SIGEVOlution

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

Spring 2007
Volume 2 Issue 1

in this issue

EC Practitioners
Gregory S. Hornby & Tina Yu

Interactive Music Composition
Ying-ping Chen

The Columns
GECCO-2007 Highlights
announcements
dissertation corner
forthcoming papers
calls & calendar

Editorial

Happy Birthday SIGEVOlution! One year ago the first issue of the first volume was distributed to the ACM SIGEVO mailing list. The newsletter has its own website since June 2006 and, since then, Google Analytics has been used to keep track of several statistics. So far, the website had 1655 unique visitors (30% of them returned more than once) and around 3200 downloads. 52% of the visitors reached the newsletter directly, through www.sigevolution.org, 41% came from a referring site, while 7% came from search engines. Among the referring web sites www.sigevo.org is responsible for 20% of the visits while www.kdnuggets.com, the major data mining web site, contributes to 4% of the visits. The newsletter is also available on www.slideshare.net where I uploaded the first four issues (spring 2006, summer 2006, autumn 2006, and winter 2006). The plan for the forthcoming year is to host the online versions of the published articles on the SIGEVOlution web site so that they can be indexed by search engines and comfortably browsed online.

In this issue, we continue our walk through the EC community with Gregory Hornby and Tina Yu who, between March 2005 and February 2006, conducted a survey of EC practitioners working in both academia and industry. Then, Ying-ping Chen shows us a framework for interactive music composition with EC.

The cover photo is by Sarah McGee: I found it while browsing www.flickr.com, she has many other beautiful photos at www.flickr.com/photos/smcgee/.

This issue was brought to you with the help of many people who helped me during the whole process: the authors, Gregory S. Hornby, Tina Yu, Ying-ping Chen, and Moshe Looks; the board members, Dave Davis and Martin Pelikan; and also Pat Cattolico, Martin V. Butz, Kumara Sastry, Xavier Llorà, Marc Schoenauer, and Erik Goodman.

Pier Luca
May 31st, 2007

SIGEVOlution
Spring 2007, Volume 2, Issue 1

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

Contributors to this Issue

Gregory S. Hornby
Tina Yu
Ying-ping Chen

Contents

EC Practitioners **2**
Gregory S. Hornby & Tina Yu
Interactive Music Composition **9**
Ying-ping
GECCO-2007 Highlights
Best Papers Nominees **17**
The Competitions **21**
The Workshops **24**
The Tutorials **26**
Announcements **28**
Dissertation Corner **31**
Forthcoming Papers **32**
Calls and Calendar **33**
About the Newsletter **39**

ISSN: 1931-8499

EC Practitioners: Results of the First Survey

Gregory S. Hornby, University of California Santa Cruz, Moffett Field, CA, hornby@email.arc.nasa.gov
Tina Yu, Computer Science Department, Memorial University of Newfoundland, NL, Canada, tinayu@cs.mun.ca

The field of Evolutionary Computation (EC) has been around for several decades [2, 3], and in recent years there has been an explosion not only in the different types of biologically inspired algorithms, but also in the number of practitioners in the field. A critical part of this growth and development of the EC field has been the technology transfer of EC from academia to industry and the successful application of EC techniques to real-world problems. To assist in the continued technology transfer of EC techniques from academia to industry we conducted a survey of EC practitioners working in both academia and industry and in this article we summarize some of our findings.

The survey was conducted between March 1, 2005 and February 28, 2006 by posting 14 survey questions on the SIGEVO website. In particular, the survey asked about participants' background information, job information and obstacles encountered while applying EC techniques to industry jobs. Some of the main findings from our results are that: there has been an exponential growth in both EC graduates and practitioners; the main source for finding a job has been networking; while most respondents to our survey are in Europe, the most growth of EC in industry has been in North America; the main application areas of EC techniques are multi-objective optimization, classification, data mining and numerical optimization; and the biggest obstacle for the acceptance of EC techniques in industry is that it is poorly understood.

The survey we ran had three parts. First, it asked several questions about the participants' background. The second part of the survey had questions on their job information, and the third part was only for non-academic jobs and asked about EC acceptance and applications at that organization.

Methodology

The respondents to this survey were not randomly selected but were recruited through a variant of the snowball sampling strategy [4]. The recruiting methods include posting the survey announcement to various EC mailing lists (such as EC-Digest and genetic-programming), e-mailing the announcement to attendants of major EC conferences (such as GECCO-05, GPTP-05, EH-05) and advertising the survey at these conferences. Snowball sampling relies on referrals from the initial subjects to generate other subjects. Although snowball sampling may introduce bias into the study, it can be effective in reaching groups having common characteristics [1]. In our case, many EC practitioners are likely to subscribe to EC-related mailing lists and attend EC-related conferences, hence they can be reached by our recruiting approach. However, snowball sampling does not qualify as a random process. Consequently, the results from this survey cannot be generalized to the entire EC practitioner population, regardless of the number of responses received. Nevertheless, these results are still useful for gaining a preliminary picture of EC-practitioners in the world.

Over the one year time period in which the survey was taken, 324 responses were received, of which 305 had some EC relation, either through graduating with a degree specialized in EC or by using EC in one of their jobs. For the results of this survey, only the 305 responses which had an EC connection were used.

Background Information

The first part of the survey asked participants to provide basic background information, such as gender, and to answer some questions about the most recent degree that they had received. We found that 71.1% of participants have a Ph.D. and the gender split is 87.5% male and 12.5% female. Looking into the geographic regions from which participants graduated, we found that most participants graduated from Europe (46.2%), which is followed by North America (35.7%), Asia (12.5%), Oceania (2.6%), South America (2.0%), and Africa (0.7%). When we looked for yearly trends in these percentages based on graduation date, we found that they have remained fairly constant throughout the years.

One change that has occurred over the years is in the amount and specialization of graduates. The graduation rate has an exponential growth, starting with only a couple of people graduating a year from the 1960's up until the end of the 1980's, at which point the numbers increase dramatically and reach a peak of 36 graduates in 2004. For the first few decades, none of those who graduated in this time period had a degree specialized in EC. The first EC grad does not show up until 1991, and then starting in 1996 the majority of graduates have an EC-specialized degree. This suggests that EC emerged as a field of its own sometime in the mid-1990s.

Job Sources

After graduation, the next step is finding a job. By far the most common source of a job was networking, through which 35.1% of participants found a job. This was followed by *other* (25.7%), *supervisor* (17.2%), *postings at university department* (13.8%), *web* (4.7%), *campus career services center* (2.0%) and *mailing list* (0.7%). Looking for differences between those who took a job in academia versus those who took a non-academic job we found that *networking* was used more for finding a non-academic job (43.0%) than it was for finding one in academia (31.5%). In contrast, the reverse was true for *postings at the university department*: it was used by 16.5% of those who took an academic position but by only 8.0% of those who took a job in industry. Of those who selected *other*, 19 found their position through a listing in a journal or society magazine (such as the Communications of the ACM and IEEE), 13 found their job through an advertisement in the newspaper, 11 founded their own company, and 6 applied and received a research grant.

Further examination of the correlation between the job areas and the job-hunting methods found only a couple of patterns. One is that *postings at the university department* helped in finding jobs in Energy, Robotics and Government laboratories, but was of little use for the other job areas. Similarly, the *campus career center* had some success only in finding jobs in Government laboratories and Other. When job-finding methods are analyzed with the job regions, it shows some additional regional trends. *Networking* was used to find over half of the jobs in North America (as well as in Africa, Oceania and South America), but for less than a third of jobs in Europe, and for only 15% of jobs in Asia. In Europe, *supervisors* helped to find roughly a quarter of all jobs, and they were also helpful in Asia but were not very useful for finding jobs in North America. The *campus career center* was used by a small percentage of the respondents in Asia and North America, but was not used in any other geographic region.

Job Regions

Looking into the distribution of jobs by geographic region, we found that most EC jobs have been in Europe (45%), followed by North America (37%), Asia (10%), Oceania (3%), South America (2%) and then Africa (2%). This geographic distribution of jobs closely matches the geographic distribution of graduates and suggests a strong correlation between where a graduate studied and where s/he worked. Also, the ratio of positions between the different geographic regions has been fairly constant over the years.

When the job positions are grouped by geological regions, analyzing the responses over the years reveals that the ratio between positions in industry and in academia has been fairly constant in recent years both in Europe (1:3) and in North America (2:3). In contrast, Asia has experienced a shift in its ratio from being predominantly in industry (100% non-academic in 1981) to being predominantly in academia (more than 75% academic in 2005). For the other geographic regions, the numbers of respondents was too small to give a meaningful interpretation.

Examining the movement of EC graduates for work reveals some interesting trends. First, none of the respondents who graduated with a degree specialized in EC from Africa or South America have left their regions for a job and only 12% of people who graduated in Europe or North America ever move to a different region for work. In contrast, 44% of EC graduates in Asia and 40% of EC graduates in Oceania move at some point

after graduation. Second, the direction of movement in Asia, Europe and North America is toward the West. Of those graduates who moved to a different geographic region for a job we found that: 62% of those graduating in Asia moved to Europe at some point, but only 25% ever moved to North America; 70% of those graduating in Europe moved to North America but only 20% moved to Asia; and 67% of those graduating in North America moved to Asia but only 17% moved to Europe for a job. Thirdly, for those people who moved from another region to North America, half moved for jobs in academia and half for jobs in industry, but for those participants who moved to a region other than North America, in all cases they went for academic positions.

EC Positions, Problem Types and Application Areas

Once in a job, we are interested in what kind of position in their organization the respondent had, as well as whether or not EC was used and how it was applied. For determining trends by year each job entry, with its start and end years, was converted into yearly *positions*. That is, a job from 1997 to 2001 was separated into five positions, one each in 1997, 1998, 1999, 2000 and 2001. To limit participants such that they had at most one position in each year, jobs with overlapping years were modified so that the second job started in the year after the first job ended. For example, if a participant had a job from 1995 to 1998 followed by one from 1998 to 2001, the start year for the second job was changed from 1998 to 1999. Using this method, the 424 jobs that used EC techniques were mapped to 2955 EC-related positions.

From our responses we found that there has been an exponential growth in positions in the field, starting with a single EC position in 1965 to just under 300 EC positions in 2005. Breaking this down into academic and non-academic positions, there has been a fairly steady proportion of just under two-thirds of the positions in a given year being academic and just over a third being non-academic. Figure 1a is a breakdown of the type of position held for those not working in academia. This figure shows that most industrial EC positions are in research, with a significant number in technical/software development and consultancy.

Looking into the types of problems that respondents worked on we found the following: 40.3% do Multi-objective optimization (MOO), 38.4% do Numerical optimization, 38.0% do Classification, 37.7% do Other, 31.6% do Data mining, 21.2% do Open-ended design, 21.2% do Scheduling,

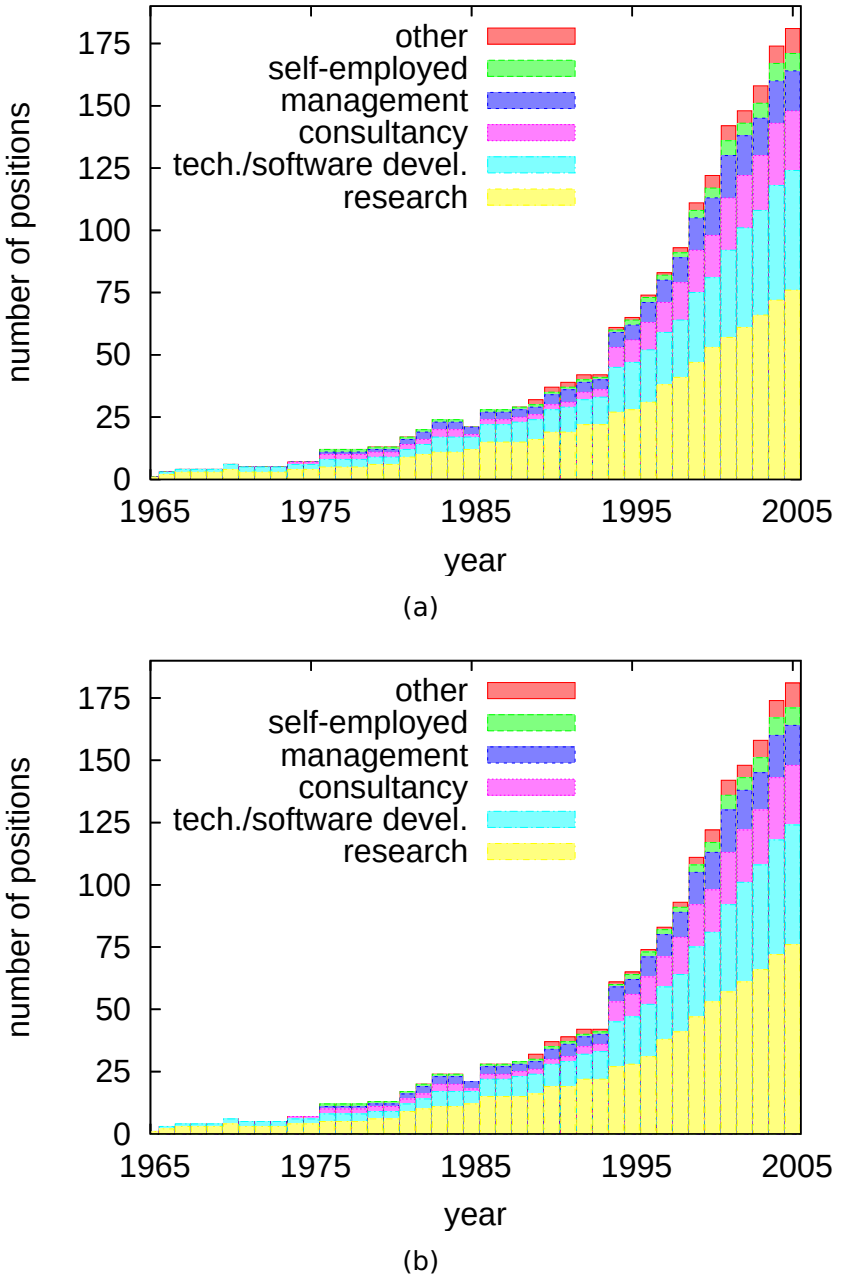

Figure 1: The figure contains boxplots of: (a) the type of industry position held; and, (b) the application area to which EC is applied.

Area	Percentage (%) working in this area	
	Academia	Industry
MOO	38.8	45.9
Classification	38.0	46.6
Num. opt.	36.7	47.0
Other	39.4	34.4
Data mining	28.7	38.6
Scheduling	19.8	37.6
Open-ended design	24.7	24.2
Planning	13.5	21.3
Sat./TSP	10.1	15.6

Table 1: Percentage of respondents working in each problem area.

13.9% do Planning, and 10.1% do Satisfiability/TSP. These values do not add up to 100% because participants were able to make multiple selections for each job. For those responses that selected "Other", participants were able to enter a response in a text field. The most popular entries that were given are: optimization and design (24); modeling and simulation (17), EC theory (15); biology and bio-informatics (11); control (11); evolutionary robotics (6); artificial life (5) and neural networks (5). Many of these entries for "Other" fit under the given categories (e.g. 'optimization and design' fits under Optimization and/or Open-ended design) with some of the other entries being an application area and not a problem type.

Comparing the distribution of problem-types worked on by academics to that of non-academics we found a significant difference (Table 1). In general, the percentage of academic positions that are working in a particular problem area is lower than that for non-academic positions. This means that academics tend to focus on fewer problem areas than those outside of academia. Specifically, those participants employed in academic positions average working on 2.24 problem areas whereas those in non-academic positions average working on 2.74 problem areas. Normalizing for this difference, *Scheduling* stands out as the one problem area which is significantly under-investigated by academics as compared to non-academics.

Different from the kind of problem being worked on (numerical optimization, scheduling, . . .), is the industry to which this problem is being applied (automotive, insurance, . . .). Figure 1b contains a histogram of EC industrial application areas by year.[1] The industry with the largest selection rate is *Other*, which was selected in 37% of all jobs. The most common areas given by those who selected *Other* were: IT (13), consulting (12), biology/medicine related (e.g. Bioinformatics, biomedicine, pharmaceutical) (10), defense and military (7), and various types of engineering (civil, structural or manufacturing) (7). For non-academic jobs, the ways in which EC is reported to be most useful are: *design* (52.3%), *operations* (33.1%), *invention* (27.8%), *testing* (15.9%) and *other* (14.6%). Of the 31 responses for *other*, 10 were for optimization.

Next we looked into how application areas varied by industry to see which combinations stand out (Table 2). Some specific combinations that we found are that those working in the automotive and robotics industries are interested in multi-objective and numerical optimization problems, while people working in the energy and entertainment industries are interested in multi-objective and classification problems. Finally, those working in insurance, telecommunications and the financial industries are predominantly interested in classification and data mining.

EC Acceptance in Industry

Next we examined non-academic jobs to see what trends exist in the distribution and acceptance of EC in industry. Even though there is an exponential growth in the number of yearly EC positions, the ratios between the different levels of distribution and acceptance have remained fairly constant throughout the years. The acceptance rate has averaged: 41.3% well accepted; 19.8% accepted; 36.9% somewhat accepted; and 2.0% rejected. The distribution rate has averaged: 36.4% well distributed; 12.3% distributed; 25.3% somewhat distributed; and 26.0% isolated. That these ratios have remained fairly constant over the years does not mean that EC is not becoming more distributed and accepted in academic organizations – in fact, the growth in number of EC positions implies the opposite. What we cannot determine from our data is whether there is an increase in acceptance and distribution within an organization over time, and this is a question for a future survey.

[1] Since each job was allowed to enter multiple application areas the total number of selected application areas can be greater than the number of positions.

Job Area	Tot	Percentage working in this problem type.								
	#	Clsf	DM	MOO	NO	Design	Plan	S/TSP	Sched	Oth
Academic	27	30	33	48	56	19	7	11	15	30
Aerospace	20	50	45	60	55	30	15	20	40	25
Auto	10	40	60	80	80	20	40	20	50	50
Energy	20	70	50	65	50	15	45	10	50	20
Enter.	6	67	33	67	50	67	17	17	50	33
Financial	20	65	80	55	35	20	20	5	35	30
Gov. Lab	34	35	32	41	38	29	15	9	24	29
Insurance	5	100	80	60	20	0	20	0	20	0
Robotics	12	33	33	50	50	25	17	8	33	50
Semi-con	8	50	25	62	25	38	0	12	12	0
Tele-com	12	58	58	50	42	17	17	33	33	17
Other	56	48	38	54	36	20	21	9	32	41

Table 2: Percentages of job areas that involve work in different problem types.

We also analyzed EC acceptance by geographic region. The breakdown of acceptance in Asia, Europe and North America is as follows (well accepted, somewhat accepted, not well accepted, rejected): Asia (53%, 7%, 40%, 0%); Europe (41%, 35%, 20%, 4%); and North America (42%, 21%, 34%, 3%); We do not give a breakdown for the other geographic regions due to insufficient responses.

Important for increasing the acceptance and distribution of EC in industry is an understanding of the obstacles to its uptake. Based on our responses, we found the obstacles to be: *poorly understood* (39.7%), *too ad hoc* (22.5%), *few successful applications to convince management* (21.2%), *commercial tools were unavailable or ineffective* (20.5%), *Other* (18.5%), *no proof of convergence* (14.6%), and *too hard to apply* (13.9%). In some ways, it is encouraging that the main obstacle is that EC is *poorly understood* because as more universities teach EC techniques, these methods should grow in familiarity and thereby gain wider acceptance in industry. Similarly, with a growth in familiarity of EC, companies may be less inclined to find it "ad hoc". The third main obstacle is *the lack of successful applications*, is being addressed through Real-World Applications tracks at EC conferences and with the Human Competitive Competition held at GECCO since 2004. Finally, *lack of useful commercial tools* suggests a possible market niche for those wanting to achieve commercial success with creating EC software. Among the 27 responses for *Other*, the most common obstacles were: lack of experience/familiarity (9), and too slow or does not scale (4).

Comments for Future Surveys

Having conducted the first survey of practitioners of evolutionary computation we have some thoughts on changes that should be done for future surveys. First off, to better understand EC education in universities it would be useful to ask for each degree received what the number of courses taken was in which EC techniques were covered. This would be beneficial for finding out how widely EC techniques are being taught to non-EC specialists and also to find out if EC is being more widely included in course curricula. Similarly, it would be useful to query people as to how many EC-specialized conferences they have attended in a given year, or the average number of such conferences they attended a year over the course of each job.

Second, in our questions on asking how well accepted/distributed EC is at a particular company, rather than having categories such as "Well accepted" to "Rejected or poorly accepted" for possible answers it would be more useful to ask for a numerical rating from 1 to 5, or 1 to 10 asking for the degree of acceptance. In this case 1 would be "Rejected" and the highest value would be "Well accepted." Such a numerical system would allow for more fine-grained ranking of acceptance and would allow for numerical processing on how acceptance has changed. Also, it would be useful to ask for the level of EC acceptance at the start of a job and the level of EC acceptance and the end of the job (or its current level of acceptance for jobs in which the respondent is still currently employed at). This would allow for analyzing whether there has been an increase in acceptance of EC at individual companies over time. Another useful question to ask for the size of the company or organization where the respondent is working. It would be interesting to see if there are trends in the size of organization that uses EC, or in its growth in acceptance.

Finally, in addition to canonical evolutionary algorithms (such as genetic algorithms and evolutionary strategies) in recent years various other biologically-inspired computing algorithms such as ant colony optimization, artificial immune systems and particle swarm optimization have been developed. It would be useful to add a question asking respondents about which techniques they have used at each of their jobs so as to track their use and also learn what applications they are being used for.

Conclusion

Over the years, the use of EC techniques has grown from a few isolated practitioners into a genuine field with a large community. This first survey on EC practitioners has provided us with a preliminary picture of its development in the world. There has been an exponential growth in the number of EC practitioners and EC-specialized graduates, with the first graduates with EC-specialized degrees appearing in the mid 1990's. After graduation, most survey participants found their jobs through networking or from their supervisors. Encouragingly, along with the growth in EC positions has been a growth in acceptance of EC techniques in industry, with the main obstacle to industry acceptance being that the technique is not well understood. EC has been applied to a wide variety of application areas and different problem domains, among which the most common problem areas are multi-objective optimization, classification, and numerical optimization. Although there are still challenges to the continued transfer of Evolutionary Computation to industry, we hope that the results of this survey will help.

Acknowledgments

We would like to thank Gerardo Valencia for putting together the survey website, and also Erik Goodman, David Goldberg, John Koza and Una-May O'Reilly for their support in conducting this survey.

Bibliography

[1] R. Atkinson and J. Flint. Accessing hidden and hard to reach populations: Snowball research strategies. *Social Research Update*, 33, 2001.

[2] K. A. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

[3] D. B. Fogel, editor. *Evolutionary Computation: The Fossil Record*. IEEE Press, Piscataway, NJ, 1998.

[4] W. P. Vogt. *Dictionary of Statistics and Methodology: A Nontechnical Guide for the Social Science*. Sage, London, 1999.

About the authors

Greg Hornby was a visiting researcher at Sony's Digital Creatures Laboratory from 1998 to 1999, received his Ph.D. in Computer Science from Brandeis University in 2002, and is currently a Project Scientist with U.C. Santa Cruz at NASA Ames Research Center. His main area of research has been toward developing a science of generative representations and complexity measures for open-ended, computer-automated design. Among his accomplishments are the autonomous evolution of the dynamic gait for the consumer version of Sony's pet robot AIBO, and the evolution of the antenna for NASA's ST-5 Mission, which in 2006 became the first artificially-evolved hardware in space and the first evolved antenna to be used in an actual application. In addition, he developed the Age-Layered Population Structure (ALPS) Evolutionary Algorithm for reducing the problem of premature convergence.

Homepage: ti.arc.nasa.gov/people/hornby
Email: hornby@email.arc.nasa.gov

Tina Yu received her Ph.D. from University of London, U.K in 1999. Between 1999 and 2005, she was with Chevron Technology Company. In 2005 she joined the Memorial University of Newfoundland as an Associate Professor. Tina's recent research focuses on advancing the understanding and application of intelligent computing systems based on the modeling of biological processes, such as evolution, learning and cooperative behavior. This research incorporates theoretical analysis, statistical and empirical methods. The target results are new insights into system behavior, novel techniques to extend the current state-of-the-art and innovative real-world applications. As part of this focus she has co-edited 3 books in applying EC to real-world problems.

Homepage: www.cs.mun.ca/ tinayu
Email: tinayu@cs.mun.ca

Interactive Music Composition with the CFE Framework

Ying-ping Chen, ypchen@cs.nctu.edu.tw
Natural Computing Laboratory (NCLab), Department of Computer Science, National Chiao Tung University, Taiwan

This article presents an interactive music composition system which utilizes the black-box optimization model of evolutionary computation. The core CFE framework—Composition, Feedback, and Evolution—is presented and described. The music composition system produces short, manageable pieces of music by interacting with users. The essential features of the system include the capability of creating customized pieces of music based on the user preference and the facilities specifically designed for generating a large amount of music. Finally, several pieces of music composed by the described system are demonstrated as showcases. This work shows that it is feasible and promising for computers to automatically compose customized or personalized music.

1 Introduction

Music plays an important role in our daily life. It makes us sad, happy, and excited. One may wish to listen to "pleasant music", but the definition of pleasant music is quite different for different people. Hence, composing music that is loved by everyone is an extremely difficult task, if not impossible. Furthermore, nowadays we are surrounded by lots of electronic devices capable of playing music or generating sound, such as alarm clocks and cellular phones. These devices oftentimes can be customized to play the user-specified music. For example, we can observe that many people try to use different, distinguishable ringtones for their cellular phones. The purpose for us to do so is not merely to distinguish phone calls but also to establish self-identities by using the music or sound that can define us. As a result, customization for pleasant music is desirable for our modern life.

In addition to music customization, for some applications, a large amount of music pieces may be needed, such as the scene music of games and the background music of web pages. It would be fantastic if ordinary people can easily create music or sound on their own. Although there are lots of computer software which can help people to compose music, such a task of composition is still very hard for unskilled or untrained people. In order to resolve this situation, we are trying to make computers able to automatically create music for us instead of merely letting us put notes into tracks. For this purpose, we develop a system which creates pieces of music by interacting with users. The generated music can be used on cellular phones, alarm clocks, or other devices of which the music can be set or loaded by the user.

In particular, we design an interactive music composition system based on the techniques borrowed from two fields. One is evolutionary computation [1, 2, 3, 4, 5]. Based on the concepts and models of evolutionary computation, we build the kernel optimization mechanism which can interact with the user and consider the scores given by the user as the objective values. The other is computer music. More specifically, we adopt the MIDI format, which is used in the system as the output format. If we create music in the MIDI format, we can guarantee that the created music can be played on computers, cellular phones, or other customizable devices.

The article is organized as follows. Section 2 briefly reviews the current state of creating music in the field of evolutionary computation. Section 3 describes the CFE framework, and section 4 presents the auxiliary functionalities for enhancing the system. The showcases are demonstrated in section 5, and the URLs for accessing these showcases are offered. Finally, section 6 concludes this article.

2 State of the Art

There have been several attempts to compose music with the techniques of evolutionary computation. In this section, we briefly review these proposed frameworks and broadly classify them by analyzing the three facets: the initialization, the grading method, and the goal to achieve.

Initialization. We can classify the frameworks based on the methods used to initialize the population in the evolutionary environment. There are several kinds of initialization procedures proposed in the literature:

- *Random initialization* [6] provides a relatively bad quality for the initial population but is limited by fewer restrictions than other methods are.
- *Complex function initialization* [6, 7] initializes the population through certain pre-designed rules and only produces individuals which satisfy the specified restrictions.
- *Song initialization* [8] creates the population by analyzing one or more available songs and by decomposing these songs into individuals.

Grading method. One of the essential components to create music is the way to judge or grade the music generated by the computer program. In the literature, we can find the following methods for grading music pieces:

- *Real audience.* One way is to judge the music with the real audience through either real-time judging [4, 9] or non-real-time judging [5]. Because lots of runs may needed in the evolutionary process, such a grading method may easily tire the audience.
- *Neural network.* Neural network modules can be trained to evaluate the generated music [5]. However, it takes tremendous time to train the neural network, and the judgment quality offered by a trained neural network is also hard to determine.
- *Artificial fitness functions.* Some frameworks utilize specific fitness functions [6, 7] to automatically grade the generated music. In these approaches, constructing an appropriate fitness function is hard and critical.

Goal to achieve. According to the different goals of the music creation frameworks proposed in the literature, we can have the following categories:

- *Theme of music:* To evolve the theme of music, a sequence of notes [4, 8] or a sequence of functions, such as $\sin(\cdot)$ and $\cos(\cdot)$ [7], is adopted as the genotype of the music.
- *Tempo of music:* To evolve the tempo of music, a sequence of tempo numbers [9] is adopted as the genotype of the music.

According to the three aforementioned facets, the differences of the present work from those previously proposed frameworks include: (1) For the initialization mechanism, the CFE framework initializes its population with a procedure in between random initialization and complex function initialization. The CFE framework randomly generates pieces of music as individuals probably with limited help of music theory, as described in section 4. (2) For the grading method, because the objective of the CFE framework is to generate personalized music, the real audience composed of only one single person is asked to evaluate the created music instead of using a real audience of many people or other computational techniques. (3) For the design goal, the proposed framework aims at creating short pieces of music instead of creating complete songs, which are usually the goal of previous studies.

3 The CFE Framework

In this section, we describe the CFE framework in detail to demonstrate that, for untrained people, creating personalized music by themselves is feasible and practical.

3.1 Overview

The CFE framework consists of three parts: Composition, Feedback, and Evolution. The structure of the system implemented in the present work is shown in Figure 1. In this framework, we try to find the best way to compose music rather than the “best” melody. To be more accurate, the

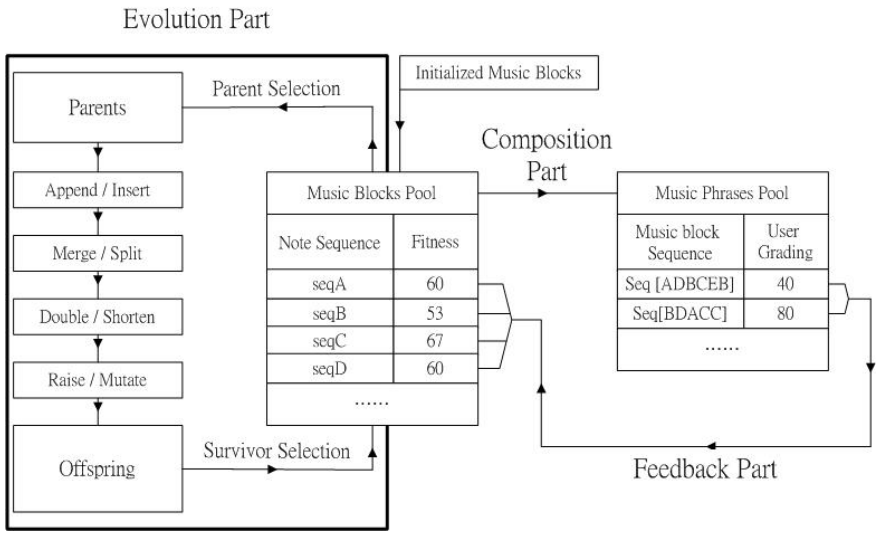

Figure 1: The structure of the CFE framework.

individuals in the evolutionary environment are no longer complete songs but some musical elements or guidelines. The Composition part uses these musical elements and guidelines to construct new melodies. The composed melodies then wait for the user's grading. After the system receives the information, the Feedback part distributes these feedbacks among the musical elements and guidelines for evaluating how fit these building blocks are. For discovering better methods, the methodology of evolutionary computation is adopted such that new elements are created into the population.

The three parts can be used separately. Therefore, once the user is satisfied with the composed music, no more work is necessary when he or she needs more pieces of music because Composition can be conducted alone. Since Composition and Evolution are isolated, for making use of the domain knowledge, such as the constraints, indications, and implications in the music theory, it is easier to embed such knowledge into Composition than to interfere with the regular operations of the evolutionary algorithm.

3.2 Composition

In the present work, the type of music which we focus on is the theme music of short, specific lengths, say, 8 or 16 measures. These music pieces are named *music phrases* in the framework. Inspired by some pop music that some subsequences appear in a song frequently and repeatedly, we take a layered approach to find out the potentially good sequences of notes. Our system deals with the short theme music by using two levels of hierarchy. The music phrase consists of short, variable-length sequences of notes, called *music blocks*. Composition picks the favored music blocks and fills in the incomplete music phrases until the specified length is reached.

3.3 Feedback

The design of the Feedback part provides the interface for users to give their responses to the system. We simply let users listen to the music phrase composed by Composition and let them grade it in the range from 0 to 100. It is not too complicated for users because the grading is episodic such that users do not have to listen to the music nervously for the need to make real-time responses like applauding. Once the grading is made, the score is distributed to all the music blocks contained in that phrase. Thus, the fitness value of a music block is determined by the average grade of all the music phrases in which the particular music block occurs. The key idea of this design is that good music blocks make good music.

3.4 Evolution

The Evolution part, seeking for the fittest music blocks, plays an essential role in the music composition system. We employ an evolutionary algorithm similar to a typical genetic algorithm, because music blocks can be easily and intuitively represented with a sequence of numbers.

The flow of the employed evolutionary algorithm works in the following way. Firstly, we initialize the population of which the individuals are music blocks containing only one single note with identical fitness values. Then, parent selection chooses one or two music blocks according to the fitness values. The common selection operators, such as tournament selection, can be used for this purpose. The selected parents will go through a set of operations, such as appending and inserting, to generate the offspring. Finally, survivor selection decides which music blocks stays in the

population and which blocks should be replaced by the newly generated music blocks. In our implementation, we use a fixed population size and remove the music blocks of the lowest fitness. In the following sections, we will introduce the operations designed for dealing with music blocks, including appending, inserting, merging, splitting, doubling, shortening, mutating, and raising.

3.4.1 Append and Insert

The Append operation concatenates two music blocks. The Insert operation, however, puts one music block into the other at a random position to search for better combinations of the two building blocks, as shown in Figure 2.

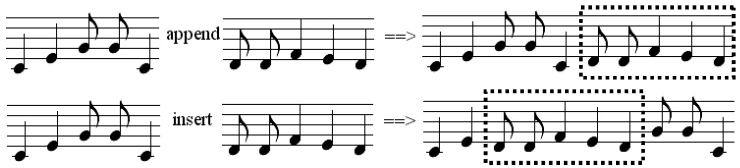

Figure 2: Operations: Append and Insert.

3.4.2 Merge and Split

The Merge operation chooses two adjacent notes in a music block and merges them into a single note with the pitch of one note and the combined tempo length of the two notes. In contrast, the Split operation selects one note and splits it into two of the same pitch and half the length of the original tempo. These two operations adjust the music block configuration locally, as shown in Figure 3.

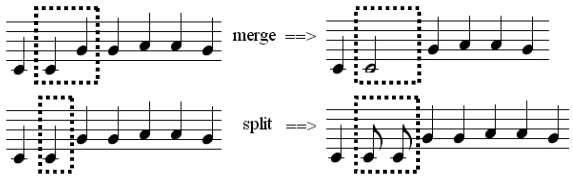

Figure 3: Operations: Merge and Split.

3.4.3 Double and Shorten

These operations operate on the tempo of notes. The Double operation uniformly doubles the tempo length of all the notes in a music block, and the Shorten operation makes the tempo length half, as shown in Figure 4.

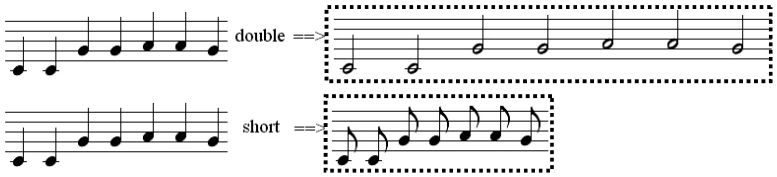

Figure 4: Operations: Double and Shorten.

3.4.4 Mutate and Raise

Different from the Double and Shorten operations, the Mutate and Raise operations only act on the pitch of notes. The note rises or falls in pitch. The Raise operation applies these changes uniformly to all the notes in a music block, while the Mutate operation only works on a randomly chosen note, as shown in Figure 5.

4 Auxiliary Functionalities

We implement a reference system based on the described CFE framework to automatically compose and customize music. By grading the music, users express their satisfactory degrees and train the evolutionary environment. After having a test drive, we find that the system needs to be enhanced for two reasons.

First, we should make the grading runs as few as possible. Our system is unlike common evolutionary computing applications which utilize programmed fitness functions. Our individuals are graded by the user. We have to take the human limitations and restrictions into consideration. Users may be tired with a large number of grading runs. As a consequence, we have to reduce the number of grading events.

Moreover, we would like to improve the capability of music composition. As the music composition in the real world, every type of music, such as

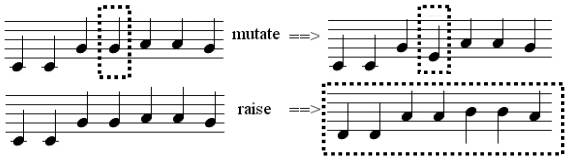

Figure 5: Operations: Mutate and Raise.

jazz, blues, and the like, has its own composition rules, styles, and guidelines. For this purpose, we embed some elements of the music theory into the system. In the following sections, we describe the enhancements to help the system compose music.

4.1 Reduce the Grading Runs

To reduce the grading runs, we design the following two mechanisms:

Block to block fitness table
The block to block fitness table is an $N \times N$ table, where N is an integer parameter, say 20. Considering the overhead, this table is unable to record all the fitness values of relations of each music block pair. Instead, the table records only the fitness values of block pairs which have a top-N fitness value in the music block pool. The table is also used to force two music blocks to be concatenated into one new music block if the fitness of their relation is higher than some specified threshold.

Adaptive evolution
For each grading event, our system can change the number of evolution rounds according to the diversity of the newly given scores. For example, the following two conditions with three scores:

- Condition 1: 30, 40, 90;
- Condition 2: 95, 85, 90.

For these two conditions, although their third scores are both 90, the third score in condition 1 is very different from the other two scores. The grade diversity in condition 1 is greater than that in condition 2. We assume that in condition 1, the third score reveals more information of the user preference. Hence, the system executes more evolutionary iterations for condition 1 than it does for condition 2.

4.2 Improve Music Composition

In order to improve music composition, we integrate the basics and elements of the music theory into the system. Our system can refer to the theoretical elements and compose music according to certain standards and/or common sense. In the present work , we employ only the fundamental elements and do not confine the variety of music styles.

Default note to note fitness table
In the system, there is a note to note fitness table. It records the fitness of relations of each note pair. During the system initialization, we set the pre-defined fitness into the note to note fitness table. We expect the default fitness table to help compose not-too-bad music at the early stage.

Music block repeat
A sequence of notes repeating in the whole song often occurs, such as that in “Happy Birthday” and in “Twinkle Twinkle Little Star”. We implement the Composition part to provide this feature. Thus, there are two options with different probabilities for choosing a music block to compose an unfinished music phrase:

- Select a new block which is in the music block pool but not in this unfinished phrase;
- Select an old block which appears in this unfinished phrase.

By doing so, the repeat of music blocks can be controlled by the probability parameter.

5 Showcases

Figures 6 to 13 demonstrate several showcases created by the system discussed in the article. These showcases indicate not only that the proposed framework can accomplish the goal for ordinary users to create music but also that the implemented system can create music of various types, styles, and lengths. If interested, the showcases, as MIDI files, can be accessed through the provided URLs. For more showcases, please visit nclab.tw/SM/2007/01.

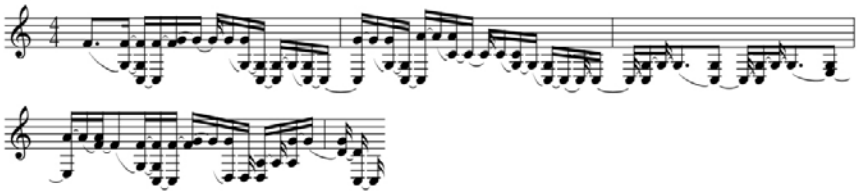

Figure 6: Listen to this showcase here.

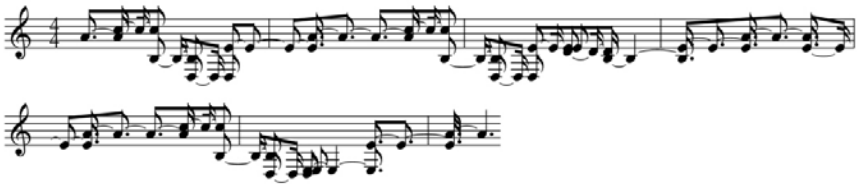

Figure 7: Listen to this showcase here.

6 Summary and Conclusions

We started by describing the motivation and the goal of this work. Inspired by previous studies in the literature and compelled by the need of having personalized music, we proposed the CFE framework. We presented the implementation of the system and the functionalities that we used to enhance the system. Finally, we provided several showcases to demonstrate that the proposed system can indeed accomplish its assigned task.

Our work shows that it is feasible and promising for computers to automatically compose customized or personalized music. Although the system currently acts only on short pieces of music, the design may be extended to compose longer music pieces, such as complete songs. The created music can be used in many applications, such as games, cellular phones, background music of web pages, and the like. With this system, everyone effectively has a private music composer at their service.

Acknowledgments

The work was partially sponsored by the National Science Council of Taiwan under grants NSC-95-2221-E-009-092 and NSC-95-2627-B-009-001 as well as by the MOE ATU Program. The author is grateful to the National Center for High-performance Computing for computer time and facilities.

Bibliography

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989, ISBN: 0201157675.

[2] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: The MIT Press, 1992, ISBN: 0262111705.

[3] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing Series. Springer, 2003, ISBN: 3-540-40184-9.

[4] J. A. Biles, "GenJam: Evolution of a jazz improviser," in *Creative evolutionary systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 165–187, ISBN: 1-55860-673-4.

[5] H.-C. Chang, "Applying genetic algorithms to music composition: Modeling and simulation of art systems," Master's thesis, National Chiao Tung University, Hsinchu, Taiwan, 2002.

[6] P. Laine and M. Kuuskankare, "Genetic algorithms in musical style oriented generation," in *Proceedings of the First IEEE Conference on Evolutionary Computation (ICEC-94)*, vol. 2, 1994, pp. 858–862.

[7] M. Marques, V. Oliveira, S. Vieira, and A. C. Rosa, "Music composition using genetic evolutionary algorithms," in *Proceedings of the 2000 Congress on Evolutionary Computation (CEC-2000)*, vol. 1, 2000, pp. 714–719.

[8] R. A. McIntyre, "Bach in a box: the evolution of four part baroque harmony using the genetic algorithm," in *Proceedings of the First IEEE Conference on Evolutionary Computation (ICEC-94)*, vol. 2, 1994, pp. 852–857.

[9] A. Pazos, A. Santos del Riego, J. Dorado, and J. J. Romero Caldalda, "Genetic music compositor," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC-99)*, vol. 2, 1999, pp. 885–890.

About the author

Ying-ping Chen received the B.S. degree and the M.S. degree in Computer Science and Information Engineering from National Taiwan University, Taiwan in 1995 and 1997, respectively, and the Ph.D. degree in 2004 from the Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, USA. He has been an Assistant Professor in the Department of Computer Science, National Chiao Tung University, Taiwan, since 2004. His research interests include linkage learning techniques, adaptive sampling methodologies, and novel real-world applications.

Homepage: nclab.tw
Email: ypchen@cs.nctu.edu.tw

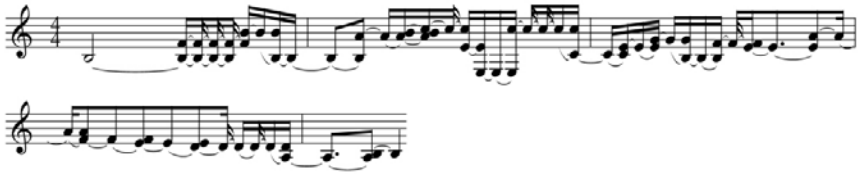

Figure 8: Listen to this showcase here.

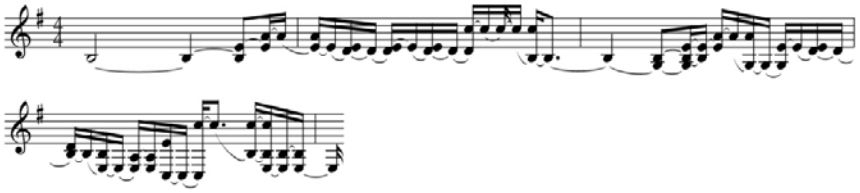

Figure 9: Listen to this showcase here.

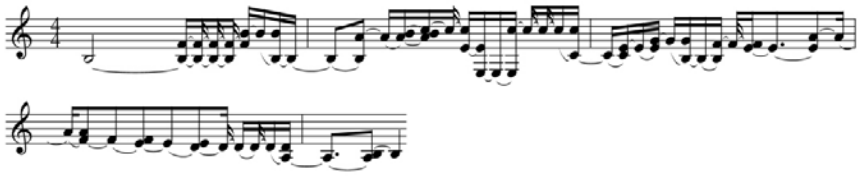

Figure 10: Listen to this showcase here.

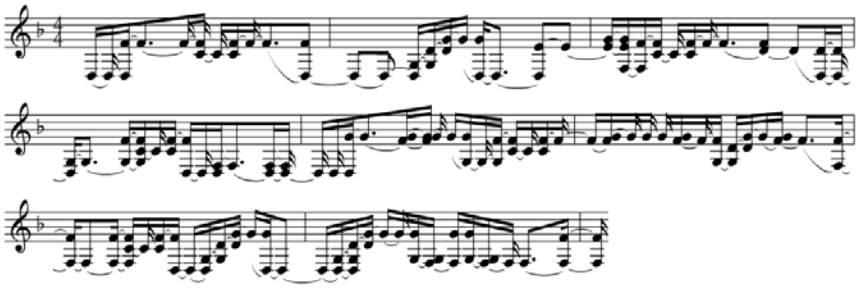

Figure 11: Listen to this showcase here.

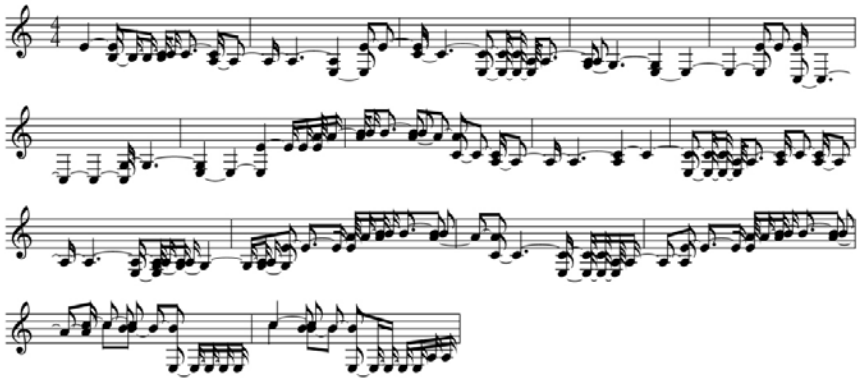

Figure 12: Listen to this showcase here.

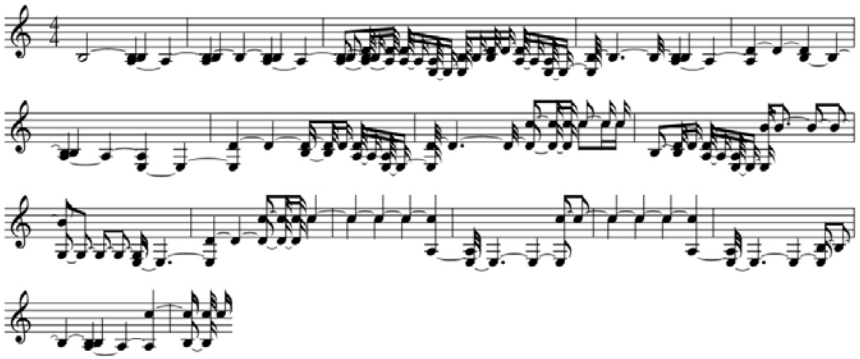

Figure 13: Listen to this showcase here.

GECCO-2007 Best Papers Nominees

In 2002, ISGEC created a best paper award for GECCO. As part of the double blind peer review, the reviewers were asked to nominate papers for best paper awards. We continue this tradition at GECCO-2007. The Track Chairs, Editor in Chief, and the Conference Chair nominated the papers that received the most nominations and/or the highest evaluation scores for consideration by the conference. The winners are chosen by secret ballot of the GECCO attendees after the papers have been orally presented at the conference. Best Paper winners are posted on the conference website. The titles and authors of all papers nominated as well as the page numbers where to find them in the Proceedings are given below:

Ant Colony Optimization and Swarm Intelligence, and Artificial Immune Systems

Analyzing Heuristic Performance with Response Surface Models: Prediction, Optimization and Robustness
Enda Ridge (The University of York)
Daniel Kudenko (The University of York)

Dendritic Cells for SYN Scan Detection
Julie Greensmith (University of Nottingham)
Uwe Aickelin (University of Nottingham)

Exact Analysis of the Sampling Distribution for the Canonical Particle Swarm Optimiser and its Convergence during Stagnation
Riccardo Poli (University of Essex)
David Broomhead (University of Manchester)

On the Runtime Analysis of the 1-ANT ACO Algorithm
Benjamin Doerr (Max-Planck-Institut für Informatik)
Frank Neumann (Max-Planck-Institut für Informatik)
Dirk Sudholt (Universität Dortmund)
Carsten Witt (Universität Dortmund)

Artificial Life, Evolutionary Robotics, Adaptive Behavior, Evolvable Hardware

Action-Selection and Crossover Strategies for Self-Modeling Machines
Josh Bongard (University of Vermont)

Hill Climbing on Discrete HIFF: Exploring the role of DNA Transposition in Long-term Artificial Evolution
Susan Khor (Concordia University)

The Effect of Learning on Life History Evolution
John A. Bullinaria (University of Birmingham)

Biological Applications

A Multi-Objective Approach to Discover Biclusters in Microarray Data
Federico Divina (Pablo de Olavide University)
Jesus S Aguilar-Ruiz (Pablo de Olavide University)

Discrimination of Metabolic Flux Profiles Using a Hybrid Evolutionary Algorithm
Stefan Bleuler (ETH Zurich)
Eckart Zitzler (ETH Zurich)

Parsimonious Regularization using Genetic Algorithms Applied to the Analysis of Analytical Ultracentrifugation Experiments
Emre H Brookes (University of Texas at San Antonio)
Borries Demeler (University of Texas Health Science Center at San Antonio)

Coevolution

Objective Fitness Correlation
Edwin D. De Jong (Utrecht University)

Optimal Nesting of Species for Exact Cover of Resources: Two against Many
Jeffrey Horn (Northern Michigan University)

Estimation of Distribution Algorithms

Cross Entropy and Adaptive Variance Scaling in Continuous EDA
Yunpeng Cai (Tsinghua University)
Xiaomin Sun (Tsinghua University)
Hua Xu (Tsinghua University)
Peifa Jia (Tsinghua University)

Population Sizing for Entropy-based Model Building in Discrete Estimation of Distribution Algorithms
Tian-Li Yu (National Taiwan University)
Kumara Sastry (University of Illinois at Urbana-Champaign)
David E. Goldberg (University of Illinois at Urbana-Champaign)
Martin Pelikan (University of Missouri-St. Louis)

Towards Billion-Bit Optimization via a Parallel Estimation of Distribution Algorithm
Kumara Sastry (University of Illinois at Urbana-Champaign)
David E. Goldberg (University of Illinois at Urbana-Champaign)
Xavier Llorà (University of Illinois at Urbana-Champaign)

Evolution Strategies, Evolutionary Programming

On the Use of Evolution Strategies for Optimising Certain Positive Definite Quadratic Forms
Dirk V. Arnold (Dalhousie University)

Performance Analysis of Niching Algorithms Based on Derandomized-ES Variants
Ofer M. Shir (Leiden University)
Thomas Bäck (Leiden University)

Reducing the Space-Time Complexity of the CMA-ES
James N Knight (Colorado State University)
Monte Lunacek (Colorado State University)

Evolutionary Multiobjective Optimization

Convergence of Stochastic Search Algorithms to Gap-Free Pareto FrontApproximations
Oliver Schütze (INRIA Futurs)
Marco Laumanns (ETH Zurich)
Emilia Tantar (INRIA Futurs)
Carlos A. Coello Coello (CINVESTAV-IPN)
El-ghazali Talbi (INRIA Futurs)

Exploring the Behavior of Building Blocks for Multi-Objective Variation Operator Design using Predator-Prey Dynamics
Christian Grimme (Dortmund University)
Joachim Lepping (Dortmund University)
Alexander Papaspyrou (Dortmund University)

Multiobjective Clustering with Automatic k-determination For Large-scale Data
Nobukazu Matake (Doshisha University Graduate School)
Tomoyuki Hiroyasu (Doshisha University)
Mitsunori Miki (Doshisha University)
Tomoharu Senda (Doshisha University)

SNDL-MOEA: Stored Non-Domination Level MOEA
Matt D Johnson (University of Missouri-Rolla)
Daniel R Tauritz (University of Missouri-Rolla)
Ralph W Wilkerson (University of Missouri-Rolla)

Formal Theory

Evolutionary Algorithms and Matroid Optimization Problems
Joachim Reichel (University of Dortmund)
Martin Skutella (University of Dortmund)

Generative and Developmental Systems

A Novel Generative Encoding for Exploiting Neural Network Sensor and Output Geometry
David B. D'Ambrosio (University of Central Florida)
Kenneth O. Stanley (Univeristy of Central Florida)

Acquiring Evolvability through Adaptive Representations
Joseph S Reisinger (The University of Texas at Austin)
Risto Miikkulainen (The University of Texas at Austin)

Methods for Open-box Analysis in Artificial Development
Adrian Grajdeanu (George Mason University)

Genetic Algorithms

Adjacency List Matchings — An Ideal Genotype for Cycle Covers
Benjamin Doerr (Max-Planck-Institut für Informatik)
Daniel Johannsen (Max-Planck-Institut für Informatik)

Empirical Analysis of Ideal Recombination on Random Decomposable Problems
Kumara Sastry (University of Illinois at Urbana-Champaign)
Martin Pelikan (University of Missouri at St. Louis)
David E. Goldberg (University of Illinois at Urbana-Champaign)

Extended Probe Method for Linkage Discovery over High-cardinality Alphabets
Shude Zhou (Tsinghua University)
Zengqi Sun (Tsinghua University)
Robert B Heckendorn (University of Idaho)

Fitness-Proportional Negative Slope Coefficient Hardness Measure for Genetic Algorithms
Riccardo Poli (University of Essex)
Leonardo Vanneschi (University of Milano-Bicocca)

Learning and Anticipation in Online Dynamic Optimization with Evolutionary Algorithms: The Stochastic Case
Peter A.N. Bosman
(Centre for Mathematics and Computer Science)
Han La Poutré
(Centre for Mathematics and Computer Science)

Rigorous Analyses of Simple Diversity Mechanisms
Tobias Friedrich (Max-Planck-Institut für Informatik)
Nils Hebbinghaus (Max-Planck-Institut für Informatik)
Frank Neumann (Max-Planck-Institut für Informatik)

Genetic Programming

Generalisation of the Limiting Distribution of Program Sizes in Tree-based Genetic Programming and Analysis of its Effects on Bloat
Stephen Dignum (University of Essex)
Riccardo Poli (University of Essex)

Genetic Programming for Cross-Task Knowledge Sharing
Wojciech Jaskowski (Poznan University of Technology)
Krzysztof Krawiec (Poznan University of Technology)
Bartosz Wieloch (Poznan University of Technology)

Learning Noise Michael D Schmidt (Cornell University)
Hod Lipson (Cornell University)

On the Constructiveness of Context-Aware Crossover
Hammad Majeed (University of Limerick)
Conor Ryan (University of Limerick)

Genetics-Based Machine Learning

Controlling Overfitting with Multi-Objective Support Vector Machines
Ingo Mierswa (University of Dortmund)

Empirical Analysis of Generalization and Learning in XCS with Gradient Descent
Pier Luca Lanzi (Politecnico di Milano)
Martin V Butz (University of Wurzburg)
David E. Goldberg (University of Illinois at Urbana Champaign)

Mixing Independent Classifiers
Jan Drugowitsch (University of Bath)
Alwyn M. Barry (University of Bath)

Real-World Applications

A Destructive Evolutionary Process A pilot Implementation
Joe Sullivan (Limerick Institute Of Technology)
Conor Ryan (University of Limerick)

An Evolutionary Keystroke Authentication Based on Ellipsoidal Hypothesis Space
Jae-Wook Lee (Seoul National University)
Sung-Soon Choi (Seoul National University)
Byung-Ro Moon (Seoul National University)

Coupling EA and High-level Metrics for the Automatic Generation of Test Blocks for Peripheral Cores
Leticia Bolzani (Politecnico di Torino)
Ernesto Sanchez (Politecnico di Torino)
Massimiliano Schillaci (Politecnico di Torino)
Giovanni Squillero (Politecnico di Torino)

Evolving Robust GP Solutions for Hedge Fund Stock Selection in Emerging Markets
Wei Yan (University College London)
Christopher D. Clack (University College London)

Multiobjective Network Design for Realistic Traffic Models
Nilanjan Nilanjan Banerjee (University of Massachusetts)
Rajeev Kumar (Indian Institute of Technology Kharagpur)

Real-Coded ECGA for Economic Dispatch
Chao-Hong Chen (National Chiao Tung University)
Ying-ping Chen (National Chiao Tung University)

Search-Based Software Engineering

Finding Safety Errors with ACO
Enrique Alba (University of Málaga)
Francisco Chicano (University of Málaga)

The Multi-Objective Next Release Problem
Yuanyuan Zhang (King's College London)
Mark Harman (King's College London)
S. Afshin Mansouri (King's College London)

The GECCO-2007 Competitions

Competition 1: Evolving trading rules

Goal: to evolve trading rules which maximize return of investment over a certain time span, using, as training data, closing values and volumes of a set of stocks over a longer period immediately preceding the one considered in the competition.

Instructions: Given the training and test files, reporting price and volume of 10 stocks over a period of about 5 years (soon available), competitors should produce:

A) The source code of a program which reads the training file and evolves a set of buy/sell trading rules

B) The source code of a program which, given an initial capital of $10000, reads the test file (of the same format as the training file), analyzes the test data and produces a log of all buy/sell transactions which would have occurred over the test time if the trading rules evolved by program A had been applied, as well as computing the difference (in percentage) between the final and initial values of the portfolio.

C) A report which provides details on the implementation of program A, as well as instructions on how to compile/run the two programs.

D) The executable file of programs A and B, statically compiled in order to be run (on a Linux-based or Windows-based computer) with no need of any other runtime library.

Every transaction will be charged commissions of 0.1% of its total value. In any single day, for each stock, no more than 40% of the average volume exchanged in the three previous days for that stock can be bought/sold.

Even if the files contain all data for the training and test time spans, decisions taken at any time t should be obviously based only on data acquired during the time interval [0, t-1].

Executables will be run on the test time series and a preliminary ranking of programs will be made. Even if a quantitative vealuation of perfomances will be made, as well as a ranking based on it, the FINAL decision on the award winner will be based BOTH on the results AND on the scientific quality and originality of the evolutionary approach by which the solution was generated.

Possible quality factors will be:

- autonomy (to what extent the program can be considered a human-competitive machine-generated solution)
- originality of the evolutionary approach
- quality of the documentation

File format
The file will be in CSV format, comprising 20 columns, which pairwise represent the traded volume and the closing price, respectively, for a set of 10 stocks. Each row contains data corresponding to a trading day.

The training data are available here.

Web resources
A good website for technical analysis with examples is stockcharts.com. The following link covers moving averages, which is the most basic technical indicator: moving averages.

A quite comprehensive list of technical indicators can be found here.

In terms of academic papers, the EDDIE project by Professor Edward Tsang at Essex is a good example.

All entries for the competition should be submitted, as a tar-gzipped or zipped attachment, **by email** to the address geccocomp@ce.unipr.it (which will be activated on May 4) specifying "**Entry for GECCO 2007 Competition n. 1**" as subject, **by June 15th**. Any enquiry regarding the competitions should also be emailed to the same address. For the most recent updates, please visit the competition page on the conference website.

Competition 2: Worst 1-MAX solver

Goal: to evolve the solution of the 1-MAX problem as late as possible within 1000 generations.

Instructions: contestants are required to design a generational evolutionary algorithm which solves the 1-MAX problem of size 15 in at least 95 runs out of 100, converging in less than 1000 generations but taking as many generations as possible to converge.

Population size must be limited: the program is allowed to make **at most 100000 fitness evaluations** within the allowed 1000 generations.

The score of each program will be based on the average number of generations needed for the successful runs to converge to the solution over five sets of 100 runs, each of which must run for at most 1000 generations after initialization . The random number generator will be seeded differently for each set. Programs which fail to converge to the solution in at least 5x95=475 runs over 500 will be disqualified.

Contestants are required to produce:

A) The source code of a program which runs 100 evolutions of a solution to the 1-MAX problem of size 15, starting from a random initialization seeded by an integer which must be provided as a command-line parameter at runtime.

B) A report which provides details on the implementation of the program, as well as instructions on how to compile/run the program.

C) The executable file of the program, statically compiled in order to be run (on a Linux-based or Windows-based computer) with no need of any other runtime library.

For the competition to be as fair as possible, we strongly recommend that the rand() and srand() functions of the GNU C (gcc) compiler be used in the programs. In any case, running 5 sets of experiments with different random seeds should reduce the dependency of results on the random number generator which is used.

Use of non-uniform random functions and explicit use of the notion of "time" by the algorithm (for example, to change the evolution operators depending on number of generations) is strictly forbidden. Programs are only allowed to let the values of evolution parameters change linearly with generations starting from a pre-set value.

Even if a quantitative vealuation of perfomances will be made, as well as a ranking based on it, the FINAL decision on the award winner will be based BOTH on the results AND on the scientific quality and originality of the evolutionary approach, and on the quality of documentation

All entries for the competition should be submitted, as a tar-gzipped or zipped attachment, **by email** to the address geccocomp@ce.unipr.it (which will be activated on May 4) specifying "**Entry for GECCO 2007 Competition n. 2**" as subject, **by June 15th**. Any enquiry regarding the competitions should also be emailed to the same address. For the most recent updates, please visit the competition page on the conference website.

Competition 3: Ant Wars

Goal: to evolve an ant which collects as much food as possible in a square toroidal grid environment in a pre-determined number of steps in the presence of a competing ant.

Instructions: contestants are required to evolve an ANSI-C function

```
int <MyAnt>_Move(int **grid, int my_row, int my_column)
```

`<My_Ant>` being the nickname each contestant must give to his/her ant, where `grid` is a pointer to an integer 2-dimensional array representing the status of cells in the grid; `my_row`, `my_column` are integers representing the coordinates of the current ant position, with cell grid[0,0] being the upper left-most cell of the grid, and scanning the grid row-wise (C-like array memory storage). The function must return the encoding of the ant's next move, defined as follows:

0 = move one step NW

1 = move one step N

2 = move one step NE

3 = move one step E

4 = move one step SE

5 = move one step S

6 = move one step SW

7 = move one step W

Contestants are required to design a generational evolutionary algorithm which solves the 1-MAX problem of size 15 in at least 95 runs out of 100, converging in less than 1000 generations but taking as many generations as possible to converge.

In each game, a random distribution of 15 pieces of food is generated.

Cell value:

- 0 corresponds to an empty cell
- 1 corresponds to a cell containing a piece of food
- 10 corresponds to the cell occupied by Ant 1
- 100 corresponds to the cell occupied by Ant 2

The size of the grid is 11x11 cells. A game lasts 35 moves per player. Ant 1 moves first. The coordinates of the starting cell for Ant 1 are (5, 2). The coordinates of the starting cell for Ant 2 are (5, 8) No piece of food can be located in the starting cells for the two ants. Each ant has a limited field of view consisting of a square neighborhood of size 5x5, the ant being located in its center.

The content of cells located outside the ant's field of view will be made unreliable (all zeroes or random numbers) when the grid is passed to the function using the pointer **grid, as the ant is by no means supposed to rely on that piece of information in deciding its next move.

If an ant moves into an empty cell nothing happens.

If an ant moves into a cell with food it scores 1 point and the cell becomes empty.

If an ant moves into the cell occupied by the opponent, it kills the opponent: no points are scored but only the survivor can go on moving until the end of the game.

Each match lasts 5 games: each game is won by the ant that reaches the highest score. In case of a tie, Ant 1 is the winner. The match winner is the first player to win 3 games.

Each game corresponds to a new run of the program which calls the <MyAnt>_Move function, so possible static variables will be re-initialized at every game restart.

In the first four games each contestant plays Ant 1 and Ant 2 alternatively. In the 5th game the player with the highest total score in the first 4 games plays Ant 1. In case both players have reached the same total score, the player with the highest score in a single game plays Ant 1. If a further tie occurs, the ants are assigned randomly.

A round robin tournament will be played between all entries. In case of tie at the end of the tournament, a tie-break game/tournament will be played between the top-ranked entries.

Contestants are required to produce:

A) A text file, no longer than 5Kbytes, containing the ANSI-C code of the function. The names of the variables used as function parameters MUST BE THE SAME as in the above function prototype.

B) A report which provides details on how the program was evolved, as well as its 'original' encoding within the evolutionary framework within which it was evolved (e.g., the tree-based representation of the program, if evolved using tree-based GP).

Even if a winner of the tournament will be determined after playing the games, the FINAL decision on the award winner will be based BOTH on the results AND on the scientific quality and originality of the evolutionary approach by which the program was generetaed.

Possible quality factors will be:

- autonomy (to what extent the program can be considered a human-competitive machine-generated solution)
- originality of the evolutionary approach
- quality of the documentation

All entries for the competition should be submitted, as a tar-gzipped or zipped attachment, **by email** to the address geccocomp@ce.unipr.it (which will be activated on May 4) specifying "**Entry for GECCO 2007 Competition n. 3**" as subject, **by June 15th**. Any enquiry regarding the competitions should also be emailed to the same address. For the most recent updates, please visit the competition page on the conference website.

GECCO-2007 Workshops

Saturday, July 7 and Sunday, July 8, 2007

The GECCO-2007 workshops will be held on Saturday, July 7 and Sunday, July 8, 2007. The proceedings will be published on CD-ROM, and distributed at the conference.

Particle Swarms: The Second Decade
Organized by Riccardo Poli, Jim Kennedy, Tim Blackwell, and Alex Freitas
Duration: Half Day
[summary | details]

Open-Source Software for Applied Genetic and Evolutionary Computation (SoftGEC)
Organized by Jason H. Moore
Duration: 2-hours
[summary | details]

Optimization by Building and Using Probabilistic Models
Organized by Kumara Sastry and Martin Pelikan.
Duration:Half Day
[summary | details]

Graduate Student Workshop
Organized by Anikó Ekárt
Duration:Full Day
[summary]

Undergraduate Student Workshop
Organized by Laurence Merkle, Clare Bates Congdon and Frank Moore.
Duration: Half Day
[summary]

Evolutionary Algorithms for Dynamic Optimization Problems
Organized by Peter A.N. Bosman and Jürgen Branke
Duration: Half Day
[summary | details]

Parallel Bioinspired Algorithms
Organized by Francisco Fernández and Erick Cantú-Paz
Duration: Half Day
[summary | details]

Learning Classifier Systems
Organized by Jaume Bacardit, Ester Bernadó-Mansilla, Martin V. Butz
Duration: Full Day
[summary | details]

Evolutionary Computation & Multi-Agent Systems and Simulation (ECoMASS)
Organized by Bill Rand, Sevan G. Ficici
Duration: Half Day
[summary | details]

Petroleum Applications of Evolutionary Computation
Organized by Alexandre Castellini, Charles Guthrie, David Wilkinson, Burak Yeten, Tina Yu
Duration: Half Day
[summary | details]

Defense Applications of Computational Intelligence
Organized by Frank Moore, Laurence D. Merkle, Stephen C. Upton
Duration: Full Day
[summary | details]

The Evolution of Natural and Artificial Systems
Metaphors and Analogies in Single and Multi-Objective Problems
Organized by Ami Moshaiov, Steven Hecht Orzack, Joshua Knowles
Duration: Half Day

Medical Applications of Genetic and Evolutionary Computation
Organized by Stephen L. Smith, Stefano Cagnoni
Duration: Half-day
[summary | details]

FX-SBSE - Foundations and cross cutting issues in Search Based Software Engineering
Organized by Mark Harman, John Clark, Xin Yao, Joachim Wegener, Christine McCulloch, Tanja Vos
Duration: Half-day

User-centric Evolutionary Computation
Organized by Iam Parmee
Duration: Half-day
[summary | details]

FD-ET: Future Directions in Evolutionary Testing
Organized by Mark Harman, John Clark, Xin Yao, Joachim Wegener, Christine McCulloch, Tanja Vos
Duration: Half-day

GECCO-2007 Tutorials

Saturday, July 7 and Sunday, July 8, 2007

Introductory Tutorials

Genetic Algorithms, Erik Goodman

Genetic Programming, John Koza

Evolution Strategies, Thomas Bäck

A Unified Approach to EC, Kenneth De Jong

Ant Colony Optimization, Christian Blum

Learning Classifier Systems, Martin V. Butz

Probabilistic Model-Building GAs, Martin Pelikan

Grammatical evolution, Conor Ryan

Coevolution, E. de Jong, K. Stanley, & P. Wiegand

Particle Swarm Optimization, A. Engelbrecht & X. Li

Beowulf Clusters for EC, A. Khoshla, P.K. Singh, & D.G. Chowdhary

Advanced Tutorials

GA Theory, Jonathan Rowe

GP Theory, R. Poli, B. Langdon

Representations for Evolutionary Algorithms, Franz Rothlauf

No Free Lunch, Darrell Whitley

Bioinformatics, Jason Moore

Human Competitive Results, John Koza

Evolutionary Multiobjective Optimization, E. Zitzler & K. Deb

Industrial Evolutionary Computation, A. Kordon, G. Smits, & Mark Kotanchek

Constraint Handling Techniques Used with EAs, Carlos Coello Coello

Statistics for EC, Mark Wineberg

Coevolution, S. Ficici, A. Bucci

Evolutionary Practical Optimisation, Kalyanmoy Deb

Computational Complexity and EC, T. Jansen, F. Neumann

Complex networks and EC, J.J. Merelo, Carlos Cotta

Particle Swarm Optimization for Fuzzy Models, Arun Khosla

Fitness Landscapes and Problem Hardness in EC, L. Vanneschi, S. Verel

An Information Perspective on EC, Yossi Borenstein

Specialized Techniques and Applications

Experimental Research in EC, M. Preuss, T.B. Beielstein

Symbolic Regression in GP, Maarten Keijzer

Evolutionary Neural Networks, Risto Miikkulainen

Quantum Computing, Lee Spector

Evolvable Hardware, Lukas Sekanina

Artificial Development, P. Haddow, G. Tufte

Evolutionary Games, Marco Tomassini

Evolutionary Design, Ian Parmee

Evolutionary Multiobjective Combinatorial Optimization, Rajeev Kumar

Evolving Neural Network Ensembles, Xin Yao

Evolutionary Computer Vision, Gustavo Olague

Bio-inspired Telecommunications, Muddassar Farooq

Distributed EC for Fun and Profit, J.J. Merelo, J.L.J. Laredo

Announcements

PhD Position in Mathematical Modelling and Genetic Algorithms

From James Smith (James.Smith@uwe.ac.uk)

Deadline May 30th 2007

The University of the West of England, Bristol, in collaboration with the Uk Office of National Statistics has an EPSRC-funded PhD position in Mathematical Modelling and Evolutionary algorithms. Further details may be found at info.uwe.ac.uk/vacancies/job_details.asp?ref=PHD/030507

EPSRC PhD Studentship / Scholarship

From Prof. Xin Yao (X.Yao@cs.bham.ac.uk)

Deadline: Until the post is filled.

Fully-funded EPSRC PhD Studentship / Scholarship at CERCIA, the University of Birmingham, UK (WWW)

Title: Study of fitness landscapes arising from software engineering problems.

In this project, the successful applicant will study and characterise fitness landscapes arising from software engineering (SE) problems, e.g., through statistical and visualisation techniques. In doing so, the applicant will draw upon and extend advanced landscape characterisation methodology from diverse fields, including evolutionary computation, biology, chemistry and physics. This research will complement ongoing theoretical research in the group on the runtime analysis of evolutionary algorithms on SE problems.

The successful applicant must have at least a 2.1 or above degree in computer science or a closely related field. We do consider outstanding applicants from mathematics, physics and engineering as long as they have sufficient computer science background. A good knowledge of optimisation is required, especially a good understanding and practical experience with modern meta-heuristics, including evolutionary algorithms, estimation of distribution algorithms, simulated annealing and others. The successful applicant must have excellent programming skills and knowledge of software engineering. S/he must be an excellent team player who can work independently and communicate well with others.

The Studentship / Scholarship
Duration: Up to three years (subject to satisfactory progress).
Payments: Living expenses: minimum GBP12,500 per year tax-free; plus tuition fee fully paid.
Eligibility: Open to European Union/United Kingdom applicants only.
Deadline: Until the post is filled.

How to Apply
For more information on the PhD programme at Birmingham: www.cs.bham.ac.uk/admissions/postgraduate-research/.

For more information about the project: www.cercia.ac.uk/projects/research/SEBASE/.

For technical enquiries only: Prof. Xin Yao (X.Yao@cs.bham.ac.uk)

New Journal: SWARM INTELLIGENCE

From Marco Dorigo (mdorigo@ulb.ac.be)

The first issue of "Swarm Intelligence", a new quarterly journal published by Springer, is forecast for the Summer 2007.

Information on the journal, including its aims and scope, is available at www.springer.com/11721

Papers can be submitted using Springer's Editorial Manager at www.editorialmanager.com/swrm/

Contributions are welcome!

Tenure-Track Position in Intelligent Systems

From Prof. Luca Maria Gambardella (luca@idsia.ch)
IDSIA - Istituto Dalle Molle di Studi sull'Intelligenza Artificiale
Manno-Lugano, Switzerland

Homepage: WWW
Starting: September 2007.
Deadline May 30th 2007

We are seeking an outstanding highly motivated and talented researcher scientist in Intelligent Systems with a strong scientific background in one or more of the following disciplines:

- Metaheuristics
  (with emphasis on ant colony optimization)
- Optimization & Simulation
  (with emphasis on transportation and vehicle routing problems)
- Robotics
  (with emphasis on autonomous and swarm robotics)

The successful candidate must have proved research and teaching experiences and the ability to set-up, drive and motivate a research team.

Candidates for this position must hold a doctoral degree in Operations Research/Simulation/Computer Science/Artificial Intelligence/Robotics. Candidate will be involved in research projects in the previously mentioned areas and, when is needed, in bachelor, master and PhD teaching activities.

Theoretical and applied knowledge is required.

Attractive Swiss salary.

Applicants should submit:

1. Detailed curriculum vitae,
2. List of three references (including their email addresses),
3. Statement on how their research interests fit the above topics (1-2 pages).

Please submit your application by email to

Prof. Luca Maria Gambardella
IDSIA, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale
Galleria 2
6928 Manno-Lugano
Switzerland
Phone : +41 91-6108663, Fax : +41 91-61208661
Homepage: www.idsia.ch/luca
Email: luca@idsia.ch

First Summer Course on Future Directions in Soft Computing

The first summer course on Future Directions in Soft Computing will take place in Mieres, Asturias, Spain, July 9 - 13, 2007, and is organized by the European Centre for Soft Computing.

The summer course reviews the fundamentals of Soft Computing, describes many real-world applications, and, in particular, treats new trends and future directions of the field. Participants will gain insight into the potential of soft computing techniques and the state of the art in the area. To achieve this, the lecturers have been selected from the leaders of the different branches of Soft Computing.

The course will cover the following topics:

- Fundamentals and New Trends on Fuzzy Set Theory - Fuzzy Systems
- Fundamentals and New Trends on Evolutionary Computation
- Fundamentals and New Trends on Neural Networks
- Computing with Words and Linguistic Data Mining
- Fusion of Soft Computing Tools
- Applications of Soft Computing

Lecturers will be

- Piero Bonissone (General Electric R&D)
- Christian Borgelt (ECSC)
- Oscar Cordon (ECSC)
- Francisco Herrera (University of Granada)

- Bob John (Monfort University)
- Janusz Kacprzyk (Polish Academy of Sciences)
- Frank Klawonn (University of Applied Sciences Braunschweig/Wolfenbuttel)
- Rudolph Kruse (University of Magdeburg)
- Pedro Larranaga (University of the Basque Country)
- Claudio Moraga (ECSC)
- Enric Trillas (ECSC)

For additional information visit the web page
www.softcomputing.es/summercourse/

Dissertation Corner

Competent Program Evolution

Doctoral Thesis by Moshe Looks

Heuristic optimization methods are adaptive when they sample problem solutions based on knowledge of the search space gathered from past sampling. Recently, competent evolutionary optimization methods have been developed that adapt via probabilistic modeling of the search space (sometimes known as estimation-of-distribution algorithms, EDAs). However, their effectiveness requires the existence of a compact problem decomposition in terms of prespecified solution parameters.

How can we use these techniques to effectively and reliably solve program learning problems, given that program spaces will rarely have compact decompositions? One method is to manually build a problem-specific representation that is more tractable than the general space. But can this process be automated? **My thesis is that the properties of programs and program spaces can be leveraged as inductive bias to reduce the burden of manual representation-building, leading to competent program evolution.**

The central contributions of my dissertation are a synthesis of the requirements for competent program evolution, and the design of a procedure, meta-optimizing semantic evolutionary search (MOSES), that meets these requirements. MOSES is an estimation-of-distribution program evolution system distinguished by two key mechanisms: (1) exploiting semantics (what programs actually mean) to restrict and direct search; and (2) limiting the recombination of programs to occur within bounded subspaces (constructed on the basis of program semantics). This occurs through a framework of representation-building – a representation in MOSES an examplar program, together with a set of possible program transformations that are expected to lead to semantically nearby programs. The underlying EDA learns how to intelligently combine these transformations in order to reach new, higher scoring programs (even if none of the programs resulting from a single transformation alone have higher scores).

It is demonstrated that representation-building can dramatically change important properties of program spaces that impact problem difficulty (e.g., fitness-distance correlation). Experimental results are provided to analyze and verify the effectiveness of MOSES, demonstrating superior performance and scalability across a range of program evolution problems in comparison to purely syntactic techniques (i.e., local search and genetic programming), without any bloating of program size. Supervised classification results for problems from computational biology are also presented, where MOSES achieves accuracies comparable to support vector machines (and superior to genetic programming), but with extremely simple and comprehensible classification rules.

Moshe Looks has a Ph.D. and M.S. from Washington University in St. Louis and a B.Sc. from the Hebrew University of Jerusalem. Beyond probabilistic approaches to program evolution, his main research interest is artificial general intelligence – the design and analysis of integrative learning and reasoning systems at the intersection of AI and cognitive science. He is currently employed at SAIC (Science Applications International Corporation), architecting a natural language processing system for national security applications that uses MOSES for document classification and analysis.

Homepage: metacog.org
Dissertation: metacog.org/main.pdf

Forthcoming Papers

Evolutionary Computation 15(1)

- **Generalization in the XCSF Classifier System: Analysis, Improvement, and Extension**, Pier Luca Lanzi, Daniele Loiacono, Stewart W. Wilson, and David E. Goldberg, pp 133–168
- **An Information-Theoretic Analysis on the Interactions of Variables in Combinatorial Optimization Problems**, Dong-Il Seo and Byung-Ro Moon, pp 169–198
- **Reducing the Number of Fitness Evaluations in Graph Genetic Programming Using a Canonical Graph Indexed Database**, Jens Niehaus, Christian Igel, and Wolfgang Banzhaf, pp 199–222
- **Revisiting Negative Selection Algorithms**, Zhou Ji and Dipankar Dasgupta, pp 223–251

Artificial Life Journal 13(1)

- **Simple Models of Assortment Through Environmental Feedback**, John W. Pepper
- **Evolvable Self-Reproducing Cells in a Two-dimensional Artificial Chemistry** Tim J. Hutton
- **The Dynamic Changes in Roles of Learning Through the Baldwin Effect**, Reiji Suzuki and Takaya Arita
- **The Effects of Cultural Learning in Populations of Neural Networks**, Dara Curran and Colm O'Riordan
- **Using the XCS Classifier System for Multi-objective Reinforcement Learning Problems**, Matthew Studley and Larry Bull
- **An Artificial Ecosystem: Emergent Dynamics and Life-like Properties**, Mauno Rönkkö
- Book review: **Endless Forms Most Beautiful**, authored by Sean B. Carroll, reviewed by Douglas H. Erwin
- Book review: **Live Evolving Molecules, Mind, and Meaning** authored by Christian De Duve, reviewed by Carlos Gershenson

Artificial Life Journal 13(2)

- **A Synthetic Vision System Using Directionally Selective Motion Detectors to Recognize Collision**, Shigang Yue and F. Claire Rind
- **Synchronization Phenomena in Surface-Reaction Models of Protocells**, Roberto Serra, Timoteo Carletti, and Irene Poli
- **Evolving Virtual Creatures and Catapults**, Nicolas Chaumont, Richard Egli, and Christoph Adami
- **An Artificial Ecosystem: Emergent Dynamics and Life-Like Properties**, Mauno Rönkkö
- **"Life is a Verb": Inflections of Artificial Life in Cultural Context** Stefan Helmreich
- Book review: **Reconstructing the Cognitive World. the Next Step** by Michael Wheeler, reviewed by Ezequiel A. Di Paolo
- Book Review: **Weak Links: Stabilizers of Complex Systems from Proteins to Social Networks**, authored by Peter Csermely, reviewed by Jennifer H. Watkins and Marko A. Rodriguez
- Book Review: **Artificial Life Models in Software**, authored by Andrew Adamatzky and Maciej Komosinski, reviewed by Hiroki Sayama
- Software Review: **NeTLogo, a multiagent simulation environment**, authored by Uri Wilensky, reviewed by Elizabeth Sklar

Calls and Calendar

July 2007

Genetic and Evolutionary Computation Conference (GECCO-2007)

July 7-11, 2007, University College London, London, UK

Homepage: WWW

The Genetic and Evolutionary Computation Conference (GECCO-2007) will present the latest high-quality results in the growing field of genetic and evolutionary computation. Topics include: genetic algorithms, genetic programming, evolution strategies, evolutionary programming, real-world applications, learning classifier systems and other genetics-based machine learning, evolvable hardware, artificial life, adaptive behavior, ant colony optimization, swarm intelligence, biological applications, evolutionary robotics, coevolution, artificial immune systems, and more.

Keynote Event

On Monday evening, 9 July 2007, Professors Richard Dawkins, Lewis Wolpert, and Steve Jones will take part in a public debate, discussing the emergence of complexity in evolution. This will be a once-in-a-lifetime opportunity to hear and interact with some of the most famous names in evolutionary biology.

Program Tracks

Three days of presentations in 15 separate and independent program tracks specializing in various aspects of genetic and evolutionary computation. Proceedings will be published and distributed to all registered attendees.

Free Tutorials and Workshops

Two days of free tutorials and workshops (included with conference registration) presented by some of the world's foremost experts in topics of interest to genetic and evolutionary computation researchers and practitioners.

Soft Computing Journal
Special Issue on "Emerging Trends in Soft Computing Memetic Algorithms"

Submission Deadline July 01, 2007

Authors Notification: December 01, 2007

Camera-ready Submission: Spring 2008

One of the recent growing areas in Evolutionary Algorithm (EAs) research is Memetic Algorithms (MAs). MAs are population-based meta-heuristic search methods inspired by Darwinian principles of natural evolution and Dawkins notion of a meme defined as a unit of cultural evolution that is capable of local refinements. Recent studies on MAs have revealed their successes on a wide variety of real world problems. Particularly, they not only converge to high quality solutions, but also search more efficiently than their conventional counterparts. In diverse contexts, MAs are also commonly known as hybrid EAs, Baldwinian EAs, Lamarkian EAs, cultural algorithms and genetic local search.

The aim of this special issue is to reflect the most recent advances in the field, and increase the awareness of the computing community at large on this effective technology. In particular, we endeavor to demonstrate the current state-of-the-art in the theory and practice of MAs. Topics of interests include (but are not limited to):

- Novel competitive, collaborative and cooperative frameworks of MAs,
- Analytical and/or theoretical studies that enhance our understanding on the behaviors of MAs,
- Using multiple memes or local searchers or exact methods,
- Adaptive MAs (e.g., meta-Lamarckian and meta-Baldwinian),
- Multi-objective MAs,
- Asymptotic global convergence analyses and/or complexity analyses of MAs,
- MAs for discrete, continuous and dynamic optimization problems,

- Surrogate-assisted MAs or MAs using approximation methods,
- MA methodologies for computationally expensive optimization problems,
- Knowledge incorporation in MAs,
- Real-world applications of MAs.

All electronic submissions must be sent to Dr. Yew-Soon Ong at asysong@ntu.edu.sg. Manuscripts should conform to the standard format of the Soft Computing journal as indicated in the Information for Authors at WWW. All submissions will be peer reviewed subject to the standards of the journal.

Manuscripts based on previously published conference papers, e.g., CEC, GECCO or otherwise, must be extended substantially. Electronic submissions in postscript or PDF are strongly preferred.

Enquiries on this special issue can be directed to any of the editors. Prospective authors are also invited to send an email indicating their interest in submitting a paper and the specific topics addressed.

Guest Editors

- Dr. Yew-Soon Ong, School of Computer Engineering,
  Nanyang Technological University, Singapore
  E-mail: asysong@ntu.edu.sg
  Homepage: www.ntu.edu.sg/home/asysong/
- Dr. Meng-Hiot Lim, School of Electrical and Electronics Engineering,
  Nanyang Technological, University, Singapore
  Email: emhlim@ntu.edu.sg
  Homepage: www.ntu.edu.sg/home/emhlim/
- Dr. Ferrante Neri, University of Jyväskylä, Finland
  Email: neferran@cc.jyu.fi
  Homepage: http://people.cc.jyu.fi/ neferran/
- Dr. Hisao Ishibuchi, Osaka Prefecture University, Japan
  E-mail: hisaoi@ie.osakafu-u.ac.jp
  Homepage: www.ie.osakafu-u.ac.jp/˜hisaoi/ci_lab_e/index.html

Journal of Neural Computing and Applications
Special Issue on "Artificial Immune Systems: Theory and Applications"

Submission Deadline July 31, 2007
Authors Notification: November 30, 2007
Camera-ready Submission: Spring 2008

Natural immune systems are sophisticated and complex defense and maintenance systems with remarkable capabilities, such as learning, memory, and adaptation. The immune system can be viewed in two parts: the innate immune system and the adaptive immune system. The innate immune system is inherited from birth, and endows on the host non-specific recognition of pathogenic material. The adaptive (or acquired) immune system is afforded by the specific recognition of pathogenic material, and adapts over the lifetime of the host through a process of cloning, mutation, and selection. During the past decade, numerous novel computational models and algorithms have been developed based on such immunological principles. Artificial Immune Systems (AIS), inspired by the natural immune systems, are an emerging kind of soft computing paradigm. Applied to a wide variety of applications, the AIS have recently gained considerable research interest from different communities. Their successful industry applications include robotics, optimization, fault tolerance, process control, etc.

This special issue focuses on presenting the latest work in the theory and applications of the artificial immune systems. The topics of interest for this special issue include, but are not limited to:

- Immunological modeling
- Population and network based immune algorithms
- Architectures and frameworks inspired by immune systems
- Novel developments in AIS, such as danger theory and cognitive immune paradigm
- Applications of AIS (including industrial employment of artificial immune algorithms)
- Hardware implementation of AIS Immunoinformatics
- Fusion of artificial immune systems and other soft computing methods Theoretical analysis of AIS

Prospective authors are invited to submit their full papers to the guest editors before the deadline.

Guest Editors

- Dr. Xiao-Zhi Gao, Helsinki University of Technology, Finland
  gao@cc.hut.fi
- Prof. Mo-Yuen Chow, North Carolina State University, USA
  chow@ncsu.edu
- Prof. David Pelta, University of Granada, Spain
  dpelta@decsai.ugr.es
- Dr. Jon Timmis, University of York, UK
  jtimmis@cs.york.ac.uk

September 2007

Ninth European Conference on Artificial Life

September 10-14, 2007, Lisbon, Portugal.

Homepage: WWW

The conference will take place in a splendidly situated, historic location of Lisbon [take a tour around the Venue and the City!] and will comprise a one-track main session, several workshops, tutorials and associated events.

Artificial Life aims at the study of all phenomena characteristic of natural living systems, through methodologies of synthesis implemented in computational, robotic or other artificial architectures. Its wide scope ranges from the investigation of how life or life-like properties develop from inorganic components to how cognitive processes emerge in natural or artificial systems.

The “European” in European Conference on Artificial Life - ECAL, merely refers to the conference location, but participation is worldwide. In this ECAL we envisage maintaining and enlarging this worldwide scope and want to emphatically encourage novelty and daring ideas, particularly amongst young researchers. Both technical and conceptual work is welcome.

We want the conference to be of importance not only to the participating researchers but also to the general public with an interest in science. Hence a diversity of parallel open events will be promoted in venues throughout the city of Lisbon, aimed at communicating A-Life ideas, and the scientific practice and consequences, to a broader audience.

A further focus will be the involvement of industry. Relevant space and time will be allocated to the presentation of demos. This will constitute an opportunity to involve people from industry, inviting them to be present and sponsoring a prize to award particularly relevant work.

ECAL 2007 Workshop on Machine Epigenesis

September 10, 2007, Lisbon, Portugal.

Homepage: WWW

Creating a machine that exhibits life-like behavior has been the very core motivation of Artificial Life since its onset. Self-replication has remained the prime study since von Neumann, however, biological systems show a far wider range of generative behavior, including differentiation and morphogenesis of multicellular structures from a single zygote, and adaptive de-differentiation and regeneration of parts in case of system failure. These characters remain largely missing in manmade, engineered systems, as well indicated by the late John Maynard-Smith in his writing:

"One reason why we find it so hard to understand the development of form may be that we do not make machines that develop: often we understand biological phenomena only when we have invented machines with similar properties... [and] we do not make ’embryo’ machines ..." - John Maynard-Smith, The Problems of Biology (1986)

The Workshop on Machine Epigenesis aims to address this issue – the means, methods and models of machine epigenesis. It is expected to establish a field of research on any constructional and epigenetic processes of machines and to initiate a collective effort of formalization of models of such epigenetic machines. Here a "machine" is broadly construed to include abstract automata, electro-mechanical devices, molecular structures, and any other physical or informational instantiation. Topics to be covered in the workshop include (but are not limited to):

- Formal theories and abstract models of machine epigenesis
- Theories of universal and non-universal constructors
- Physical implementation of epigenetic machines
- Self-replicating and self-repairing machines

- Self-organization in modular and swarm robots
- Biological analogs relevant to the realization of machine epigenesis
- Philosophical and ethical issues in creating epigenetic machines
- Extending the mechanist model of living systems - philosophy

IEEE Congress on Evolutionary Computation

September 25-28, 2007, Singapore.

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.

The 7th International Conference on Evolvable Systems: From Biology to Hardware

September 21-23, 2007, Wuhan, China

Homepage: WWW

The 7th International Conference on Evolvable Systems: From Biology to Hardware (ICES 2007) will be held on September 21-23, 2007 at Wuhan, China.

ICES 2007 will address the theme "From Laboratory to Real World", explaining how to shorten the gap between evolvable hardware research and design for real-world applications. Cross-fertilization of evolvable hardware, intelligent computation and newly emerging technologies is strongly encouraged. It will feature world-renowned plenary speakers, state-of-the-art special sessions, regular technical sessions, poster interactions, and entertaining social activities.

International Symposium on Intelligence Computation and Applications (ISICA 2007)

September 21-23, 2007, Wuhan, PRC,

Homepage: WWW

The 2nd International Symposium on Intelligence Computation and Applications (ISICA 2007) will be held on September 21-23, 2007 in Wuhan , China, at the same time as the 7th International Conference on Evolvable Systems: From Biology To Hardware (ICES 2007).

In order to cover both high-level and most up-to-date results, the ISICA Program Committee plan to publish two separate proceedings. Papers from the oral presentations will be published in the Lecture Notes by the Springer. These papers will emphasise the development of theories and methodologies in the field of computational intelligence.

Papers from the poster sessions will be collected in a separate proceedings which will be published by the China University of Geosciences Press. The ISICA 2005 proceedings were also published by the China University of Geosciences Press, and have been accepted into the Index To Scientific & Technical Proceedings (ISTP). Papers in the poster sessions will focus on innovative applications of computational intelligence.

All papers in PDF format should be submitted electronically through the conference website. The manuscripts should be written in English and follow the LNCS format provided by Springer). Full papers are limited to maximum 8 pages.

October 2007

7th International Conference on Intelligent Systems Design and Applications (ISDA'07)

October 22-24, 2007, Rio de Janeiro, Brazil

Homepage: WWW

Intelligent Systems Design and Applications (ISDA'07) is the 7th International conference that brings together international soft computing, artificial intelligence, computational intelligence researchers, developers, practitioners and users. The aim of ISDA'07 is to serve as a forum to present current and future work as well as to exchange research ideas in this field.

ISDA'07 will focus on the following topics:

- Intelligent Systems Architectures and Applications
- Intelligent Image and Signal Processing
- Intelligent Internet Modeling
- Intelligent Data mining
- Intelligent Business Systems
- Intelligent Control and Automation
- Intelligent Agents
- Intelligent Knowledge Management

Prospective authors are invited to submit a full paper of 8 pages (PDF), for oral presentation. Authors must use the double columns IEEE format.

The submission of a paper implies that the paper is original and has not been submitted under review or copyright protected by the author if accepted. Besides papers in regular sessions, papers in special sessions are also invited to provide forums for focused discussions on new topics and innovative applications of established approaches. A special session consists of at least four related papers.

All papers should be submitted electronically via Online Paper Submission System (electronic link available). The format of the initial submissions can be PDF. The file of the final accepted papers should be in either Word or Latex.

June 2008

2008 IEEE World Congress on Computational Intelligence

June 1-6, 2008, Hong Kong

Homepage: WWW

Deadline December 1, 2007

The 2008 IEEE World Congress on Computational Intelligence (WCCI 2008) will be held at the Hong Kong Convention and Exhibition Centre during June 1-6, 2008. Sponsored by the IEEE Computational Intelligence Society, co-sponsored by the International Neural Network Society, Evolutionary Programming Society and the Institution of Engineering and Technology, WCCI 2008 is composed of the 2008 International Joint Conference on Neural Networks (IJCNN 2008), the 2008 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008) and the 2008 IEEE Congress on Evolutionary Computation (CEC 2008). WCCI 2008 will be the fifth milestone in this series with a glorious history from WCCI 1994 in Orlando, WCCI 1998 in Anchorage, WCCI 2002 in Honolulu, to WCCI 2006 in Vancouver.

Call for Contributed Papers

Researchers are invited to contribute high-quality papers to WCCI 2008. All papers are to be submitted electronically through the Congress website by December 1, 2007. All submitted papers will be refereed by experts in the fields based on the criteria of originality, significance, quality, and clarity. For inquiries, contact IJCNN2008 Program Chair Derong Liu at dliu@ece.uic.edu, FUZZ-IEEE2008 Program Chair Gary Feng at megfeng@cityu.edu.hk, or CEC2008 Program Chair Zbigniew Michalewicz at zbyszek@cs.adelaide.edu.au.

Call for Special Sessions

WCCI 2008 Program Committees solicit proposals for special sessions within the technical scopes of the three conferences. Special sessions, to be organized by internationally recognized experts, aim to bring together researchers in special focused topics. Cross-fertilization of the three research areas of computational intelligence with new emerging technologies is strongly encouraged. Papers submitted for special sessions are to be peer-reviewed with the same criteria used for the contributed papers. Researchers interested in organizing special sessions are invited to submit formal proposals to IJCNN2008 Special Sessions Chair Jagath C. Rajapakse at? asjagath@ntu.edu.sg, FUZZ-IEEE2008 Special Sessions Chair Xiao-Jun Zeng at x.zeng@manchester.ac.uk, CEC2008 Special Sessions Chair Yuhui Shi at shi@ieee.org, or Special Sessions Chair on Emerging Areas Byoung-Tak Zhang at btzhang@bi.snu.ac.kr by November 1, 2007. A special session proposal should include a proposed session title, a brief description of the scope and motivation, biographic and contact information of the organizer(s).

Call for Tutorials and Workshops

WCCI 2008 will also feature pre-congress tutorials and post-congress workshops covering fundamental and advanced computational intelligence topics. A tutorial proposal should include title, outline, expected enrollment, and presenter biography. Any inquires regarding the tutorials should be addressed to Tutorial Chairs Wlodzislaw Duch at wduch@is.umk.pl, Russell Eberhart at reberhar@iupui.edu, and Qiang

Shen at qqs@aber.ac.uk by January 1, 2008. Any inquires regarding the workshops should be addressed to Workshops Chairs Irwin K.C. King at king@cse.cuhk.edu.hk and Yangmin Li at ymli@umac.mo by January 1, 2008.

Call for Competitions

WCCI 2008 will host competitions to stimulate research in computational intelligence, promote fair evaluations, and attract students. The proposals should include descriptions of the problems addressed, motivations and expected impact on computational intelligence, data description, evaluation procedures and established baselines, schedules, anticipated number of participants, and a biography of the main team members. Proposals are invited to be sent to Competitions Chairs Isabelle Guyon at isabelle@clopinet.com for IJCNN2008, Leszek Rutkowski at rutko@kik.pcz.czest.pl for FUZZ-IEEE2008, or Philip Hingston at p.hingston@ecu.edu.au for CEC2008 by October 1, 2007.

Important Due Dates:

- Competition Proposal: October 1, 2007
- Special Session Proposal: November 1, 2007
- Paper Submission: December 1, 2007
- Tutorial/workshop Proposal: January 1, 2008
- Decision Notification: February 1, 2008
- Camera-Ready Submission: March 1, 2008

More information can be found at http://www.wcci2008.org

For general inquiries, please contact General Chair Jun Wang at jwang@mae.cuhk.edu.hk.

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 by 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.