-C04-01 of the Spanish Ministry of Science and Technology. The Ucinet software [3] has been used in some stages of our analysis.

Bibliography

[1] A.-L. Barabási, H. Jeong, R. Ravasz, Z. Neda, T. Vicsek, and A. Schubert. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.

[2] Albert-Laszlo Barabási and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

[3] S.P. Borgatti, M.G. Everett, and L.C. Freeman. *Ucinet for Windows: Software for Social Network Analysis*. Analytic Technologies, Harvard, MA, 2002.

[4] L. Freeman. A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41, 1977.

[5] H. Kretschmer. Patterns of behaviour in coauthorship networks of invisible colleges. *Scientometrics*, 40(3):579–591, 1997.

[6] A.J. Lotka. The frequency distribution of scientific production. *J. Wash. Acad. Sci.*, 16:317–323, 1926.

[7] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[8] M.E.J. Newman. Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E*, 64(1):016131, 2001.

[9] M.E.J. Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132, 2001.

[10] M.E.J. Newman. Who is the best connected scientist? A study of scientific coauthorship networks. In E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors, *Complex Networks*, pages 337–370. Springer-Verlag, Berlin, 2004.

[11] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, Cambridge, 1994.

[12] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442, 1998.

About the authors

Carlos Cotta received the M.Sc. and Ph.D. degrees in 1994 and 1998, respectively, in Computer Science from the University of Málaga (Spain). He is Associate Professor at the Department of Computer Science of the University of Málaga since 2001. His research interests are primarily in evolutionary algorithms, especially memetic algorithms, and cross-paradigm hybridization. He is also interested in combinatorial optimization, and in bioinformatics.

Homepage: www.lcc.uma.es/~ccottap/
Email: ccottap@lcc.uma.es

Juan-Julián Merelo Guervó got a degree in theoretical physics from the University of Granada in 1988 and a PhD in Physics in 1994. He got interested in evolutionary algorithms more or less at the same time, having published papers on evolutionary optimization of neural networks and an evolutionary algorithm that plays MasterMind. He is currently an Associate Professor at the Computer Architecture Department of the University of Granada.

Homepage: geneura.ugr.es/~jmerelo/
Email: jj@merelo.net

The Electric Sheep

Scott Draves, Spotworks, San Francisco, spot@draves.org

Electric Sheep is a distributed screen-saver that harnesses idle computers into a render farm with the purpose of animating and evolving artificial life-forms known as *sheep*. The votes of the users form the basis for the fitness function for a genetic algorithm on a space of abstract animations. Users also may design sheep by hand for inclusion in the gene pool.

Dreams in High Fidelity is a new manifestation of the sheep. Physically it consists of a small computer driving a framed liquid crystal display. It plays sheep selected, edited, and sequenced by the artist at triple the resolution and more stately motion than the screen-saver.

1 Introduction

Electric Sheep [6] was inspired by SETI@Home [1] and has a similar design. Both are distributed systems with client/server architecture where the client is a screen-saver installable on an ordinary PC. Electric Sheep distributes the rendering of animations, each one 128 frames long and known as a sheep.

Besides rendering frames, the client also downloads completed sheep and displays them to the user. The user may vote for the currently displayed sheep by pressing the up arrow key. Each sheep is specified by a genetic code comprised of about 240 floating-point numbers. The codes are generated by the server according to a genetic algorithm where the fitness is determined by the collective votes of the users. This is a form of aesthetic evolution, a concept first realized by Karl Sims [9].

All the software is open source and users may participate in the network freely and anonymously. Anyone can download and install it. When their computer is idle and goes to sleep, the sheep animations appear, and in parallel the computer goes to work creating new sheep and sharing its results with all other users.

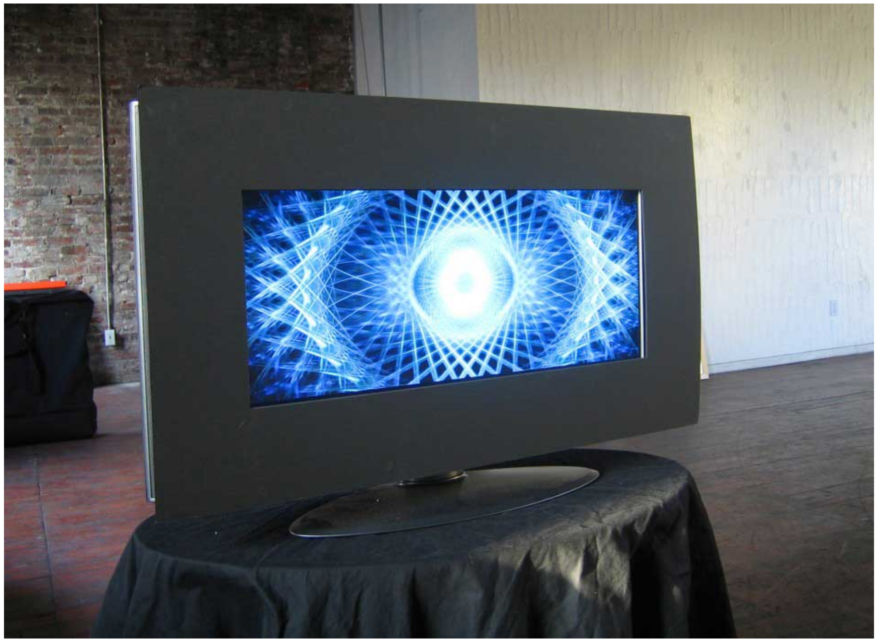
One can download additional software to manually design genomes and post them to the server where they join the flock [10]. These human designers collaborate and compete with the genetic algorithm (GA). The voting on 6000 sheep from five months of 2005 was about equally divided between human and GA designed sheep, more precisely the *creative amplification factor* of the GA was measured at 2.1 [8]. See Figure 1 for two fragments of families of sheep from this time.

2 Dreams in High Fidelity

Physically *Dreams In High Fidelity* consists of a small computer driving a large 1280×720 liquid crystal display. The computer creates a continuously morphing, non-repeating, abstract animation.

The animations are rendered by the Electric Sheep, but at triple the resolution and six times more frames per sheep. The image quality is striking on a large display. It requires 20 times the computation to make a high fidelity sheep.

The artist selects his favorite sheep from the archives and public flock, and sends them back to be re-rendered at high fidelity: heaven for an electric sheep. This new flock has grown to 50GB, totaling 8 hours if played end-to-end, and requiring over 1 million CPU hours to render.



Each *Dreams* made has a slightly different flock and includes sheep unique to it. Each flock resonates differently on playback so its sheep have a unique frequency distribution. The artist tunes each flock by adding and removing transitions between sheep to balance the distribution.

The *Dreams* has a symbiotic relationship to the free screen-saver. The free version provides the design laboratory and gene pool from which the best sheep are extracted. It also provides the distributed supercomputer needed to realize the high fidelity content. To complement, the hifi version brings the income required to keep the whole project in operation, and develop it further.

3 The Genetic Code

Each image produced by Electric Sheep is a fractal flame [5], a generalization and refinement of the Iterated Function System (IFS) category of fractals [3]. The genetic code used by Electric Sheep is just the parameter set for these fractals. It consists of about 240 floating-point numbers.

A classic IFS consists of a recursive set-equation on the plane:

$$S = \bigcup_{i=0}^{n-1} F_i(S)$$

The solution S is a subset of the plane (and hence a two-tone image). The F_i are a small collection of n affine transforms of the plane.

The equation is normally solved with the chaos game, which generates a large set of particles (610M for the screen-saver and 2.4G for *Dreams*) by starting anywhere then iteratively picking i at random and applying F_i.

A fractal flame is based on the same recursive equation, but the transforms may be non-linear and the solution algorithm produces a full-color image.

The transforms are linear blends of a set of 31 basis functions known as *variations*. The variations are composed with an affine matrix, like in classic IFS. So each transform F_i is:

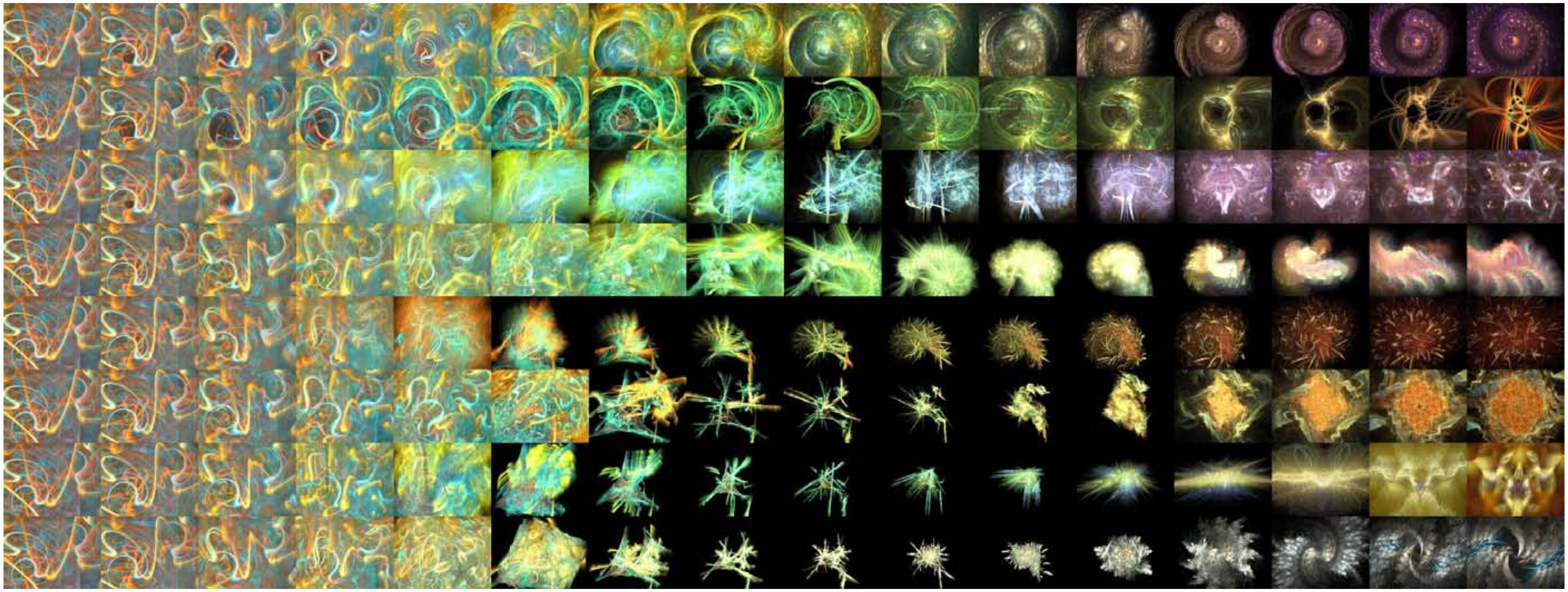
$$F_i(x,y) = \sum_j v_{ij} V_j(a_i x + b_i y + c_i, d_i x + e_i y + f_i)$$

where v_{ij} are the 31 blending coefficients for F_i, and a_i through f_i are 6 affine matrix coefficients. The V_j are the variations, here is a partial list:

$$\begin{aligned}
V_0(x,y) &= (x,y) \\
V_1(x,y) &= (\sin x, \sin y) \\
V_2(x,y) &= (x/r^2, y/r^2) \\
V_3(x,y) &= (r\cos(\theta+r), r\sin(\theta+r)) \\
V_4(x,y) &= (r\cos(2\theta), r\sin(2\theta)) \\
V_5(x,y) &= (\theta/\pi, r-1)
\end{aligned}$$

where r and θ are the polar coordinates for the point (x,y) in rectangular coordinates. V_0 is the identity function so this space of non-linear functions is a superset of the space of linear functions. See [7] for the complete list.

There are 3 additional parameters: color, and weight and symmetry, not covered here. Together these 40 (31 for v_{ij} plus 6 for a_i to f_i plus 3 is 40 total) parameters make up one transform, and are roughly equivalent to a gene in biological genetics. The order of the transforms in the genome does not effect the solution image. Many transforms have visually identifiable effects on the solution, for example particular shapes, structures, angles, or locations.



Normally there are up to 6 transforms in the function system, making for 240 (6×40) floating-point numbers in the genome. Note however that most sheep have most variational coeffients set to zero, which reduces the effective dimensionality of the space. On the other hand, many sheep have up to 12 transforms.

Animation and Transitions

The previous section described how a single image rather than an animation is defined by the genome. To create animations, Electric Sheep rotates over time the 2×2 matrix part (a_i, b_i, d_i, and e_i) of each of the transforms. After a full circle, the solution image returns to the first frame, so sheep animations loop smoothly.

The client does not just cut from one looping animation to another. It displays a continuously morphing sequence. To do this the system renders transitions between sheep in addition to the sheep themselves. The transitions are genetic crossfades based on pair-wise linear interpolation, but using a spline to maintain C^1 continuity with the endpoints.

4 Empirical Results

The primary dataset was collected from the server's database starting May 13th 2004 until October 13th, 153 days later. May 13th is when version 2.5 became operational. Previous versions of the server did not keep a record of the sheep: when they died they were completely deleted from the server. The data we have collected are a starting point to understanding the system and its behavior. However, they are somewhat confounded:

- The client uses the ratings to prioritize downloading. Since the server is busy enough that most clients cannot download all the sheep, this causes a snowball effect where a high rating itself causes more votes.
- The audience is fickle: sheep with identical genomes regularly receive completely different ratings (See Figure 6). Possibly the audience becomes fatigued by repeated exposure to variations of a successful genome, and stops voting for them. Even once popular sheep reintroduced much later do not necessarily fare well.

- Designers enlist their friends to vote for their sheep, post many similar sheep, or repost the results of automatic evolution. There are three administrators who occasionally kill sheep, explicitly direct mating, mutation, reincarnation, and vote without limit.

In April 2006 there are 30,000 users of the screen-saver almost everyday. About 900 of them vote by pushing the arrow keys on the keyboard, and 20 vote while browsing the database on the web. 14 genomes are submitted by 8 different designers. Over the six months of generation 198, 44 people submitted 5 or more genomes.

Amplification of Creativity

In a system with human-computer collaboration, the creative amplification is the ratio of total content divided by the human-created content. If we compare the posted genomes with their evolved descendents we can measure how much creative amplification Electric Sheep provides.

In the primary dataset there were 21% hand-designed, posted sheep and 79% evolved sheep. If the sum is weighted by rating, then we get 48% to 52%, for an amplification factor of 2.08 (1+52/48). One could say the genetic algorithm is doubling the output of the human posters.

Of the 79% evolved sheep, 42% of them result from the totally random genetic operator. Their fraction of total ratings is only 3.8%.

There are some caveats to this metric. For example, if the genetic algorithm just copied the posted genomes, it might receive some votes for its ‘creativity’. Or if it ignored the posted genomes and evolved on its own, it would receive some votes but they would not represent ‘amplification’.

Figure 1 shows the distribution of lengths of lineages of the sheep. The lineage length of a sheep is the maximum number of generations of children that issue from it. Sheep with no children are assigned one, and sheep with children are assigned one plus the maximum of the lineage lengths of those children. Instead of fitness increasing along lineage, we find it dying out: the rating of the average parent is 6.7 but the average maximum rating of direct siblings is only 3.8.

The decay in ratings may result from the audience losing interest in a lineage because it fails to change fast enough, rather than a decay of absolute quality of those sheep. The viewpoint of watching the screensaver and seeing sheep sequentially is different from the viewpoint of browsing the archive and comparing all the sheep. Neither can be called definitive.

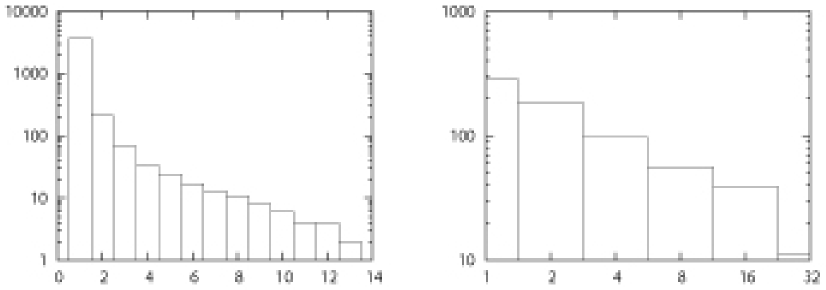


Figure 1: On the left is a histogram of lengths of lineages, and on the right is a histogram of the ratings of the sheep.

Genetic algorithms normally run for many tens to hundreds or thousands of generations. In contrast, the lineages (number of generations) of the sheep are very short: the longest is 13.

5 Motivation and Direction

Electric Sheep illustrates the process by which the longer and closer one studies something, the more detail and structure appears. It investigates the role of experiencers in creating the experience. If nobody ran the client, there would be nothing to see.

The sheep are parasites of human attention. Because the whole collective guides the evolution, no one user feels responsible for voting. Instead people respond to inspiration and their own initiative. The network is used to assemble these bits of judgement efficiently. This avoids the common pitfall of aesthetic evolution, which is a dearth of input.

Currently the limiting factor to the growth of the user and sheep populations is the server bandwidth. To deliver all the sheep to all the clients would require 20TB/day. The Coral CDN currently employed caps the sheep at 250GB/day. A client with BitTorrent [4] built-in is currently in beta and is delivering 150GB/day of sheep.

The goal of Electric Sheep is to create a self-supporting, network-resident life-form. Right now the sheep depend on a central server, requiring disk, bandwidth, and administration. Hopefully these inputs can be eliminated and the network can be made symmetric by using BitTorrent and Distributed Hash Tables [2];

The genetic algorithm is also dependent on the human designers. The highest rated sheep are human designed, and the children do not equal their parents. The most straightforward way to address this is to increase the population: more children are more chances for the lucky mutation. Hopefully decentralization will enable this too.

In the meantime we are experimenting with putting a human filter on the genetic algorithm, a technique we call the *brood*. The server now daily generates 128 potential children, but only renders one frame of each (this is about as expensive as rendering one ordinary sheep). The shepherd picks the best 10 or 20 of the brood. Future invocations of the genetic algorithm then use these picks, if available.

In general, how to combine human judgement and machine intelligence most efficiently remains an open and rewarding topic of research.

6 Conclusion

I look forward to many more generations of sheep at ever higher resolutions, with more expressive genetic codes, in three dimensions, responding to music, performing feats not yet imagined.

Bibliography

[1] David Anderson et al. Seti@home: An experiment in public-resource computing. *Communications of the ACM*, 45:56–61, 2002.

[2] Hari Balakrishnan, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Looking up data in p2p systems. *Communications of the ACM*, 46(2):43–48, 2003.

[3] Michael Barnsley. *Fractals Everywhere*. Academic Press, 1988.

[4] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[5] Scott Draves. The fractal flames.
Open source release on http://flam3.com, 1992.

[6] Scott Draves. The electric sheep screen-saver.
Open source release on http://electricsheep.org, 1999.

[7] Scott Draves. The fractal flame algorithm.
http://flam3.com/flame.pdf, 2004.

[8] Scott Draves. The electric sheep screen-saver: A case study in aesthetic evolution. In *Applications of Evolutionary Computing, LNCS 3449*. Springer Verlag, 2005.

[9] Karl Sims. Artificial evolution for computer graphics. In *Proceedings of SIGGRAPH*. ACM, 1991.

[10] Mark Townsend. Apophysis. http://apophysis.org, 2004.

About the author

Scott Draves a.k.a. Spot is a visualist and programmer residing in San Francisco. He is the creator of the Fractal Flame algorithm, the Bomb visual-musical instrument, and the Electric Sheep distributed screensaver. All of Draves' software artworks are released as open source and distributed for free on the internet. His award-winning work has appeared in Wired Magazine, the Prix Ars Electronica, the O'Reilly Emerging Technology Conference, and on the dance-floor at the Sonar festival in Barcelona. In 1997 Spot received a PhD in Computer Science from Carnegie Mellon University for a thesis on metaprogramming for media processing. He projects live video for underground parties and at clubs, and self-publishes SPOTWORKS, a DVD of abstract animation synchronized with electronic music. His current project is Dreams in High Fidelity, a painting that evolves.

Homepage: http://draves.org/
Email: spot@draves.org
Dreams in High Fidelity: http://hifidreams.com/

About the cover

Dream 165.25305, a sheep born on 2004.10.06 with a peak rating of 5 and living for just 36 hours.

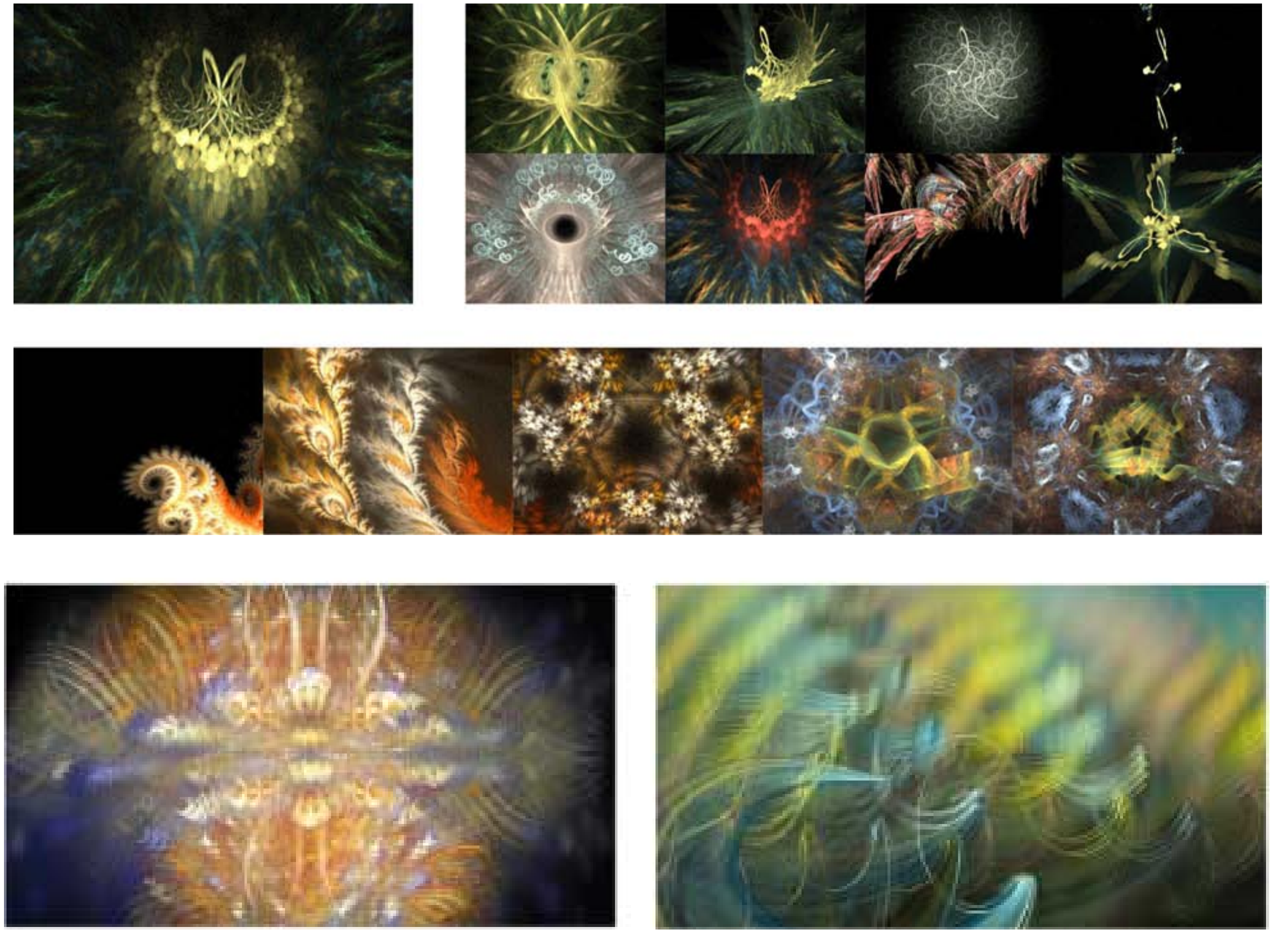
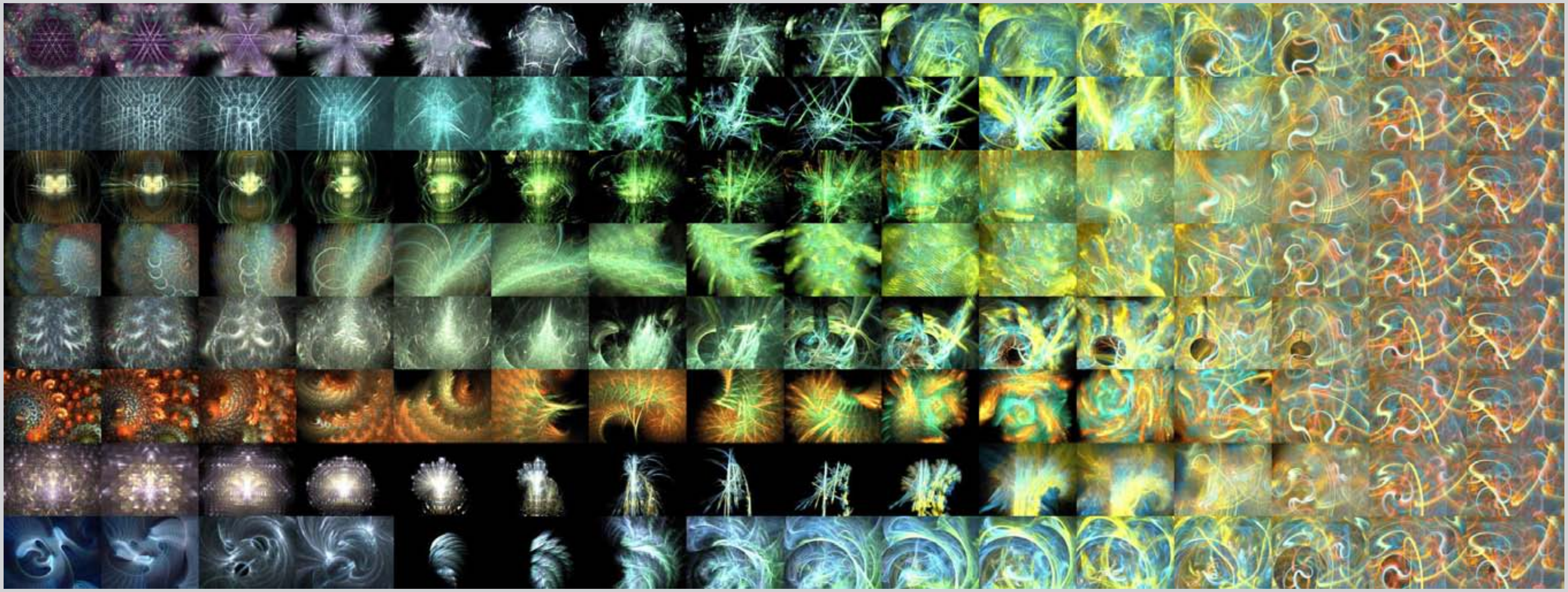


Figure 2: Sheep 15875 of generation 165, on the top-left, was born on August 16 and died 24 hours later after receiving one vote. It was one of 42 siblings. It was reincarnated on October 28 as sheep 29140, received a peak rating of 29, lived 7 days, and had 26 children, 8 of which appear to its right. In the middle are five generations of sheep in order starting on the left. Their numbers are 1751, 1903, 2313, 2772, and 2975. The last is a result of mutation, the previous three of crossover, the first was posted by etomchek. Bottom left is *Dream 191.21054* and bottom right is *Dream 198.19616*.



Eight possible transitions from a sheep to other sheep. Many transitions pass through a degenerate point where the image is mostly black. These bad frames have small file size in JPG format, so to avoid them, the rows are sorted by the minimum size in each row, and the top rows are more often used.

A Brief Survey of μGP

Massimiliano Schillaci, Politecnico di Torino, Italy, massimiliano.schillaci@polito.it
Edgar Ernesto Sánchez Sánchez, Politecnico di Torino, Italy, sanchez@evancon.polito.it

The first version of μGP has been proposed at the Congress of Evolutionary Computation in Honolulu (Hawaii) in 2002. Stemming from a simpler GA for optimizing parameters in test programs [1], μGP has undergone several revisions and enhancements, and is now a rather sophisticated tool that incorporates many key ideas from the theory of evolutionary computation [2, 3] as well as from extensive on-field experience [4, 5, 6].

μGP is an evolutionary approach to generic problem solving with an emphasis on assembly program generation for microprocessor test and verification. It employs a modular structure comprising three independent, but connected, blocks: an evolutionary core which cultivates a population of individuals, exploiting several evolutionary operators; an instruction library, used to map individuals to test programs or other representations; an external evaluator, which computes a fitness value for every individual, thus providing the evolutionary core with the necessary feedback.

A μGP individual is represented as a graph made up of loosely-connected subgraphs. Each node of the graph, except the first and last node of a subgraph, has a predecessor and a successor node, which determine the order of appearance in the external representation (program); additionally, every node can have references to any other node in its subgraph, or to the beginning of a different subgraph. This representation mimics the syntax of assembly programs. Every node can also contain parameters of several types: integer or hexadecimal numbers, symbolic constants, and real numbers. Whether or not a node contains parameters or references to other nodes, how many and of which type is specified in the instruction library: its role is to constrain the otherwise limitless variety of possible individuals, making them suited to the intended application.

The evolutionary core cultivates a population of individuals using a $(\mu + \lambda)$ evolutionary strategy, while starting the evolutionary process from an initial population of ν (nu) individuals. The evolutionary core is able to self-adapt several parameters, such as the activation probability and strength for each genetic operator, and the tournament size for parent selection. It implements two recombination operators: crossover and safe crossover (a special operator tuned for very small individuals). There are also five mutations: add (add a node from the graph); delete (remove a node from the graph); add/delete (a sequence of both add and remove operations); modify (change an internal node parameter); local mutation (slightly tune an internal parameter); and a search-mutation (exhaustively explore all possibilities for a node parameter).

The core is able to maintain population diversity by detecting clones, or identical individuals, scaling their fitness and optionally exterminating them. For the same purpose it also implements a fitness entropy hole [2]. It also allows to fine-tune its originally elitist scheme letting the user specify a maximum age for the individuals and a number of elite individuals (which do not age).

The instruction library instructs the evolutionary core about what constraints should be automatically enforced on individuals and also on how to map them to their external representation. The assembly language of a target microprocessor can be exploited, making use of specific syntactic constructs; different operator types can be used for different instructions, and different addressing modes can be used for the same or different instructions.

The third block of the evolutionary approach is an external evaluator that computes a fitness value for every evaluated individual. It is usually written in the form of a script that starts a simulator with the individual as input, collects the results and produces a numeric fitness following some predetermined metric. Being independent from both the evolutionary core and the instruction library it can be tuned to optimize individuals, even of the same type, for different goals.

It is worth noting that to write an instruction library for microprocessor test purposes the needed information can be found in the processor's user manual, since everything is specified in the ISA for the target machine. Thus an engineer needs only to write an instruction library, using information readily available in manuals, and a fitness function that specifies the goals and metrics.

Application

μGP has proved to be a versatile tool. It has been used for diverse activities, some of which very distant from the test and verification fields. Indeed, it has been successfully employed to evolve corewar programs, to cultivate functions as an ordinary GP, and even for antenna array optimization. Its modular structure has been extremely useful in all these activities.

Most activity has, however, been performed on the primary fields of microprocessor test and verification, with some activity on diagnosis. Some of the used processor models are reported in Table 1.

The models for all these processors are very different from each other, reflecting the different complexity of the machine. Simulatable gate-level models have been employed for the 68hc11, 8051 and Plasma (Mips I) processors; The Leon2 processor, implementing a SPARCv8 architecture, has been synthesized and mapped on a FPGA device; DLX has been tackled as a RT-level VHDL model; for the PPC750 a RT-level SystemC model has been exploited; finally, to effectively approach the high complexity of a Pentium® 4 the performance counters on an actual processor have been used to compute suitable metrics.

The activities different from processor test and verification have not only been challenging in themselves, but also useful to enhance the evolutionary core performance for applications more focused on CAD. Their details are reported in Table 2. An updated publication list about μGP can be found in [7].

Processor	Complexity	Activities
Mc68hc11	$\sim 10^5$ transistors	Verification
i8051	$\sim 10^5$ transistors	Test, verification, and diagnosis
DLX	$\sim 10^5$ transistors	Verification
Plasma	$\sim 10^5$ transistors	Test
Leon2	$\sim 10^5$ transistors	Test and verification
PowerPC	$\sim 10^6$ transistors	Test and verification
P4	$\sim 10^7$ transistors	Post-silicon verification

Table 1: Tackled processors ordered by increasing complexity

Topic	Activities
Corewar	Evolution of warriors for winning corewars competitions
Function building	Evolution of a function for maximal set covering
Antenna array	Optimization of an antenna array factor

Table 2: Other applications.

μGP Corewar Tutorial and Demo

To start an evolutionary experiment with μGP two main components are needed, besides the actual μGP executable: an instruction library and a fitness script. Often an external tool will also be necessary, and the fitness script will just be a stub to launch it and collect its results. In the following a brief example on evolving corewars programs (warriors) will be exposed. It will be assumed that a Unix/Linux environment is used.

Redcode, the assembly-like language of corewars, is totally orthogonal, so the instruction library could be made very simple. For illustration purposes, however, an instruction library longer than necessary will be used below.

```
.commentformat  "; %s\n"
.labelformat    "%s:\n"

.type INT integer -799 799
.type OFFSET integer -20 20
.type INSTRUCTION_MODIFIERS constant a b ab ba f x i
.type ADDRESSING_MODES constant # $$ @ * { } < >
.type OPCODE constant jmp spl mov add sub nop dat seq sne slt

.globalprologue
;assert CORESIZE == 800\n
.endglobalprologue

.globalepilogue
.endglobalepilogue

.section MAIN
.prologue
        org $1
.parameter inner_forward_label
.endprologue

.epilogue
end
.endepilogue

.macro
.probability 1
        mov.$1 $2$3+$4, $5$6+$7
.parameter type INSTRUCTION_MODIFIERS
.parameter type ADDRESSING_MODES
.parameter inner\_generic_label
.parameter type OFFSET
.parameter type ADDRESSING_MODES
.parameter inner_generic_label
.parameter type OFFSET
.endmacro

.macro
.probability 13
        $1.$2 $3$4, $5$6
.parameter type OPCODE
.parameter type INSTRUCTION_MODIFIERS
.parameter type ADDRESSING_MODES
.parameter type OFFSET
.parameter type ADDRESSING_MODES
.parameter type OFFSET
.endmacro
```

```
.macro
.probability 1
        mov.$1 $2$3, $4$5
        djn.$6 $2$3, $4$5
.parameter type INSTRUCTION_MODIFIERS
.parameter type ADDRESSING_MODES
.parameter type OFFSET
.parameter type ADDRESSING_MODES
.parameter type OFFSET
.parameter type INSTRUCTION_MODIFIERS
.endmacro
```

In the beginning the format of the labels and the comments is defined, following the same syntax as the format string of the C printf() function. Then some types are defined, as they would be in a preprocessing language.

Every macro specifies a way to generate a piece of code. Not every macro is necessarily used, and a single macro may originate several code fragments.

The above instruction library is composed of three different macros, each of which originates different pieces of code. The first macro generates code lines such as “`mov.a *n12+4, >n15-30`,” in which the opcode (`mov`) is fixed, while the rest of the line may vary as specified in the type definitions; the “`n12`” and “`n15`” are label references: some other code lines, before or after the one shown, will be preceded by those labels. The second macro makes also the opcode a parameter (indeed, in this macro only the dot, comma and spaces are fixed), originating code lines such as “`add.f {40, @-3`.” In these lines the label references are not present. The third macro originates couple of code lines, such as “`mov.a <35, }5`” and “djn.x <35, }5,” in which the two opcodes are fixed, the instruction modifiers are chosen independently, but the operands are the same for both instructions. Please notice that the parameter number relates with the order in which they are defined, not the order in which they appear in the macro.

Every macro is assigned a probability (actually a statistical weight) that specifies how often it may be chosen to be included into an individual. With the current assignment, on average every 150 code fragments 10 will be generated using the first macro, 130 using the second and the last 10 using the third.

To give the evolutionary core the necessary feedback a fitness script has to be prepared. Only the general scheme for such a script will be given, as the actual code would take too much space. In this case the fitness script acts as a stub to the actual redcode virtual machine inside which the programs execute and fight to remain the only executing program in the machine.

In previous corewars experiments programs such as pmars or exhaust have been used; these execute two warriors and show the number of wins and ties for the first one, allowing to compute a score for both.

The easiest performance measure that can be obtained from an evolved warrior is its score against a fixed set of known-strong warriors. A possible scheme for a fitness script is then the following:

```
fitness(warrior w, set<warrior> reference){
  if (checkWarrior(w) == ValidWarrior){
    score := 0;
    for (r in reference){
      performMatch(w, r)
      score := score + 3 * wins + ties
    }
  }
  write score in “fitness.out”
}
```

“fitness.out” is the default name for the file that contains the result of the fitness script. Note that currently the fitness script has no way to know at run-time what this name should be, and must use one decided at compile time. A different name can be used if required.

Actual code has been developed in perl to match a generated warrior against a set of existing adversaries at every possible position in memory to avoid biasing or random effects; the result of the match is captured from the virtual machine’s standard output.

Once the instruction library and the fitness script are ready the experiment can be started. The recommended practice is to put the files pertaining to a single experiment (IL, fitness script and other related material, but not the executables) in a dedicated directory; a single evolutionary run can produce many different files, which should be kept together.

The μGP executable recognizes many command-line switches to set its parameters, evolutionary and run-time. The main ones are: population size, number of genetic operator applications, initial size of individuals, number of generations, name of generated individuals, fitness script, instruction library, fitness result file. To start a first experiment with corewars a command line such as

```
ugp -i 10 -g 1000 -N 1 -f warrior.red -p ./fitness.pl
```

will be enough, provided that the μGP and other executables are in the search path and both the IL and fitness script are in the current directory.

This will start a run using the default parameters of 30 individuals in the population, 20 genetic operator applications per generation, IL and result file, and specifying an individual initial size of 10 macros, a run length of 1000 generations, “warrior.red” as the warrior name and “fitness.pl” as fitness script (assuming a perl script), and a number of results (fitness values) per individual equal to 1 (default is 2).

Current Development

An effort is currently under way to generalize the approach and extend its applicability to problems not necessarily related to the CAD field. To this end, the evolutionary core is being rewritten in C++, further emphasizing its modular structure and enhancing upgradability. In its new form the tool will allow easy experimentation of new and different evolutionary strategies which would now require significant coding.

Bibliography

[1] F. Corno, M. Sonza Reorda, G. Squillero, M. Violante, *On the Test of Microprocessor IP Cores*, DATE2001: IEEE Design, Automation & Test in Europe Conference, 2001, pp. 209-213

[2] R. Poli, em A simple but theoretically-motivated method to control bloat in genetic programming, EuroGP 2003, pp. 204-217

[3] J. Koza, *Genetic Programming: on the programming of computers by means of natural selection*, MIT press, 1992

[4] E. Sanchez, G. Squillero, M. Sonza Reorda, *Automatic Completion and Refinement of Verification Sets for Microprocessor Cores*, Lecture Notes in Computer Science, Vol 3449, “Applications on Evolutionary Computing: EvoWorkshops 2005”, Lausanne (CH), March 2005, pp. 205-214

[5] F. Corno, E. Sanchez, G. Squillero, *“On The Evolution of Corewar Warriors”*, CEC2004, Congress on Evolutionary Computation, Portland (Oregon), June 20-23, 2004

[6] E. Sanchez, M. Schillaci, M. Sonza Reorda, G. Squillero, L. Sterpone, M. Violante, “New Evolutionary Techniques for Test-Program Generation for Complex Microprocessor Cores”, GECCO05: Genetic and Evolutionary Computation Conference, Washington, DC, USA, June 25-29 2005, pp. 2193-2194

[7] μGP homepage, [WWW]

About the authors

Massimiliano Schillaci is a Ph.D. student at the Politecnico di Torino. His research interests include test techniques for advanced processors and evolutionary algorithms. Massimiliano Schillaci has a degree in electronic engineering from Politecnico di Torino.

Homepage: http://www.cad.polito.it/~schillac/
Email: massimiliano.schillaci@polito.it

Edgar Ernesto Sánchez Sánchez is a Post-doctoral researcher at the Politecnico di Torino. His research interests include test techniques for advanced processors and evolutionary algorithms. Sánchez has a degree in electronic engineering from Universidad Javeriana, Colombia.

Homepage: http://www.cad.polito.it/~sanchez/
Email: edgar.sanchez@polito.it

For additional information about uGP, please visit [WWW].

GECCO-2006 Highlights

GECCO General Schedule

July 8-9	8:30 - 18:00	Free Workshops and Tutorials
July 9	19:00 - 22:00	Opening Reception.
July 10-11	8:30 - 18:00	Paper Presentations
July 11	19:00 - 22:00	Poster Session (Reception)
July 12	8:30 – 15:40	Paper Presentations

Full schedule available at [WWW].

Human Competitive Results

There will be a special presentation of 8 entries for $10,000 in Human Competitive Awards at the GECCO-2006 conference in Seattle on July 10 at 1:45-3:45 PM. Copies of the 8 entries may be viewed at www.human-competitive.org.

Workshops at GECCO-2006

- Medical Applications of Genetic and Evolutionary Computation Workshop (MedGEC) [WWW]
- Biological applications of Genetic and Evolutionary Computation (BioGEC) [WWW]
- User-centric Evolutionary Computation Workshop [WWW]
- Optimization by Building and Using Probabilistic Models [WWW]
- Military and Security Applications of EC [WWW]
- Undergraduate Student Workshop [WWW]
- Learning Classifier Systems (LCS) [WWW]
- Adaptive Representations [WWW]
- Complexity through Development and Self-Organizing Representations (CODESOAR) [WWW]

Papers Nominated for Best Paper Awards

Ant Colony Optimization and Swarm Intelligence

An Ant-Based Algorithm for Finding Degree-Constrained Minimum Spanning Tree
Thang N. Bui (Penn State Harrisburg)
Catherine M. Zrncic (Penn State Harrisburg)

Particle Swarm with Speciation and Adaptation in a Dynamic Environment
Xiaodong Li (School of Computer Science and IT, RMIT University)
Juergen Branke (Institute AIFB, University of Karlsruhe)
Tim Blackwell (Goldsmiths College, University of London)

PSO and Multi-Funnel Landscapes: How cooperation might limit exploration
Andrew M. Sutton (Colorado State University)
Darrell Whitley (Colorado State University)
Monte Lunacek (Colorado State University)
Adele Howe (Colorado State University)

Artificial Immune Systems

Properties of the Bersini Experiment on Self-Assertion
Werner Dilger (Chemnitz University of Technology)
Steve Strangfeld (Chemnitz University of Technology)

Applicability Issues of the Real-Valued Negative Selection Algorithms
Zhou Ji (St. Jude Children’s Research Hospital)
Dipankar Dasgupta (The University of Memphis)

Artificial Life, Evolutionary Robotics, Adaptive Behavior

Facilitating Neural Dynamics for Delay Compensation and Prediction in Evolutionary Neural Networks
Heejin Lim (Texas A&M University)
Yoonsuck Choe (Texas A&M University)

A Method for Parameter Calibration and Relevance Estimation in Evolutionary Algorithms
Volker Nannen (Vrije Universiteit Amsterdam)
A.E. Eiben (Vrije Universiteit Amsterdam)

Modular Thinking: Evolving Modular Neural Networks for Visual Guidance of Agents
Ehud Schlessinger (Ins. of Ophthalmology, University College London)
Peter J. Bentley (Dept. of Computer Science, University College London)
R. Beau Lotto (Ins. of Ophthalmology, University College London)

Biological Applications

Genetic Programming for Human Oral Bioavailability of Drugs
Francesco Archetti (University of Milano-Bicocca)
Stefano Lanzeni (University of Milano-Bicocca)
Enza Messina (University of Milano-Bicocca)
Leonardo Vanneschi (University of Milano-Bicocca)

Comparing Mathematical Models on the Problem of Network Inference
Christian Spieth (Centre for Bioinformatics)
Nadine Hassis (Centre for Bioinformatics)
Felix Streichert (Centre for Bioinformatics)
Jochen Supper (Centre for Bioinformatics)
Nora Speer (Centre for Bioinformatics)
Klaus Beyreuther (Centre for Bioinformatics)
Andreas Zell (Centre for Bioinformatics)

Coevolution

The Parallel Nash Memory for Asymmetric Games
Frans A Oliehoek (University of Amsterdam)
Edwin D de Jong (Utrecht University)
Nikos Vlassis (University of Amsterdam)

Archive-based Cooperative Coevolutionary Algorithms
Liviu Panait (George Mason University)
Sean Luke (George Mason University)
Joseph F. Harrison (George Mason University)

The Effects of Interaction Frequency on the Optimization Performance of Cooperative Coevolution
Elena Popovici (George Mason University)
Kenneth De Jong (George Mason University)

Robustness in Cooperative Coevolution
R. Paul Wiegand (US Naval Research Laboratory)
Mitchell A. Potter (US Naval Research Laboratory)

Estimation of Distribution Algorithms

Probabilistic Modeling for Continuous EDA with Boltzmann Selection and Kullback-Leibeler Divergence
Yunpeng Cai (Tsinghua University)
Xiaomin Sun (Tsinghua University)
Peifa Jia (Tsinghua University)

Evolution Strategies, Evolutionary Programming

Hierarchically Organised Evolution Strategies on the Parabolic Ridge
Dirk V. Arnold (Dalhousie University)
Alexander MacLeod (Dalhousie University)

Reconsidering the Progress Rate Theory for Evolution Strategies in Finite Dimensions
Anne Auger (CoLab ETH Zurich)
Nikolaus Hansen (CoLab ETH Zurich)

A Computational Efficient Covariance Matrix Update and a (1+1)-CMA for Evolution Strategies
Christian Igel (Ruhr-Uni. Bochum)
Thorsten Suttorp (Ruhr-Uni. Bochum)
Nikolaus Hansen (Swiss Federal Institute of Technology (ETH) Zurich)

On the Local Performance of Simulated Annealing and the (1+1) Evolutionary Algorithm
Thomas Jansen (Universität Dortmund)
Ingo Wegener (Universität Dortmund)

Evolutionary Combinatorial Optimization

A New Hybrid Evolutionary Algorithm for the k-cardinality Tree Problem
Christian Blum (ALBCOM, LSI, Universitat Politècnica de Catalunya)

Anisotropic Selection in Cellular Genetic Algorithms
David Simoncini (Universite Nice Sophia Antipolis)
Sébastien Verel (Universite Nice Sophia Antipolis)
Philippe Collard (Universite Nice Sophia Antipolis)
Manuel Clergue (Universite Nice Sophia Antipolis)

Adaptation for Parallel Memetic Algorithm Based on Population Entropy
Jing Tang (Nanyang Technological University)
Meng Hiot Lim (Nanyang Technological University)
Yew Soon Ong (Nanyang Technological University)

Evolutionary Multiobjective Optimization

Combining Gradient Techniques for Numerical Multi-Objective Evolutionary Optimization
Peter A.N. Bosman (Centre for Mathematics and Computer Science)
Edwin D. de Jong (Utrecht University)

Towards Estimating Nadir Objective Vector Using Evolutionary Approaches
Kalyanmoy Deb (IIT Kanpur)
Shamik Chaudhuri (IIT Kanpur)
Kaisa Miettinen (Helsinki School of Economics)

On The Effect of Populations in Evolutionary Multi-objective Optimization
Oliver Giel (Universität Dortmund)
Per Kristian Lehre (Norwegian University of Science and Technology)

Inside a Predator-Prey Model for Multi-Objective Optimization: A Second Study
Christian Grimme (University of Dortmund)
Karlheinz Schmitt (University of Dortmund)

Evolvable Hardware

Filter Approximation Using Explicit Time and Frequency Domain Specifications
Varun Aggarwal (CSAIL, MIT)
Wesley O. Jim (CSAIIL, MIT)
Una-May O'Reilly (CSAIL, MIT)

Genetic Algorithms

Comparing Evolutionary and Temporal Difference Methods in a Reinforcement Learning Domain
Matthew E. Taylor (The University of Texas at Austin)
Shimon Whiteson (The University of Texas at Austin)
Peter Stone (The University of Texas at Austin)

Dynamic Multi-Objective Optimization with Evolutionary Algorithms: A Forward-Looking Approach
Iason Hatzakis (Massachusetts Institute of Technology)
David R Wallace (Massachusetts Institute of Technology)

Non-Wrapping Order Crossover: An Order Preserving Crossover Operator
that Respects Absolute Position
Vincent A Cicirello (The Richard Stockton College of New Jersey)

Multi-Objective Test Problems, Linkages, and Evolutionary Methodologies
Kalyanmoy Deb (IIT Kanpur)
Ankur Sinha (IIT Kanpur)
Saku Kukkonen (Lappeenranta University of Technology)

An Empirical Investigation of How and Why Neutrality Affects Evolutionary Search
Edgar Galvan-Lopez (University of Essex)
Riccardo Poli (University of Essex)

Genetic Programming

Characterizing the Dynamics of Symmetry Breaking in Genetic Programming
Jason M. Daida (The University of Michigan)

Using Contextaware Crossover to Improve the Performance of GP
Hammad Majeed (University of Limerick)
Conor Ryan (University of Limerick)

Dynamics of Evolutionary Robustness
Alan T Piszcz (University of Idaho)
Terence Soule (University of Idaho)

Alternative Evolutionary Algorithms for Evolving Programs
Darrell Whitley (Colorado State University)
Ross Beveridge (Colorado State University)
Marc Richards (Colorado State University)
Andre Barreto (Universidade Federal do Rio de Janeiro)

LCS and other GBML

Classifier Prediction based on Tile Coding
Pier Luca Lanzi (Politecnico di Milano)
Daniele Loiacono (Politecnico di Milano)
Stewart W Wilson (Prediction Dynamics)
David E Goldberg (University of Illinois at Urbana Champaign)

A Bayesian Approach to Learning Classifier Systems in Uncertain Environments
Davide Aliprandi (Politecnico di Milano)
Alex Mancastroppa (Politecnico di Milano)
Matteo Matteucci (Politecnico di Milano)
Andrea Bonarini (Politecnico di Milano)

Bounding XCS's Parameters for Unbalanced Datasets
Albert Orriols-Puig (Universitat Ramon Llull)
Ester Bernadó-Mansilla (Universitat Ramon Llull)

Real-World Applications

Automating the Drug Scheduling with Different Toxicity Clearance in Cancer Chemotherapy via Evolutionary Computation
Yong Liang (The Chinese University of Hong Kong)
Kwong-Sak Lueng (The Chinese University of Hong Kong)
Tony Shu Kam Mok (The Chinese University of Hong Kong)

Multiobjective Genetic Algorithms for Multiscaling Excited State Direct Dynamics in Photochemistry
Kumara Sastry (University of Illinois at Urbana-Champaign)
D. D. Johnson (University of Illinois at Urbana-Champaign)
Alexis L. Thompson (University of Illinois at Urbana-Champaign)
David E. Goldberg (University of Illinois at Urbana-Champaign)
Todd J. Martinez (University of Illinois at Urbana-Champaign)
Jeff Leiding (University of Illinois at Urbana-Champaign)
Jane Owens (University of Illinois at Urbana-Champaign)

The Complete-Basis-Functions Parameterization in ES and its Application to Laser Pulse Shaping
Ofer M. Shir (Leiden Institute of Advanced Computer Science)
Christian Siedschlag (FOM-Instituut AMOLF)
Thomas Bäck (Leiden Institute of Advanced Computer Science)
Marc J. J. Vrakking (FOM-Instituut AMOLF)

Search-Based Software Engineering

Clustering the Heap in Multi-Threaded Applications for Improved Garbage Collection
Myra B. Cohen (University of Nebraska-Lincoln)
Shiu Beng Kooi (University of Nebraska-Lincoln)
Witawas Srisa-an (University of Nebraska-Lincoln)

Search-based Determination of Refactorings for Improving the Class Structure of Object-Oriented Systems
Olaf Seng (FZI Forschungszentrum Informatik)
Johannes Stammel (FZI Forschungszentrum Informatik)
David Burkhart (FZI Forsch)

NCSA/IlliGAL Gathering on Evolutionary Learning 2006 (NIGEL 2006)

Xavier Llorà
Automated Learning Group & Illinois Genetic Algorithms Laboratory
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
1205 W. Clark Street, Urbana, IL 61801, USA
xllora@illigal.ge.uiuc.edu

On May 16th and 17th, a group formed by more than twenty researchers got together in Urbana-Champaign (Illlinois) to participate in the gathering on evolutionary learning organized by the National Center for Supercomputer Applications and the Illinois Genetic Algorithms Laboratory (NIGEL 2006). The goals were to discus current state-of-the-art research in learning classifier systems and other genetics-based machine learning, and to identify future research trends and applications where evolutionary learning might provide a competitive advantage.

The gathering lasted for two days. The first one was dedicated to public presentations of the current research efforts of the participants. These presentations also included new research topics and problems, challenges to the evolutionary learning community, and retrospective looks to the origins of the field and the lessons they could teach us. The second day of the gathering, on the other hand, was reserved for hands-on discussions.

We conducted three small workshops that took the form of quick brainstorming sessions. During the first one we brainstormed about the current status of learning classifier systems and genetics-based machine learning, and what should be the areas we should concentrate on.

The second one was targeted to identify new areas of application of evolutionary learning techniques. The last workshop focused on new techniques and trends to keep improving and advancing the understanding of the current techniques.

To help disseminate the discussions of the NIGEL and using NCSA facilities, we archived the videos, slides, and photos of the gathering on a freely available website. The website can be accessed at

http://www.ncsa.uiuc.edu/Conferences/NIGEL

List of participants: *Loretta Auvil, Jaume Bacardit, Alwyn Barry, Lashon Booker, Ester Bernadó, Will Browne, Martin Butz, Jorge Casillas, Helen Dam, Dipankar Dasgupta, Deon Garrett, David E. Goldberg, Noriko Imafuji, Pier Luca Lanzi, Xavier Llorà, Kumara Sastry, Kamran Shafi, Kenneth Turvey, Michael Welge, Ashley Williams, Stewart Wilson, and Paul Winward.*

Pygene Open Source Evolutionary Computation Tool

The purpose of this letter is to inform those in the evolutionary computation community of the existence of “Pygene” and to offer a suggestion as to a possible expansion of the tool to cover quantum computing simulation. To those unfamiliar with this tool, "Pygene", is an open source, evolutionary computation engine written in Python. Pygene can be downloaded free of charge from:

http://www.freenet.org.nz/python/pygene/

Pygene supports the following features (listed from the Pygene website):

- Simple, comfortable, pythonic classes
- Several simple working examples, including: Quadratic equation solver, Two-dimensional convergence, String cracker, and Traveling Salesman Problem
- Support for genetic programming
- The genetic algorithms classes are based on classical Mendelian genetics
- Fully customizable recombination, mutation, inheritance, natural selection
- A rich set of primitive Gene classes to use as-is, or subclass
- Hermaphroditic sexual reproduction
- Configurable ’incest’ option (high-fitness parents stay in the population and mate with children)
- Configurable spontaneous appearance of random new organisms
- Ability to dump Populations, Organisms and/or Genes out to an XML file or string.

In order to use Pygene you must download the package listed at the web site above. You must also have python 2.3 or later. If you want to run the GUI version of the Traveling Salesman Problem, you’ll need to install fltk (fltk.sf.net) and pyfltk (pyfltk.sf.net).

I have found Pygene to be extremely useful. It has saved many hours of development. Many of the reproduction parameters are clearly identifiable in the code and easy to manipulate. The Pygene library contains several sample problems written in Python which can be used as demonstrations or as a set of building blocks for more complex algorithms.

I would like to make the following suggestion to the creator’s of “Pygene", and the evolutionary computation community as a whole. As a researcher in the quantum information field, it would greatly assist my efforts if Pygene versions of the Fredkin, Toffoli, and Walsh-Hadamard Transformations gates were freely available as open source Pygene modules, or any open source evolutionary package for that mater. I believe that the development of quantum information systems may be greatly enhanced by evolutionary circuit design techniques. It would serve the quantum information research community well if we had a set of “Open Source Quantum Circuits in Pygene" to perform the many experiments still required on the march towards the successful implementation of quantum computing.

Special thanks to David at the Freenet organization for granting permission to use the material at the Pygene website and for his efforts in creating a very useful tool. Requests for further information concerning Pygene or its future capabilities should be directed by email to David@freenet.org.nz.

Jack Lenahan
Imagine-One Corporation
Staff Scientist, Office of the Chief Engineer
Space and Naval Warfare Systems Center
Charleston, South Carolina
Email - jlena16682@aol.com

Forthcoming Papers

Evolutionary Computation, Volume 14, Number 3

- **An Analysis of Mutative σ-Self-Adaptation on Linear Fitness Functions**, Nikolaus Hansen, 255–276
- **A Step Forward in Studying the Compact Genetic Algorithm**, R. Rastegar and A. Hariri, 277–290
- **Optimum Tracking with Evolution Strategies**, Dirk V. Arnold and Hans-Georg Beyer, 291–344
- **A Comparison of Bloat Control Methods for Genetic Programming**, Sean Luke and Liviu Panait, 309–344
- **Automated Global Structure Extraction for Effective Local Building Block Processing in XCS**, Martin V. Butz, Martin Pelikan, Xavier Llorà, and David E. Goldberg, 345–380

Artificial Life Journal, Volume 12, Number 3

- **Distributed Coordination of Simulated Robots Based on Self-Organisation**, Gianluca Baldassarre, Domenico Parisi, and Stefano Nolfi
- **Embryonics: a Path to Artificial Life?**, Xuegong Zhang, Gabriel Dragffy, and Anthony G. Pipe
- **Axiomatic Scalable Neurocontroller Analysis Via The Shapley Value**, Alon Keinan, Ben Sandbank, Claus C. Hilgetag, Isaac Meilijson, and Eytan Ruppin
- **A Neural Learning Classifier System with Self-Adaptive Constructivism for Mobile Robot Control**, Jacob Hurst & Larry Bull
- **Evolution and Development of a Multi-Cellular Organism: Scalability, Resilience and Neutral Complexification**, Diego Federici & Keith Downing
- **Self-Replication and Self-Assembly for Manufacturing**, Robert Ewaschuk & Peter D. Turney
- **Neurocontroller Analysis via Evolutionary Network Minimization**, Zohar Ganon, Alon Keinan, and Eytan Ruppin

Artificial Life Journal, Volume 12, Number 4

- **Spatial Patterns Induced Evolution of Self-Replicating Loop Network**, Keisuke Suzuki & Takashi Ikegami
- **From Machine and Tape to Structure and Function: Formulation of a Reflexively Computing System**, Chris Salzberg
- **The Origin of the Eukaryotic Cell Based on Conservation of Existing Interfaces**, Albert D.G. de Roos
- **Through the Interaction of Neutral and Adaptive Mutations, Evolutionary Search Finds a Way**, Tina Yu & Julian Francis Miller
- **Genetic Stability and Territorial Structure Facilitate the Evolution of Tag-Mediated Altruism**, Lee Spector & Jon Klein
- **Evolving Physically Simulated Flying Creatures for Efficient Cruising**, Yoon-Sik Shim & Chang-Hun Kim
- **The Infection Algorithm: an Artificial Epidemic Approach for Dense Stereo Correspondence**, Gustavo Olague, Francisco Fernández, Cynthia B. Pérez, and Evelyne Lutton
- **Simulating the Hallmarks of Cancer**, Robert G. Abbott, Stephanie Forrest, and Kenneth J. Pienta

New Books

Series on Genetic and Evolutionary Computation

Series Editors: Goldberg, D.E., Koza, J.R.
Series Homepage: [WWW]

- *Linear Genetic Programming*, Brameier, Markus, Banzhaf, Wolfgang 2006, Approx. 320 p., Hardcover. [WWW]
- *Evolutionary Algorithms for Solving Multi-Objective Problems*, Coello Coello, Carlos A., Van Veldhuizen, David A., Lamont, Gary B. 2nd ed., 2006, Approx. 700 p. 50 illus., Hardcover [WWW]
- *Evolvable Hardware*, Higuchi, Tetsuya; Liu, Yong; Yao, Xin (Eds.) 2006, VIII, 224 p., 30 illus., Hardcover. [WWW]
- *Adaptive Learning of Polynomial Networks: Genetic Programming, Backpropagation and Bayesian Methods*, Nikolaev, Nikolay, Iba, Hitoshi 2006, XIV, 316 p. 62 illus., Hardcover [WWW]

  Genetic Programming Theory and Practice III Series: Genetic Programming , Vol. 9 Volume package: Genetic Programming Theory and Practice Yu, Tina; Riolo, Rick; Worzel, Bill (Eds.) 2006, XVI, 310 p., Hardcover [WWW]

Series on Studies in Computational Intelligence

Series Editor: Kacprzyk, Janusz
Series Homepage: [WWW]

- *Genetic Systems Programming: Theory and Experiences*, Nadia Nedjah, Luiza deMacedo Mourelle, Ajith Abraham (Eds.) Series: Studies in Computational Intelligence, Vol. 13 2006, XXII, 233 p., 114 illus., Hardcover. [WWW]
- *Advances in Evolutionary Algorithms: Theory, Design and Practice*, Ahn, Chang Wook Series: Studies in Computational Intelligence , Vol. 18 2006, XV, 171 p., 56 illus., Hardcover. [WWW]
- *Gene Expression Programming Mathematical Modeling by an Artificial Intelligence*, Ferreira, Cândida Series: Studies in Computational Intelligence, Vol. 21 2nd edition, 2006, XX, 478 p., 159 illus., Hardcover. [WWW]
- *Parallel Evolutionary Computations*, Nedjah, N.; Alba, E.; Macedo Mourelle, L. de (Eds.) Series: Studies in Computational Intelligence , Vol. 22 2006, XXIII, 201 p., 98 illus., Hardcover. [WWW]
- *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Pelikan, Martin; Sastry, Kumara; Cantú, Erick (Eds.) Series: Studies in Computational Intelligence, Vol. 33 2006, Approx. 285 p., Hardcover. [WWW]
- *Swarm Intelligent Systems* Nedjah, Nadia; Mourelle, Luiza de Macedo (Eds.) Series: Studies in Computational Intelligence , Vol. 26 2006, XX, 186 p., 65 illus., Hardcover. [WWW]

MIT Press

- *Metacreation: Art and Artificial Life* Mitchell Whitelaw, April 2006 ISBN 0-262-73176-2 6 x 9, 296 pp., 34 illus. [WWW]
- *Artificial Life X* Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems Luis Mateus Rocha, Larry S. Yaeger, Mark A. Bedau , Dario Floreano, Robert L. Goldstone and Alessandro Vespignani (Eds.) August 2006 [WWW]

Series on Studies in Fuzziness and Soft Computing

Series Editor: Kacprzyk, Janusz
Series Homepage: [WWW]

- *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, Endika Bengotxea (Eds.) Series: Studies in Fuzziness and Soft Computing, Volume 192 / 2006. [WWW]

- *Engineering Hybrid Soft Computing Systems*, Abraham, Ajith, Dote, Yasuhiko, Series: Studies in Fuzziness and Soft Computing, 2006, Approx. 500 p., Hardcover. [WWW]

Lecture Notes in Computer Science

- *Theory and Applications of Models of Computation*. Third International Conference, TAMC 2006, Beijing, China, May 15-20, 2006, Vol. 3959. Cai, Jin-Yi; Cooper, S. Barry; Li, Angsheng (Eds.) 2006, XV, 794 p., Softcover [WWW]
- *Engineering Self-Organising Systems* Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 25, 2005, 3910. Brueckner, Sven A.; Di Marzo Serugendo, Giovanni; Hales, David (Eds.) 2006, XII, 245 p., Softcover. [WWW]
- *DNA Computing* 11th International Workshop on DNA Computing, DNA11, London, ON, Canada, June 6-9, 2005, Vol. 3892. Carbone, Alessandra; Pierce, Niles A. (Eds.) 2006, XI, 440 p., Softcover. [WWW]
- *Artificial Evolution* 7th International Conference, Evolution Artificielle, EA 2005, Vol. 3871 Talbi, E.-g.; Liardet, P.; Collet, P.; Lutton, E.; Schoenauer, M. (Eds.) 2006, XI, 310 p., Softcover. [WWW]
- *Biologically Inspired Approaches to Advanced Information Technology* Second International Workshop, BioADIT 2006, Osaka, Japan 26-27, 2006, Vol. 3853. Ijspeert, Auke Jan; Masuzawa, Toshimitsu; Kusumoto, Shinji (Eds.) 2006, XIV, 388 p., Softcover [WWW]
- *Membrane Computing* 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Vol. 3850. Freund, R.; Paun, G.; Rozenberg, G.; Salomaa, A. (Eds.) 2006, IX, 371 p., Softcover [WWW]

Springer

- *Handbook of Nature-Inspired and Innovative Computing Integrating Classical Models with Emerging Technologies* Zomaya, Albert Y. (Ed.) 2006, XVI, 736 p. 211 illus., Hardcover [WWW]

  Adaptive Business Intelligence Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C. 2006, X, 220 p., Hardcover. Available: August 2006. [WWW]

2007 EVO*

www.evostar.org

EUROGP
EVOCOP
EVOBIO
EVOWORKSHOPS

11-13 APRIL 2007
valencia::spain

MORPHOGENESIS SERIES #12
(EVOLVED PLANT FORM)> JON McCORMACK 2006

Calls and Calendar

August

ACM KDD-2006

August 20 - 23, 2006 Philadelphia, USA

http://www.acm.org/sigs/sigkdd/kdd2006/

The 12th ACM SIGKDD conference will provide a forum for researchers from academia, industry, and government, developers, practitioners, and the data mining user community to share their research and experience. The SIGKDD conference will feature keynote presentations, oral paper presentations, poster presentations, workshops, tutorials, and panels, as well as the KDD Cup competition.

ECAI Workshop on Evolutionary Computation

Riva del Garda, Italy, 28 August 2006

http://www.ce.unipr.it/ec2ai2006

The workshop will comprise tutorials and technical presentations, in order to address the participation of as wide an audience as possible, from researchers and students who are already working in the field of Evolutionary Computation and Artificial Intelligence, to representatives of industry and everybody who is interested in approaching evolutionary computation from the point of view of both basic research and applications.

September

PPSN - Parallel Problem Solving from Nature IX

Reykjavik, Iceland, September 9 - 13, 2006

http://ppsn2006.raunvis.hi.is/

The conference emphasises original theories and novel applications of natural computing. World-leading researchers in the field of natural computing will present keynote talks and tutorials at the conference. The conference proceedings will be published by Springer in its LNCS series. A number of specialist workshops will be run just before the main conference.

Workshop on Bio-Inspired Computing in Computational Biology

Reykjavik, Iceland, September 9, 2006

Homepage: WWW

Deadline July 31, 2006

Organized by: Systems Optimization Group, ETH Zurich, Switzerland

Computational biology aims at understanding biological phenomena by means of mathematical models and computational methods, e.g., optimization techniques. In the analysis of biological data sets, a variety of complex optimization problems need to be addressed. Bio-inspired computing offers high flexibility in this respect as it can deal with complex problem formulations and does not require a specific algorithm to be developed for each variation of a problem formulation. The increasing interest of the biology community and the increase of available high-throughput data creates many opportunities for innovative contributions in this highly interdisciplinary field.

We invite submissions of original, possibly preliminary work as extended abstracts of two pages together with a tentative title of the talk, author name(s), affiliation(s), and email addresses. The extended abstracts should be submitted in PDF format via email to Simon Barkow: barkow@tik.ee.ethz.ch. Accepted contributions will be included in the workshop proceedings, which will be distributed among the workshop participants, and will be presented by one of corresponding authors at the workshop.

From Animals to Animats 9

25-29 September 2006, CNR, Roma, Italy

http://sab06.org

The objective of this interdisciplinary conference is to bring together researchers in computer science, artificial intelligence, alife, control, robotics, neurosciences, ethology, evolutionary biology, and related fields so as to further our understanding of the behaviors and underlying mechanisms that allow natural and artificial animals to adapt and survive in uncertain environments.

ABiALS-2006

30 September 2006, CNR, Roma, Italy

http://www-illigal.ge.uiuc.edu/ABiALS/

The Anticipatory Behavior in Adaptive Learning Systems (ABiALS) workshops are designed to encourage interdisciplinary research on anticipatory behavior in animals, animats, and artificial intelligence systems. Submission that investigate anticipatory behavior mechanisms are encouraged.

October

SEAL'06

15-18 October 2006, Hefei, Anhui, China

http://nical.ustc.edu.cn/seal06/

Evolution and learning are two fundamental forms of adaptation. SEAL'06 aims at exploring these two forms of adaptation and their roles and interactions in adaptive systems. Cross-fertilisation between evolutionary learning and other machine learning approaches will be strongly encouraged by the conference. The other major theme of the conference is optimisation by evolutionary approaches or hybrid evolutionary approaches.

December

ICARA 2006: Special Session on Artificial Life and Living Robots

December 12-14, 2006, Palmerston North, New Zealand

http://icara.massey.ac.nz/

Deadline July 15, 2006

Organised by Peter J. Bentley, University College, London

The study of living systems has never been more relevant to robotics. Artificial life now informs us of how organisms can reproduce, build and repair themselves. Bio-inspired algorithms give our technology the ability to adapt, evolve and learn. Bio-mimetics teaches us how novel materials may be exploited to achieve new capabilities. This special session focuses on such exciting new areas, and aims to produce an in-depth discussion about how biology should be used to improve our technology. Papers should be submitted to Peter Bentley p.bentley@cs.ucl.ac.uk following the ICARA conference guidelines.

BIONETICS 2006

December 11-13, 2006, Madonna di Campiglio, Italy.

http://www.bionetics.org/

Deadline July 31, 2006

The BIONETICS conference aims at bringing together researchers and scientists from several disciplines in computer science and engineering where bio-inspired methods are investigated. We are soliciting high-quality original papers in the following five areas including but not limited to the following topics: a) Bio-inspired mathematical models, methods and tools; b) Bio-inspired software; c) Bio-inspired security mechanisms; d) Bio-inspired networks and communication systems; e) Bio-inspired and bio-based nano-scale communication and information systems. Submission instructions at [WWW].

January 2007

Foundations of Genetic Algorithms

7-11 January 2007, Mexico City, Mexico

Deadline September 20, 2006

Requests for attendance due September 20, 2006

We invite submissions of extended abstracts for the ninth biennial workshop on the Foundations of Genetic Algorithms. The workshop covers the theoretical foundations of all forms of evolutionary computation. FOGA will be held 7-11 January, 2007 in Mexico City. Attendance at the workshop will be limited; the goal is to create a small interdisciplinary forum with close interaction among participants from different fields - evolutionary computation, population genetics, animal behaviour, physics and biochemistry, among others. Individuals submitting papers will be given priority for attendance, and some slots will be reserved for students. Anyone wishing to attend must indicate this by either submitting a paper or requesting attendance in advance (see deadline).

Extended abstracts must be received by 20th September, 2006. Submissions should address theoretical issues in evolutionary computation. Papers that consider foundational issues and/or are of a multidisciplinary nature are especially encouraged. This does not preclude the acceptance of papers that use an experimental approach, but such work should be directed towards validation of suitable hypotheses concerning foundational matters.

Extended abstracts should be between 10-12 pages (single column). To submit an extended abstract, please email a compressed postscript or a pdf file to stephens@nucleares.unam.mx and mtoussai@inf.ed.ac.uk no later than 20th September 2006. In their submission message authors should provide the title of the paper, and the name, address and affiliation of the author(s). Authors should submit papers in single column format with standard spacing and margins, and 11pt or 12pt font for the main text. Authors using LaTeX should either use the standard article style file or the FOGA style file which can be found at the conference web-site.

April 2007

EuroGP 2007
Tenth European conference on Genetic Programming

April 11-13, 2007, Valencia, Spain.

Homepage: WWW

Deadline November 1, 2006

EuroGP is the premier conference in Europe devoted entirely to genetic programming. We invite submissions on all aspects of evolutionary generation of computer programs featuring new original research. The standard for submissions is high. Reviewing is double-blind. The conference will feature a mixture of oral presentations and poster sessions. Accepted papers will be published as papers in a volume of the Springer Lecture Notes in Computer Science.

EuroGP2007 will be held in Valencia, Spain, in conjunction with EvoBIO (5th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics), EvoCOP2007 (7th European Conference on Evolutionary Computation in Combinatorial Optimization) and EvoWorkshops. High quality papers are sought on topics strongly related to the evolution of computer programs, ranging from theoretical work to innovative applications.

EvoCOP 2007 - Seventh European Conference on Evolutionary Computation in Combinatorial Optimisation

April 11-13, 2007, Valencia, Spain.

Homepage: WWW

Deadline November 1, 2006

The EvoCOP series, started in 2001 and held annually since then, was the first event specifically dedicated to the application of evolutionary computation and related methods to combinatorial optimization problems. Following the general trend of hybrid metaheuristics and diminishing boundaries between the different classes of metaheuristics, EvoCOP has broadened its scope and now explicitly invites submissions on any kind of metaheuristic for combinatorial optimization. Each accepted paper will be presented orally at the conference and printed in the proceedings published by Springer in the LNCS series (see LNCS volumes 2037, 2279, 2611, 3004, 3448, and 3906 for the previous proceedings).

The conference will be held in conjunction with the 10th European Conference on Genetic Programming (EuroGP 2007), the Fifth European Conference on Evolutionary Computation (EvoBIO 2007) and EvoWorkshops 2007, a collection of application-oriented workshops in the field of evolutionary computation.

EvoBIO 2007 - Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics

April 11-13, 2007, Valencia, Spain.

Homepage: WWW

Deadline November 1, 2006

EvoBIO covers research in all aspects of Evolutionary Computation, Machine Learning and Data Mining in bioinformatics.

The goal of the conference is to not only present recent research results and to identify and explore directions of future research, but also stimulatesynergy and cross fertilization among Evolutionary Computation, Machine Learning and Data Mining for Bioinformatics.

The emphasis is on novel advanced techniques addressing important problems in molecular biology, proteomics, genomics and genetics, that have been implemented and tested in simulations and on real-life datasets, in particular microarray analysis, phylogeny, biomarker discovery, proteomics, high-throughput biotechnology, sequence analysis and alignment, ecological modelling, cell simulation and modelling, protein interaction. The conference will be held in conjunction with EuroGP2007, EvoCOP2007, and EvoWorkshops.

EvoWorkshops 2007

April 11-13, 2007, Valencia, Spain.

Deadline November 1, 2006

EvoWorkshops 2007 is a joint event of eight different workshops on Applications of Evolutionary Computation. Since 1998, EvoWorkshops has represented a unique opportunity for a broad and continuously increasing number of researchers to meet and present their advances in various application areas of evolutionary computation techniques. As a result, over the last seven years, EvoWorkshops has become one of the major events focusing solely on applicational aspects of EC, constituting an important link between EC research and the application of EC in a a wide range of domains. The standard of submissions is high, and the reviewing process is double-blind. Accepted papers are published in a volume of Springer Lecture Notes in Computer Science.

EvoWorkshops 2007 will be held in conjunction with the Tenth European Conference on Genetic Programming, the Seventh European Conference on Evolutionary Computation in Combinatorial Optimisation, and the Fifth European Conference on Evolutionary Computation on Evolutionary Bioinformatics. Next year's EvoWorkshops will comprise of the following individual workshops:

- EvoCOMNET: Communications, networks, and connected systems
- EvoIASP: Image analysis and signal processing
- EvoHOT: Evolutionary algorithms for hardware optimization techniques
- EvoInteraction: Interactive evolution and humanized computational intelligence
- EvoMUSART: Evolutionary music, art, and creative systems
- EvoPhD: Graduate student workshop on evolutionary computation
- EvoSTOC: Stochastic and dynamic environments
- EvoTransLog: Transportation and logistics

IEEE Symposium Series on Computational Intelligence and Scheduling

April 1-5, 2007, Hilton Hawaii Village Resort, Honolulu, HI, USA

Deadline October 31 2006

http://www.cs.nott.ac.uk/~rxq/cis/CIS2007.htm

CISched 2007 aims to bring together leading researchers and practitioners in computational intelligence and scheduling. Scheduling problems are often not amenable to be being tackled by exact approaches due to the huge search spaces that have to be explored. Therefore we often resort to techniques which fall under the term of Computational Intelligence. These can include Evolutionary Computation, Neural Networks, Fuzzy Logic etc. This symposium aims to explore recent advances in this area.

September 2007

IEEE Congress on Evolutionary Computation

September 25-28, 2007, Singapore.

Deadline March 15, 2007

www.cec2007.org

CEC 2007 will feature a world-class conference that aims to bring together researchers and practitioners in the field of evolutionary computation and computational intelligence from all around the globe. Technical exchanges within the research community will encompass keynote speeches, special sessions, tutorial workshops, panel discussions as well as poster presentations. On top of this, participants will be treated to a series of social functions, receptions and networking sessions, which will serve as a vital channel to establish new connections and foster everlasting friendship among fellow counterparts.

About the Newsletter

SIGEVOlution is the newsletter of SIGEVO, the ACM Special Interest Group on Genetic and Evolutionary Computation.

To join SIGEVO, please follow this link [WWW]

Contributing to SIGEVOlution

We solicite contributions in the following categories:

Art: Are you working with Evolutionary Art? We are always looking for nice evolutionary art for the cover page of the newsletter.

Short surveys and position papers: We invite short surveys and position papers in EC and EC related areas. We are also interested in applications of EC technologies that have solved interesting and important problems.

Software: Are you are a developer of an EC software and you wish to tell us about it? Then, send us a short summary or a short tutorial of your software.

Lost Gems: Did you read an interesting EC paper that, in your opinion, did not receive enough attention or should be rediscovered? Then send us a page about it.

Dissertations: We invite short summaries, around a page, of theses in EC-related areas that have been recently discussed and are available online.

Meetings Reports: Did you participate to an interesting EC-related event? Would you be willing to tell us about it? Then, send us a short summary, around half a page, about the event.

Forthcoming Events: If you have an EC event you wish to announce, this is the place.

News and Announcements: Is there anything you wish to announce? This is the place.

Letters: If you want to ask or to say something to SIGEVO members, please write us a letter!

Suggestions: If you have a suggestion about how to improve the newsletter, please send us an email.

Contributions will be reviewed by members of the newsletter board.

We accept contributions in $\LaTeX$, MS Word, and plain text.

Enquiries about submissions and contributions can be emailed to editor@sigevolution.org.

All the issues of SIGEVOlution are also available online at www.sigevolution.org.

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in the Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance be the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.