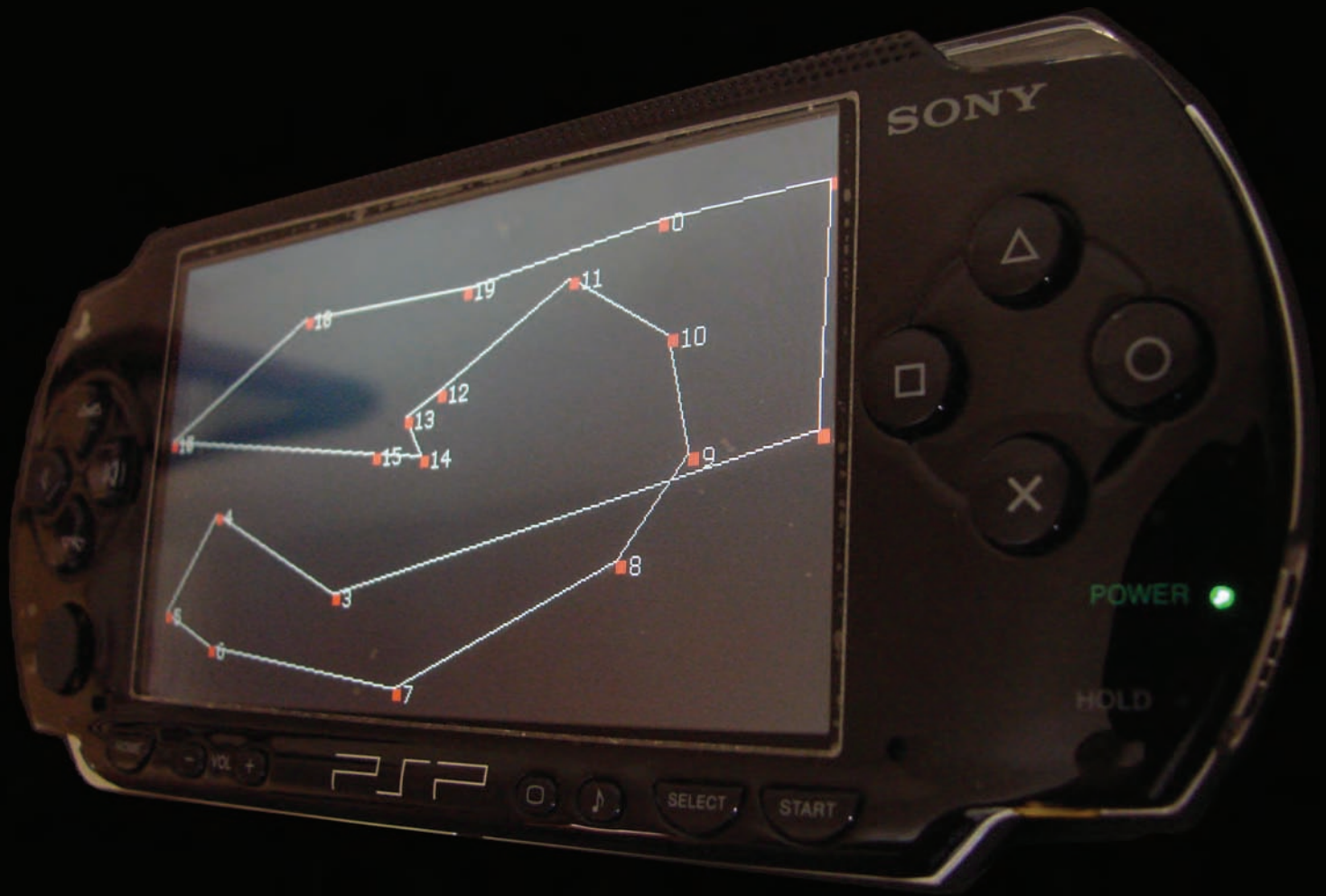
SIGEVOlution

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

Volume 4
Issue 1



in this issue

Editorial

Wow! In the last editorial, I said I was impressed by Stephanie Forrest for the three awards she received at GECCO-2009. Today, I am impressed even more. I was just told that her paper "Automatically Finding Patches Using Genetic Programming", co-authored with Westley Weimer, ThanhVu Nguyen, and Claire Le Goues, has been awarded with the IFIP TC2 Manfred Paul Award and the ACM SIGSOFT Distinguished Papers Award during 31st International Conference on Software Engineering (ICSE 2009) — the major ACM conference on software engineering.

Congratulations Stephanie! Two awards at a major conference outside our field are an exceptional way to bring evolutionary computation to the people.

Also the authors of the articles in this new issue work toward the same goal, in their own way. First, Arthur Kordon tells us about marketing computational intelligence with humor. The paper is just an appetizer of Arthur's new book "Applying Computational Intelligence: How to Create Value" to be published December 2009 by Springer. If you like the appetizer, there is more information about the main course here. In the second paper, Christian Perone presents Pyevolve, his open-source framework for genetic algorithms which can run on any platform supporting pure Python 2.5, such as the Sony PlayStation Portable and cellphones (the cover photo actually shows a genetic algorithm solving TSP on such a device!).

Could I do all this by myself? Absolutely not! In fact, I owe my thanks to Arthur Kordon, Chistian S. Perone, Martin V. Butz, Xavier Llorá, Kumara Sastry, Cristiana Bolchini, Mario Verdicchio, Viola Schiaffonati, and board members Dave Davis and Martin Pelikan.

Oops! I almost forgot this. The deadline for papers for GECCO-2010 is less than three months away. The deadline for submissions is January 13th 2010, so it is probably time to start those experiments! The call for papers is available here and if you wish to add a note to your calendar, just follow this link.

Pier Luca
November 4th, 2009

SIGEVOlution Volume 4, Issue 1

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

Contributors to this Issue

Arthur Kordon
Christian S. Perone

Contents

ISSN: 1931-8499

Computational Intelligence Marketing

Arthur Kordon, The Dow Chemical Company, akordon@dow.com

"A model is like a political cartoon. It picks-up the substantial part of a system and exaggerates it." — John H. Holland

Research marketing relies on capturing the attention of the targeted audience with inspiring messages and attractive presentations. Simple non-technical language, interesting stories, and informative visualization are a must in this process. Humor is another significant factor for success and is widely used in many forms.

Research Marketing Principles

Selling research ideas and methods is not a familiar activity for most of the academic and industrial research communities. The usual attitude is negative and the marketing efforts are treated by many as a "car-dealer-type of activity far below the high academic standards". Common practices for communicating research ideas and methods are via publications, patents, conference presentations, and proposals for funding. In this way the targeted audience is limited to the scientific peers in the corresponding research areas. The average size of this audience is a couple of hundred researchers and barely exceeds several thousand for the most popular specialized scientific communities. If this number is sufficient for spreading the message across the academic world, it is absolutely inadequate for selling the concept to the broad audience of potential users in the real world.

Different research methods require distinct marketing efforts. The well-known and widely used first-principles models or statistics do not need research marketing of their capabilities. However, new technologies, like computational intelligence, need significant and systematic marketing efforts. Of course, the objective is not to transform researchers into advertisement agents or car dealers, but to raise their awareness of the need for research marketing and demonstrate some of the necessary techniques.

Research Marketing Techniques

One of the key specifics of research marketing is that the predominant marketing efforts are of direct promotion of the product to the potential user. In this case, the quality of the presentation and the attractiveness of the delivery are critical. We'll focus on three techniques that are of high importance for marketing success – attractive message delivery, effective visualization, and, of course, humor.

Message Delivery

Sending a clear message and combining it with a touching story are critical in the promotion process. Successful marketers are just the providers of stories that consumers choose to believe. With the almost universal access to the Internet, there is a fundamental shift in the way ideas are spread. Either you're going to tell stories that spread, or you will become irrelevant. The key suggestions for effective message delivery, recommended by marketers, are given below:

Define attractive headlines – As we know, company and product icons and headlines are a substantial part of any form of advertisement. They try to capture the essence of the message and to succeed in the tough competition for customer's attention. The same idea is valid for research marketing, and some suggestions for computational intelligence methods are given in Fig. 1. Ideally, the headline must represent the substance of the method and the key source of value creation.

Tell a true story – In direct marketing, the best way to capture immediate attention is by telling interesting stories that stay in the audience memory longer than product features. The stories must be short, striking, and true. Here is an example from the author's experience.

In a presentation promoting symbolic regression-based inferential sensors, the author begins with a real story of customer dissatisfaction with the existing neural-network-based solutions, applied in a chemical plant. The plant engineer was asked how the neural network model predictions had been used recently. She looked ironically and asked: "Do you really want to know the truth?" She then walks over to the trashcan and retrieves from the garbage the last printout of model estimates. The predictions for a week were a constant negative number, which made no physical sense. The credibility of the model was at the level of the place it was "used" – a piece of junk in a trashcan. No more explanations were needed.

Deliver, and then promise – The most important part of the message is to define clear deliverables to the customer. In the case of the above-mentioned example, the advantages of the proposed alternative to the neural network fiasco — symbolic regression — were well emphasized. They were illustrated with delivered inferential sensors in different manufacturing plants, which were heavily used. Good practice is to give references from customers. Plant engineers and operators liked the simplicity of the derived solutions. One operator even advertised his appreciation from implemented symbolic-regression inferential sensors to his peers with the highest possible form of admiration in this straight-talk community: "very cool stuff".

Do not oversell – Avoiding the "snake oil" salesmen image is critical in direct marketing. Unfortunately, the relative ignorance of the generic audience towards the capabilities of computational intelligence gives opportunities for the "irrational exuberance" type of marketing. A typical example was the initial oversell of neural networks in the 1990s, often with disappointing results, like the one given above.

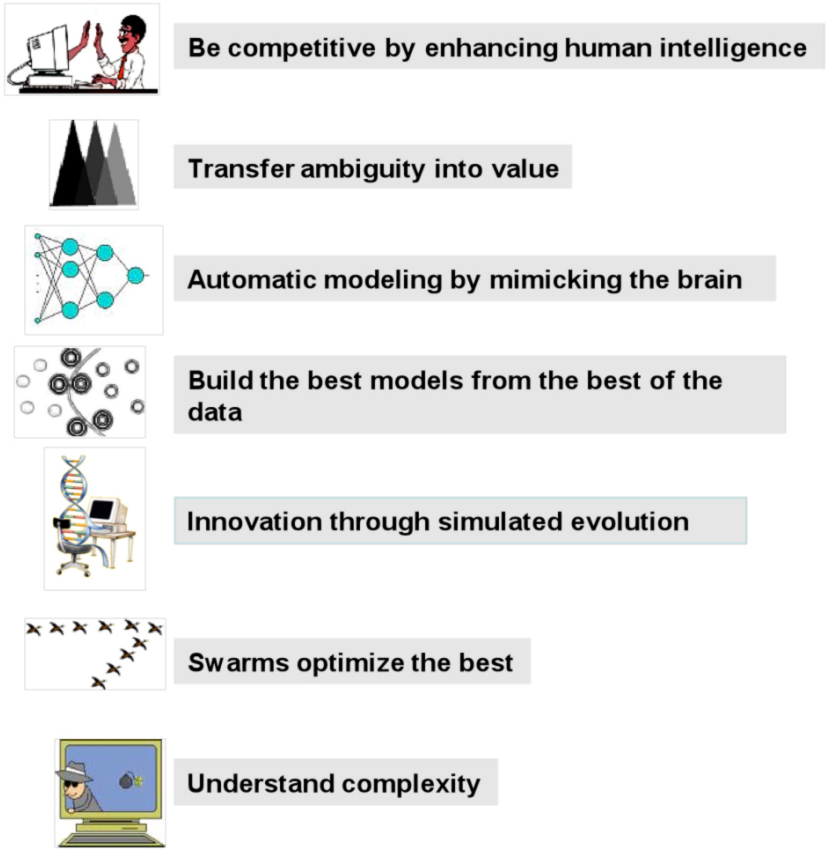


Fig. 1: Ideas for icons and headlines of the key computer intelligence approaches (fuzzy systems, neural networks, support vector machines, evolutionary computation, swarm intelligence, and agent-based systems).

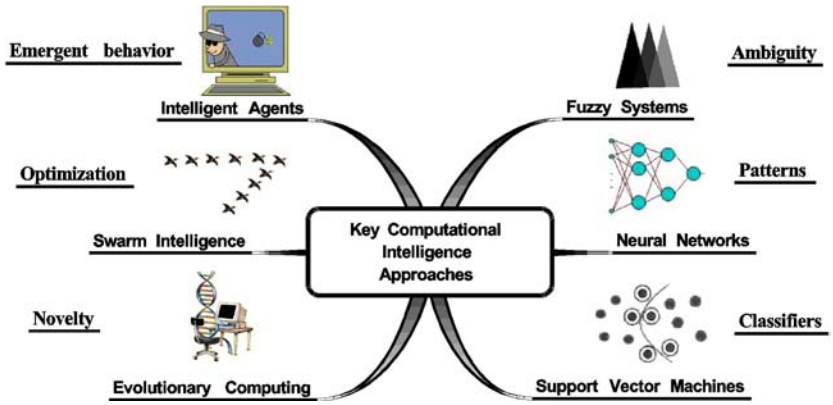


Fig. 2: An example of combining mind-maps and clip art.

The best strategy to avoid oversell is by defining clear deliverables and discussing openly the technology limitations. For the given symbolic regression inferential sensor example, the well-known issues with this type of model, such as dependence on data quality, unreliable predictions 20% outside the range of training data, and more complex maintenance and support, were identified. In addition, examples of how these issues had been addressed in the previous applications were given.

Effective Visualization

Images are critical for marketing. Without attractive visual representation, the defined product has low chances for sale. In the case of research marketing, finding a simple and memorable image is not trivial. Exploring successful ads, imagination and using different visualization techniques are needed. Some examples are given below.

Combining Mind-maps and Clip Art

One recommended visualization approach, which is broadly used in the book, is combining the power of mind-mapping with the attractiveness of clip art. An example is given in Fig. 2.

The figure clearly illustrates the advantage of exploring the synergy between both techniques. While mind-maps represent structures and links very effectively, they are not visually attractive and the chance that a mind-map will stick in our memory is very low. Let's not forget that, in the end, remembering the presentation is one of the main marketing objectives. Adding vivid clip-art icons and images brings "life" into the presentation and increases the chance of attracting potential customers' attention.

Some Visualization Techniques

Among the multitude of known visualization techniques, we'll focus on some suggestions, given by the famous visualization guru Edward Tufte. His definition of effective visualization is as follows: "Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space." Some of the key techniques he recommends to achieve this goal are:

- Show comparison
- Show causality
- Show multivariate data
- Integrate information
- Use small multiples
- Show information adjacent in space

An example of a presentation based on some of these techniques is shown in Fig. 3. The presentation represents the results of a comparative study between classic first-principles modeling and accelerated fundamental modeling, using symbolic regression. The competitive nature of the study is emphasized by horizontal parallel comparison in the similar steps of the approaches. Each step is additionally visualized either with icons or graphs. The key synergetic message is sharpened by a clear verbal equation and supported by the yin-yang icon, a well-known symbol of synergy. The final results from the competition – the striking differences in model development time – are directly represented by a calendar and a clock.

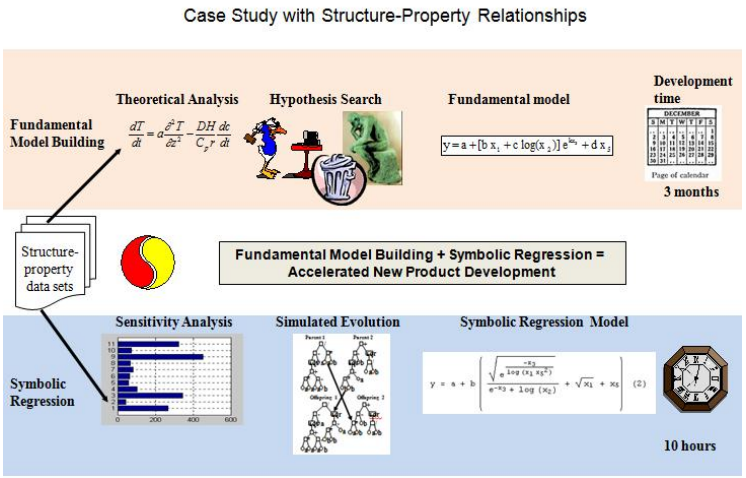


Fig. 3: An example of applying horizontal visual comparison between first-principles modeling and symbolic regression.

To PP or Not to PP?

One of the critical factors in research marketing is mastering PP (PowerPoint) presentations. According to Tufte, between 10 and 100 billion PP slides are produced yearly[1]. Unfortunately, the average information content of bullet-rich PP slides is extremely low, not to mention the sleeping pill effect.

There is a lot of criticism of the prevailing cognitive style of PP, which, rather than providing information, allows speakers to pretend that they are giving a real talk, and audiences to pretend that they are listening. This prevailing form of the bureaucratic-style of communication may lead to complete information abuse and must be avoided.

One alternative, recommended by Tufte, is preparing paper handouts of text, graphics, and images. The big advantage of handouts is that they leave permanent traces with the potential customers. One suggested paper size for presentation handouts is A3, folded in half to make four pages. According to Tufte, that one piece of paper can contain the content-equivalent of 50–250 typical PP slides!

[1] E. Tufte, Beautiful Evidence, Graphics Press, Cheshire, CT, 2006

From our experience, even using a double-sided A4 format paper is very effective for capturing the attention of the audience. Another more realistic alternative, having in mind the ubiquitous role of PP, is increasing the information content and attractiveness of the slides and reducing their number as much as possible. It is our experience that even the most complex topics can be condensed into fewer than 20–25 information-rich slides. The slide shown in Fig. 3 demonstrates the recommended design of such slides.

Humor

Successful marketing must be funny. The paradox is that very often the short funny part of a presentation that includes cartoons, jokes, quotations, and Murphy's laws stays longer in the memory than any technical information. That is why it is very important to include humor in the marketing process. An example of several cases with useful quotations and Murphy's laws, related to computational intelligence, are given below (examples with Dilbert cartoons are given in the book). They are a good starting point to inspire future presenters to develop their own humor libraries.

Useful Quotations

The caricatures in this section are from the Bulgarian cartoonist Stelian Sarev.

"*All models are wrong; some models are useful.*"

— George Box

"*Don't fall in love with a model.*"

— George Box

"*A silly theory means a silly graphic.*"

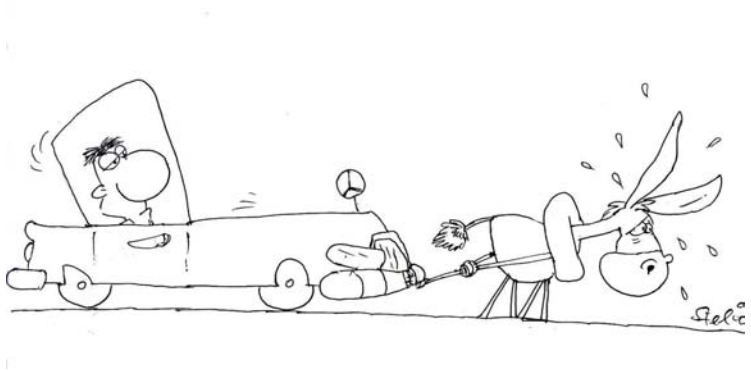
— Edward Tufte

"*The complexity of a model is not required to exceed the needs required of it.*"

— D. Silver

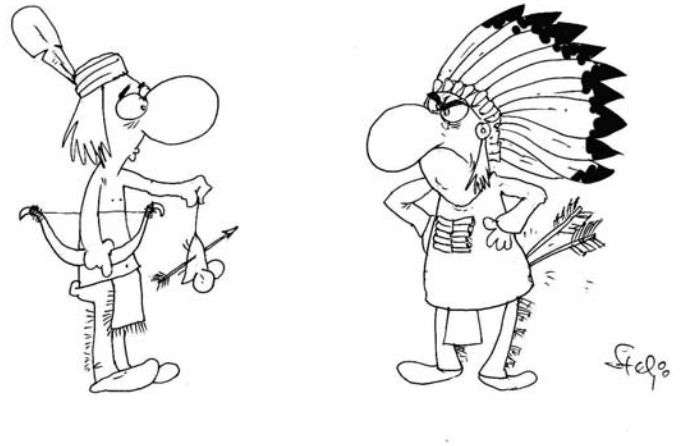
"Complex theories do not work, simple algorithms do."

— Vladimir Vapnik



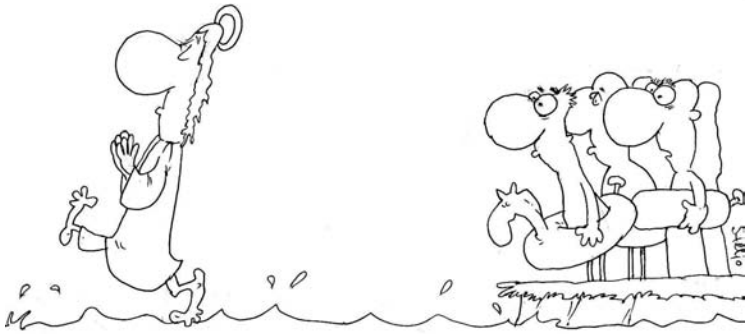
"It is better to solve the right problem approximately than to solve the wrong problem exactly."

— John Tukey



Murphy's Laws Related to Computational Intelligence

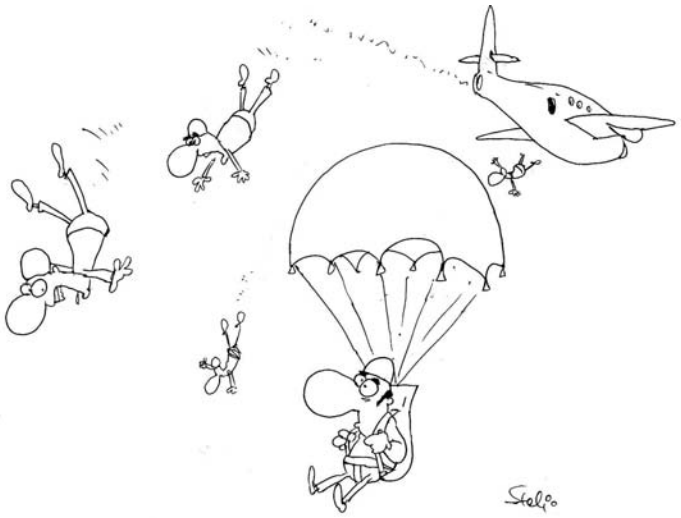
- **Murphy's law**
  If anything can go wrong, it will.
- **Box's law**
  When Murphy speaks, listen.
- **Law of ubiquitous role of data mining**
  In God we trust. All others bring data.



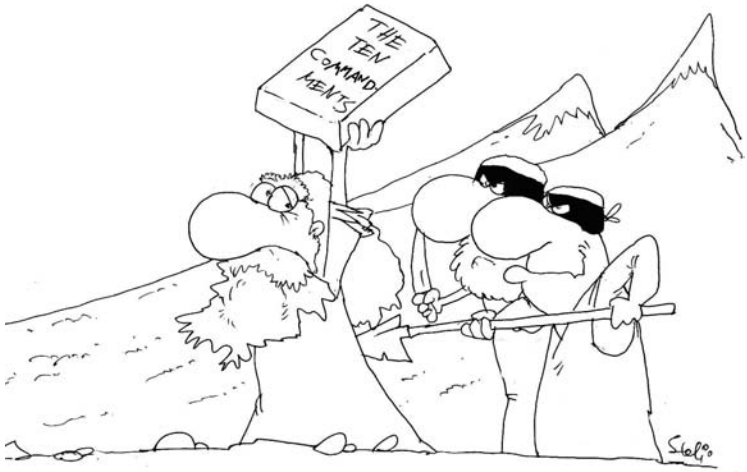
- **Murphy's law on machine learning**
  The number one cause of computer problems is computer solutions.
- **Menger's law on modeling success**
  If you torture data sufficiently, it will confess to almost anything.
- **Murphy's law effects on data**
  If you need it, it's not recommended.
  If it's recommended, it's not collected.
  If it's collected, it's missing.
  If it's available, it's junk.
- **Knuth's law on evil optimization**
  Premature optimization is the root of all evil.

■ **Kordon's law on computational intelligence**
Computers amplify human intelligence.
Positive Corollary: Smart guys get smarter.
Negative Corollary: Dumb guys get dumber.

■ **Murphy's law on rule-based systems**
Exceptions always outnumber rules.



■ **Murphy's law on sources of fuzzy logic**
Clearly stated instructions will consistently produce multiple interpretations.



■ **Murphy's law on evolutionary computation**
Simulated evolution maximizes the number of defects which survive a selection process.[2]

Marketing Computational Intelligence to a Technical Audience

The success of research marketing depends on two key audiences — technical and business. The technical audience includes the potential developers of computational intelligence-based applications. Usually this audience has an engineering and mathematical background and is interested in the technology principles and functionality. The business audience includes the potential users of the applied systems, who are more interested in the value creation capability of the technology and how easily it can be integrated into the existing work processes.

[2] R. Brady, R. Anderson, R. Ball, Murphy's Law, the Fitness of Evolving Species, and the Limits of Software Reliability, University of Cambridge, Computer Laboratory, Technical Report 471, 1999.

The marketing approach for the technical audience is discussed in this section. The interested reader can find the description of the marketing approach for the nontechnical audience in the corresponding chapter in the book. We'll concentrate on two topics: (1) how to prepare an effective presentation for an audience with a technical background; and (2) how to approach technical gurus.

Guidelines for Preparing Technical Presentations for Applied Computational Intelligence

The main objective of a good technical presentation is to demonstrate the technical competitive advantage of the proposed approaches. The key expectations from a technical introduction of a new technology are: well-explained main principles and features, defined competitive advantage, potential application areas within users' interest, and assessment of implementation efforts. One of the challenges in the case of computational intelligence is that the technology is virtually unknown to the technical community at large. The assumption is that computational intelligence must be introduced from scratch with minimal technical details. In order to satisfy the different levels of knowledge in the audience, a second, more detailed presentation can be offered off-line to those who are interested. One of the key topics in the presentation is the differences between the proposed methods and the most popular approaches, especially those used frequently by the targeted audience, such as first-principles models and statistics. The audience must be convinced about the technical competitive advantage of a given approach relative to another technology with arguments based on their scientific principles, simulation examples, and application capabilities. A realistic assessment with a balance between the strengths and weaknesses is strongly recommended. An example of an introductory slide for neural networks is shown in Fig. 4.

Two central topics of neural networks are addressed: (1) the capability of a neural network to generate nonlinear models; and (2) the danger that these solutions are inefficient due to local optimization. The first topic is demonstrated by showing how nonlinearity is captured by the hidden layer of a three-layer neural network. The second topic, generating nonoptimal models due to the potential of the learning algorithm to be entrapped in local minima, is represented visually in the left part of the slide. The trashcan icon can be used during real presentations as a reminder to tell the real story of lost credibility due to inefficient design of neural networks (the trashcan story previously discussed).

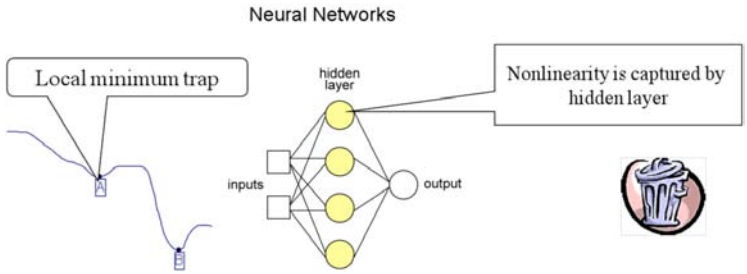


Fig. 4: An example of introducing neural networks to a technical audience.

Another, more complicated slide for representing the key technical features of genetic programming (GP) is shown in Fig. 5.

This is a typical slide for introducing a new method to technical audience. We call it the "kitchen slide" since it includes information about the technology kitchen in generating the final solution. The slide template is organized in the following way. The title section includes a condensed message that represents the essence of the method, supported by an appropriate visualization. In the case of GP, the condensed message is "We'll turn your data into interpretable equations!" visualized by the icons showing the transformation of the data into equations by GP (in this case, the standard evolutionary computation icon is used).

The slide is divided into three sections: (1) method description; (2) advantages; and (3) application areas. The method description section presents a very high-view visual description of the approach. In the case of GP, it includes the images of competing mathematical functions, represented by trees and the image of a Pareto front figure with an equation on the zone of optimal accuracy and complexity. The method description section sends two clear messages to the audience: (1) at the basis of GP is the simulated evolution of competing mathematical functions, and (2) the selected solutions are robust based on the optimal trade-off between accuracy of predictions and model complexity.

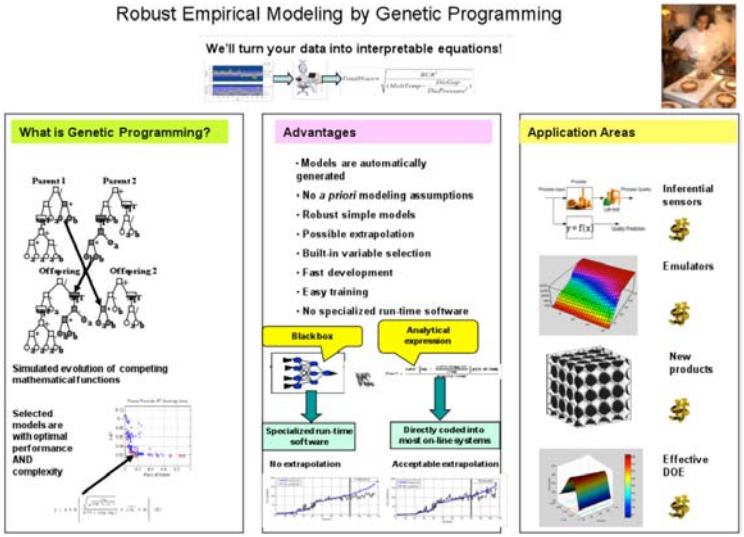


Fig. 5: An example of introducing genetic programming to technical audience.

The advantages section summarizes with text and images the key benefits of the approach. In the case of the GP example in Fig. 5, a visual comparison with neural networks is given. It demonstrates some of the key advantages of symbolic regression, generated by GP – analytical expressions, directly coded into the user's system with acceptable performance in extrapolation mode.

The application area section focuses on the key implementation successes of the method. It is strongly recommended to give specific numbers of the created value, if possible.

The proposed template can be used for any technology. It can also be given as a printed handout before the presentation.

Key Target Audience for Technical Presentations

A typical technical audience includes industrial researchers, technical subject matter experts, practitioners, system integrators, and software developers.

Their incentives to break the cozy status quo and introduce a new technology depend on the benefits they'll get as a result of increased productivity, ease of use, and rewarding their risk-taking with potential career development and bonuses. In principle, it is very difficult to expect that introducing computational intelligence will satisfy all of these incentives. Thus, management support is required to start the efforts. The critical factor in gaining the needed support is the opinion of the technical leaders (gurus) in the organization or the research community. It is practically impossible to open the door for the new technology without their blessing. That is why in the rest of the section we'll focus our attention on approaching effectively this key category of the technical audience.

In order to specify the marketing efforts as much as possible, we divide technical gurus into six categories based on their support of a new idea. The bell curve of the different types is shown in Fig. 6, where the level of support increases to the right and the level of rejection of the new idea increases to the left.

Two types of gurus – The Visionary and The Open Mind – can fully support the new technology. On the opposite side are The 1D Mind and The Retiring Scientist gurus, who will try by all means to kill or postpone the introduction of the new idea. Most technical gurus are neutral to the new technology and their support will depend on increasing their personal technical influence (in the case of The Technical King) or gaining political benefits (in the case of The Political Scientist). The typical behavior of the different gurus and the recommended approaches for presenting the new idea to them are given below.

Visionary Guru

Typical behavior: Having the blessing of a Visionary Guru is the best-case scenario of full enthusiastic support of the new idea. The Visionary Guru is the driving force of innovation in an organization. She/he is technically sharp and has an abundance of ideas, i.e. shares her/his own ideas and enjoys exploring new ones. Other important characteristics of the Visionary Guru are: outside focus with respect to external expertise, risk taking, and political skills to convince management. The Visionary Guru is well informed and her/his office is full with bookshelves of technical journals and books from diverse scientific areas.

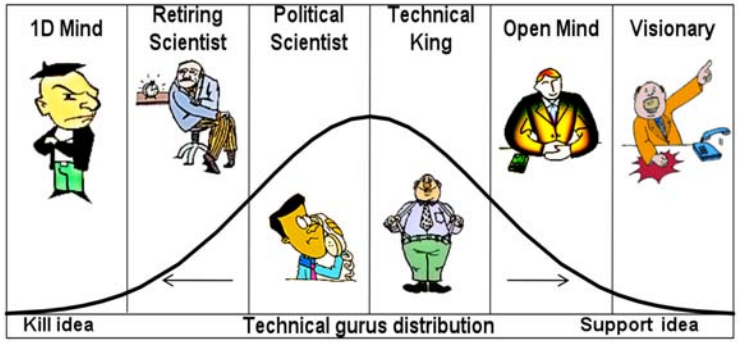


Fig. 6: Technical guru distribution based on their support of a new idea.

Recommended approach: Prepare an inspirational talk with a very solid technical presentation. Be ready to answer detailed technical questions and to demonstrate competence about the scientific basis of the idea. Convincing the Visionary Guru is not easy, but if successful, the chance of a positive decision to apply the proposed approach is almost 100%.

Open Mind Guru

Typical behavior: The Open Mind Guru accepts new ideas but will not initiate change without serious technical arguments. She/he is more willing to support methods that require gradual changes or have an established application record. The risk-taking level and enthusiasm is moderate. Before supporting the new idea, the Open Mind Guru will carefully seek the opinion of key technical experts and especially managers. Her/his bookshelf is half the size of the Visionary Guru.

Recommended approach: Focus on detailed technical analysis that demonstrates competitive advantages over the methods closer to the Open Mind Guru's personal experience. Show specific application examples or impressive simulations in areas similar to the targeted business. It is also recommended to discuss the synergetic options between the new method and the most used approaches in the targeted business. An important factor for gaining the Open Mind Guru's support is if the new method is implemented by an established software vendor.

Technical King Guru

Typical behavior: The Technical King Guru dominates an organization with her/his own ideas, projects, and cronies. The key factor in her/his behavior is gaining power, i.e. the best way to have the blessing for a new idea is if it will increase her/his technical influence. In this case, the Technical King Guru will fully support the idea and will use all of her/his influence to apply it. Otherwise, the chances for success depend only on top management push. The Technical King Guru shines in her/his area of expertise and has a good internal network and business support. On her/his bookshelf one can see only books related to her/his favorite research topics.

Recommended approach: Understand key areas of expertise of the Technical King Guru and prepare a presentation that links the proposed new approach with the identified areas. Recognize the importance of the Technical King and her/his contribution and try to demonstrate how the new idea will fit in with her/his technical kingdom and will increase her/his glory and power. Independently, top management support could be pursued to counteract eventual idea rejection.

Political Scientist Guru

Typical behavior: The Political Scientist Guru is on the negative side of "new idea support" distribution, shown in Fig. 6. By default, she/he rejects new approaches, since they increase the risk of potential technical failure with corresponding negative administrative consequences. Technically, the Political Scientist Guru is not in the list of "the best and the brightest" and this is another reason for looking suspiciously at any new research idea. The real power of the Political Scientist is in using effectively political means to achieve technical objectives. From that perspective, new idea support depends on purely political factors, such as top management opinion, current corporate initiatives, and the balance of interests between the different parts of the organization related to the new technology. On her/his bookshelves one can see a blend of books on technical and social sciences with favorite author Machiavelli.

Recommended approach: The marketing efforts must include a broader audience than technical experts. It is critical to have a separate presentation to top management first, which emphasizes the competitive advantages of the proposed approach. The ideal argument will be if some of the competitors are using the new methods. The presentations must have minimal technical details, be very visual, and application-oriented.

Retiring Scientist Guru

Typical behavior: The Retiring Scientist Guru is counting the remaining time to retirement and trying to operate in safe mode with maximum political loyalty and minimum technical effort. In order to mimic activity, she/he uses the sophisticated rejection technique, known as "killing the new idea by embracing it". The Retired Scientist is a master of freezing the time by combining bureaucratic gridlock with departmental feuds. As a result, the new idea is buried in an infinite interdepartmental decision-making loop. Often the possible solution is in her/his potential career plans as a consultant after retirement. Don't expect bookshelves in her/his office. Only outdated technical manuals occupy the desk.

Recommended approach: Broaden marketing efforts to several departments. Sell the new idea directly to top management first. Be careful to avoid the "embrace and kill" strategy. Try to understand if the new technology may fit in with Retirement Scientist's consultant plans after leaving office.

1D Mind Guru

Typical behavior: The 1D Mind Guru is the worst-case scenario with almost guaranteed new idea rejection. She/he has built her/his career on one approach only, which has created value. This mode of ideas deficiency creates fear of novelty and aggressive job protection. Any new idea is treated as a threat that must be eliminated. The 1D Mind Guru is well informed on the weaknesses of any new approach and actively spreads negative information, especially to top management. Bookshelves are also absent in 1D Mind Guru's office. However, the walls are often filled with certificates and company awards.

Recommended approach: Don't waste your time. Try other options.

About the author

Arthur Kordon is a Data Mining & Modeling Leader in the Data Mining & Modeling Group, The Dow Chemical Company in Freeport, Texas, USA. He is an internationally recognized expert in applying computational intelligence technologies in industry. Dr. Kordon has successfully introduced several novel technologies for improved manufacturing and new product design, such as robust inferential sensors, automated operating discipline, accelerated fundamental model building, etc. His research interests include application issues of computational intelligence, robust empirical modeling, intelligent process monitoring and control, and data mining. He has published more than 60 papers, one book and nine book chapters in the area of applied computational intelligence and advanced control. Dr. Kordon is a member of the Technical Committee on Evolutionary Computation of IEEE Computational Intelligence Society. Dr. Kordon holds a Master of Science degree in Electrical Engineering from the Technical University of Varna, Bulgaria in 1974 and a Ph.D. degree in Electrical Engineering from the Technical University of Sofia, Bulgaria in 1990.

Company webpage: www.dow.com
Email: akordon@dow.com

Pyevolve: a Python Open-Source Framework for Genetic Algorithms

Christian S. Perone, christian.perone@gmail.com

Pyevolve is an open-source framework for genetic algorithms. The initial long-term goal of the project was to create a complete and multi-platform framework for genetic algorithms in pure Python. However, the most recent developmental versions currently support also Genetic Programming (GP)[3]; accordingly, Pyevolve now aims at becoming a pure Python framework for evolutionary algorithms.

The project started back in 2007; the current stable version is Pyevolve 0.5, which was released on January 22nd, 2009 and publically available since then from the project site (http://pyevolve.sourceforge.net). The features presented in this article are from the last development version, which will be included in the next official distribution Pyevolve 0.6 — available in the near future.

Pyevolve is organized in packages to keep the API simple and intuitive. Figure 1 shows how the different packages are organized while Figure 2 shows how the components work together. Pyevolve offers several interesting features.

- **It is multiplatform** – the framework was written in pure Python, so it can run on Mac, Windows, Linux platforms, and on any portable devices where Python is available (e.g., the Sony PlayStation Portable or Symbian OS based cellphones);
- **It is extensible** – the user can easily add new chromosome structures, genetic operators, database or visualization adapters, etc.
- **It is loaded with standard features** – Pyevolve provides implementations of the most commonly used genetic operators, scaling schemes, selection methods, etc.

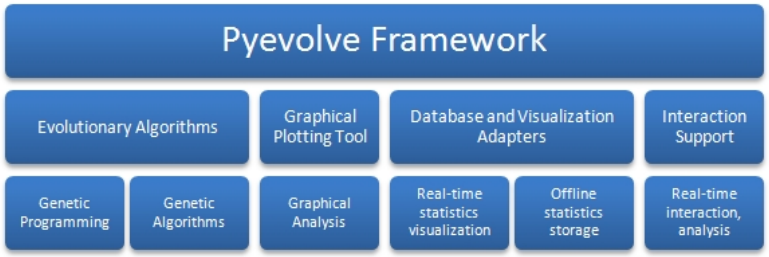


Fig. 1: Structure of the Pyevolve framework

- **It heavily uses default parameters** – to increase users' productivity, there are default values for almost all parameters;
- **It has easy-to-use and well documented APIs** – the API is very easy to learn and was developed to be simple and intuitive. Every component of the framework is documented and public available with examples[1];
- **It supports multiprocessing** – the framework supports the use of multiple cores, it will alternate the CPU cores of the machine to evaluate the fitness function for each individual in population;
- **It is open-source!** – the framework is entirely open-source and it is licensed upon a very permissive PSF[2]-like license.

[1] At the URL: http://pyevolve.sourceforge.net, the project home site
[2] The same used by Python Language

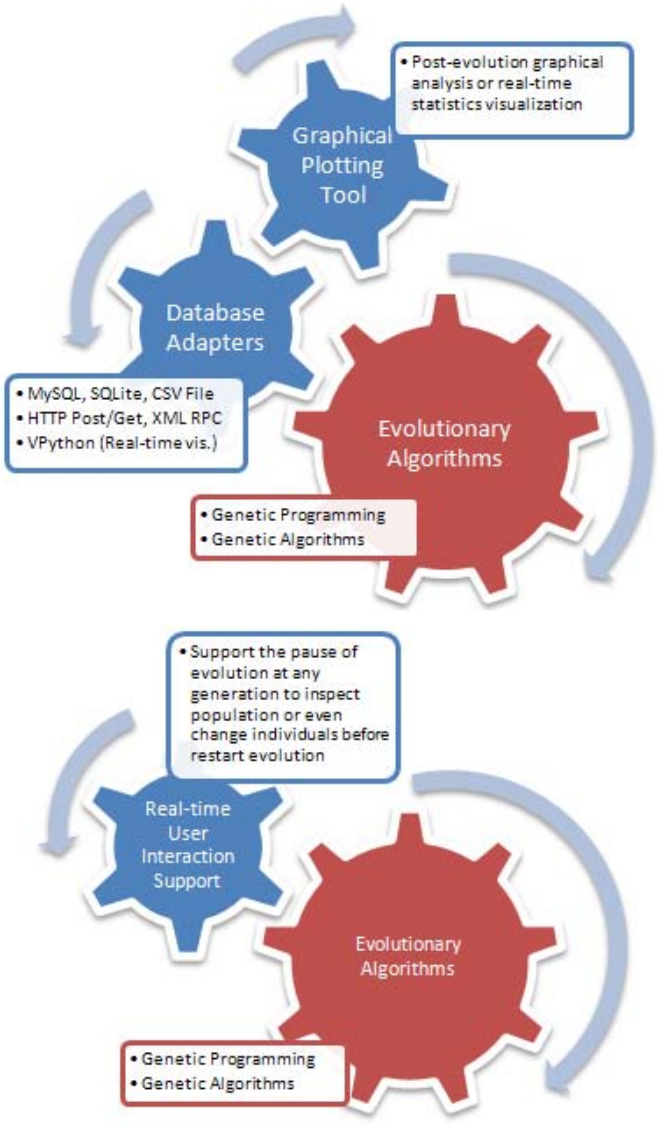


Fig. 2: Overview of the framework components

Genetic Algorithms in Pyevolve

The genetic algorithm core implemented in Pyevolve is based on the algorithm described in [1].

Chromosome Representations

The framework includes the classical representations used in genetic algorithms (**1D Binary String** and **2D Binary String**) and several other representations (**1D List**, **2D List**, and **Trees**).

What defines the data type of the representation is the genetic algorithm initialization function. Pyevolve provides ready-to-use built-in initialization routines for integer numbers, real numbers and for user-defined alleles. Besides the few representations available in the distribution, the flexibility of Python allows the creation of new representations by extending the existing ones.

Initialization

- for **1D Binary String** chromosomes: binary string initialization;
- for **1D List** chromosomes: integer initialization, real initialization, user-defined allele initialization;
- for **2D List** chromosomes: integer initialization, real initialization, user-defined allele initialization;
- for **Tree** chromosomes: integer initialization, user-defined allele initialization;

Reproduction/Crossover Operators

- for **1D Binary String** chromosomes: single-point crossover, two-point crossover, uniform crossover;
- for **1D List** chromosomes: single-point crossover, two-point crossover, uniform crossover, order crossover;
- for **2D List** chromosomes: single vertical point crossover, single horizontal point crossover, uniform crossover;
- for **Tree** chromosomes: sub-tree crossover, strict sub-tree crossover.

Mutation Operators

- for **1D Binary String** chromosomes: swap mutation, flip mutation;
- for **1D List** chromosomes: swap mutation, integer-range mutation, real-range mutation, integer-Gaussian mutation, real-Gaussian mutation, integer-binary mutation, user-defined allele mutation;
- for **2D List** chromosomes: swap mutation, integer-Gaussian mutation, real-Gaussian mutation, user-defined allele mutation;
- for **Tree** chromosomes: swap mutation, integer-range mutation, real-range mutation, Gaussian-integer mutation, Gaussian-real mutation.

Scaling methods

Pyevolve provides several scaling schemes: Linear Scaling, Sigma Truncation Scaling, Power Law Scaling, Boltzmann Scaling, and Raw Scaling.

Selection methods

It also includes several selection methods: Rank Selection, Uniform Selection, Tournament Selection, and Roulette Wheel Selection.

Interactive Mode

The interactive mode allows the user to pause evolution and interactively work on the population. The user can collect statistics, analyze individuals or even modify individuals and then restart the evolutionary process.

When the user enters the interactive mode, the framework gives access (i) to the evolutionary engine, (ii) to the population and (iii) to several numerical/graphical analysis methods like for instance the population fitness histograms (see Figure 3).

An example of how the Interactive Mode works is reported below (part of the output was omitted for the sake of brevity):

```
# python pyevolve_ex7_rastrigin.py
Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw)  [(...)]
Gen. 20 (3.33%): Max/Min/Avg Fitness(Raw) [(...)]
Loading modules for Interactive Mode...  done !
## Pyevolve v.0.6 - Interactive Mode ##
Press CTRL-Z to quit interactive mode.

>>> population[0].fitness
189.50864407423373

>>> population[0].mutator
Slot [Mutator] (Count: 1)
            Name: G1DListMutatorRealGaussian - Weight: 0.50
            Doc:  The mutator of G1DList, Gaussian Mutator

   Accepts the rangemin and rangemax genome params (...)

>>> ga_engine.getCurrentGeneration()
30

>>> population.getStatistics()
- Statistics
        Minimum raw score                       = 155.42
        Minimum fitness                         = 189.51
        Total fitness                           = 26292.87
        Standard deviation of raw scores        = 36.42
        (...)
```

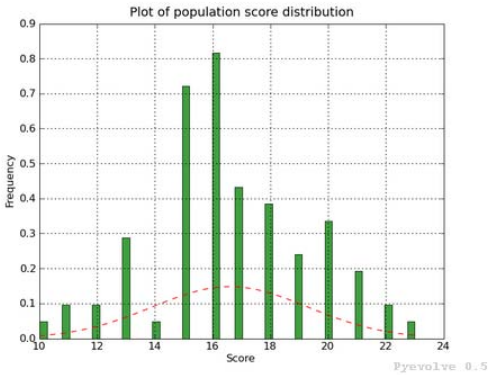


Fig. 3: Histogram for population fitness, created using the Interactive Mode

Database and Visualization Adapters

Pyevolve has several database and visualization adapters to save or load statistics and data collected during the evolutionary process:

- CSV file format adapter
- SQLite3 database adapter
- MySQL database adapter
- HTTP Get/Post adapter
- XML RPC adapter
- VPython visualization adapter

The SQLite3 database format is also used by the plotting tool included in Pyevolve. The user can get real-time statistics using HTTP Get/Post or XML RPC adapters, or visualize the real-time statistics using the VPython adapter (see Figure 4).

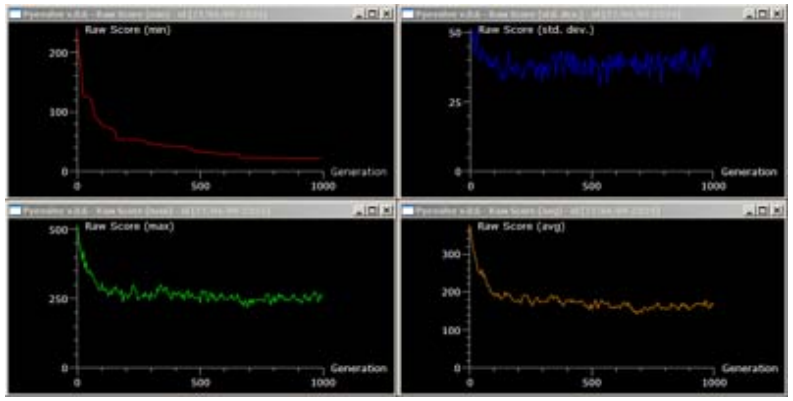


Fig. 4: The VPython real-time visualization adapter

Visualization

Pyevolve comes with a graphical plotting tool that can be used to visualize several statistics collected during the evolutionary process. As an example, Figure 5 shows a plot for the propagation of most adapted individuals: the x-axis represents each individual in the population; the y-axis reports the generation. Other examples of the available visualizations are shown in Figure 6 and Figure 7.

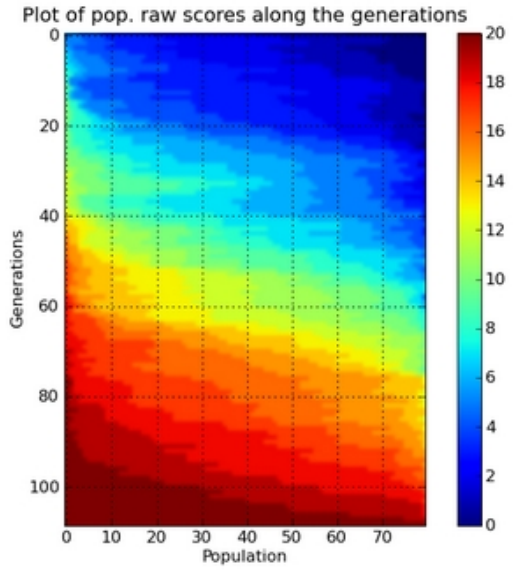


Fig. 5: A heat plot for raw fitness using Gaussian interpolation

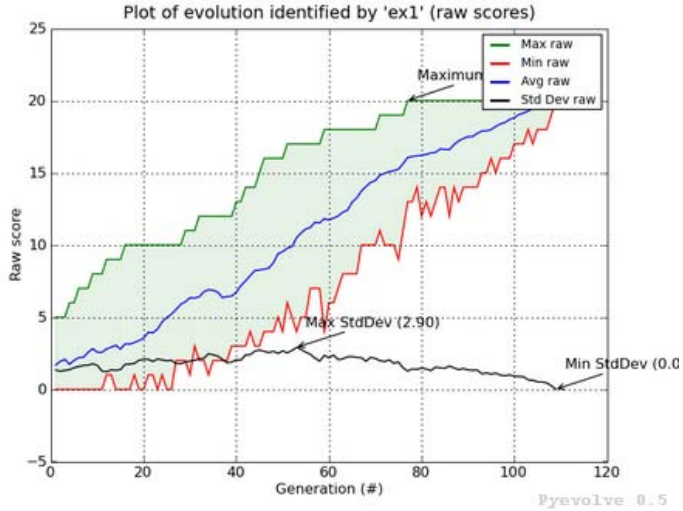


Fig. 6: Plot of the raw fitness during evolution

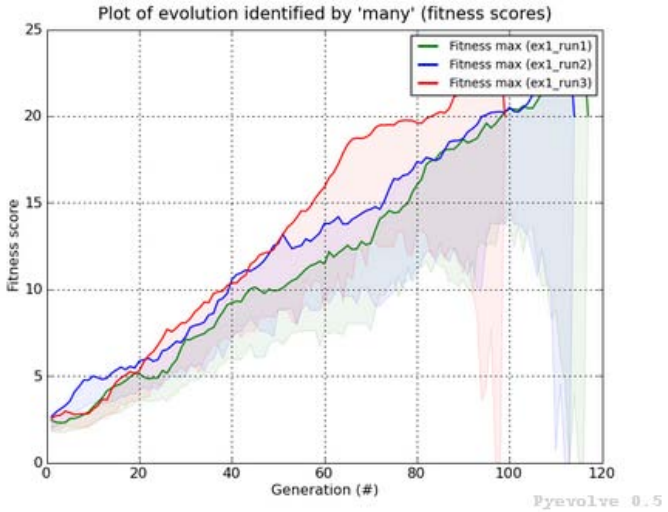


Fig. 7: Comparison of population fitness during different runs

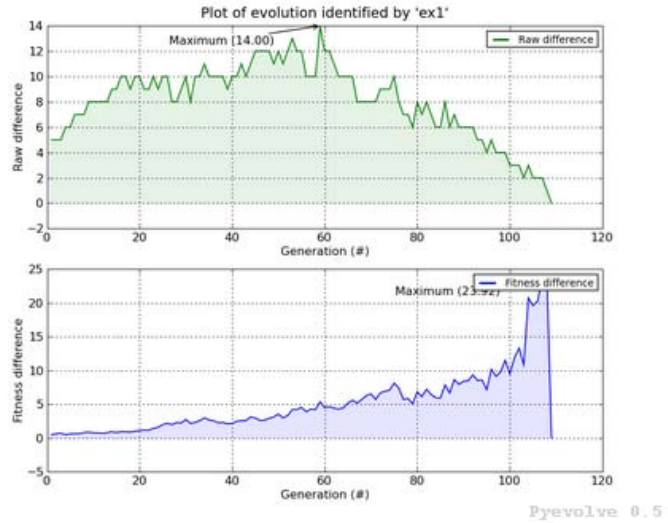


Fig. 8: A max and min raw/scaled fitness difference over the generations

Coding Example

As an example, we report the code for a genetic algorithm solving the minimization of the Rastrigin [6] function f(x) with 20 variables,

$$f(x) = 10n + \sum_{i=1}^{20} x_i^2 - 10\cos(2\pi x_i) \quad (1)$$

where n is the number of variables and -5.12 $\leq x_i \leq$ 5.12.

```
import math
from pyevolve import *

def rastrigin(genome):
   fitness = 0.0
   for gene in genome:
      fitness += gene**2 - 10*math.cos(2*math.pi*gene)
   return (10*len(genome)) + fitness

genome = G1DList.G1DList(20)
genome.setParams(rangemin=-5.12, rangemax=5.13)
genome.initializator.set(
   Initializators.G1DListInitializatorReal)
genome.mutator.set(Mutators.G1DListMutatorRealGaussian)
genome.evaluator += rastrigin

ga = GSimpleGA.GSimpleGA(genome)
ga.setMinimax(Consts.minimaxType["minimize"])
ga.setGenerations(800)
ga.setMutationRate(0.06)

ga.evolve(freq_stats=40)
print ga.bestIndividual()
```

.NET and Java Interoperability

Pyevolve is compatible with Jython[3] and with IronPython[4] interpreters. Jython is a Python interpreter written in Java, which means that all the applications running in Jython can be executed on any Java platform. Accordingly, users can extend and use Pyevolve on any Java platform.

[3] Tested with the Jython 2.5b1
[4] Tested with the IronPython 2.0.1

IronPython is a Python interpreter for the .NET platform that allows the interoperability between Pyevolve and .NET when the framework is run on IronPython. Note that there are some performance issues when exploiting interoperability and that there are also compatibility issues related to some Python modules that are missing from Jython and IronPython (e.g., SQLite3 used in Pyevolve to store and plot statistical data). However, the core evolutionary engine of Pyevolve runs properly on both Jython and IronPython.

What Does "multi" in Multi-platform Stand For?

Pyevolve is written in pure Python and it can run on *any* platform supporting the Python 2.5 core. Thus, we can run it on the usual operating systems (Linux, Windows and Mac OS X), but also on the many portable devices which support the Python 2.5 core (e.g., the Sony PlayStation Portable (PSP) and the Symbian OS based cellphones).

Pyevolve on Sony PSP

The PlayStation Portable (PSP) is a handheld game console manufactured and marketed by Sony Computer Entertainment. The PSP has a 333MHz clock, 32MB RAM, and a GPU with 2 MB onboard VRAM running at 166 MHz. Using the Stackless Python port for PSP[5] and the basic drawing API, the Traveling Salesman Problem (TSP) was ported to the PSP (see Figure 9); it took 42 seconds to find an optimal solution for a problem with 20 cities, using a population of 200 individuals using order crossover with a 100% probability and swap mutation with a 10% probability.

Pyevolve on Nokia N73

The Nokia N73 cellphone uses a Dual ARM 9 220 MHz CPU and has Symbian OS 9.1, S60 3rd edition. Nokia has developed the PyS60[6], which is the open-source Python language port to the Symbian S60 smartphone platforms. PyS60 has recently included the Python 2.5.4 core, so we can run Pyevolve using PyS60 as it is — no change in the framework is needed.

[5] http://code.google.com/p/pspstacklesspython/

[6] Python for S60, available at http://wiki.opensource.nokia.com/projects/PyS60

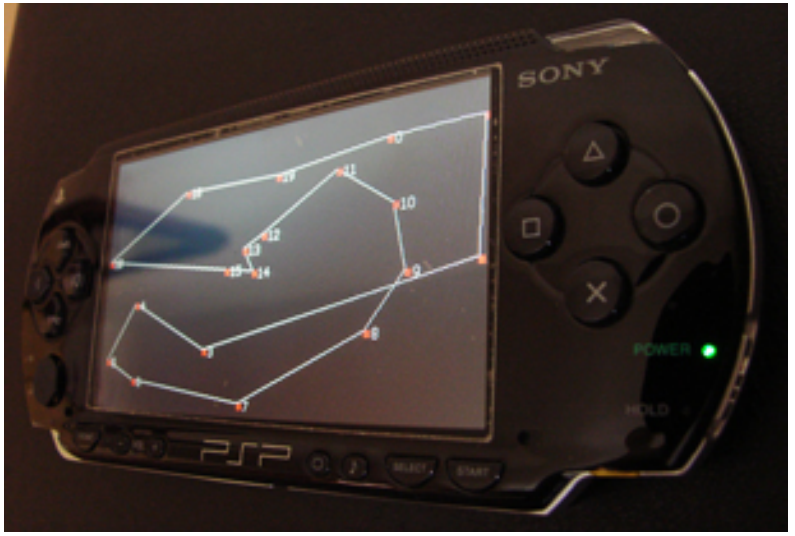


Fig. 9: Traveling Salesman Problem running on the Sony PlayStation Portable

The framework was tested on the Nokia N73 cellphone. The TSP was implemented for PyS60 using Pyevolve in the same manner as on the Sony PSP, but using the Canvas API of PyS60 (see Figure 10); it took 129 seconds to find an optimal solution for a problem with 20 cities, using a population of 200 individuals with a crossover rate of 100% and a mutation rate of 10%. The order crossover and the Swap mutation operation were used in this example.

Genetic Programming

Pyevolve tries to make Genetic Programming as simple and intuitive as possible, resulting in a good learning curve for people approaching this paradigm for the first time. For this purpose, Pyevolve uses simple solutions to reduce the programming burden:

- **Function set definition**: Pyevolve does not require the creation of full classes for defining very simple functions. Each function needs just one line of code and it's very intuitive (as shown in the later example);

Fig. 10: Traveling Salesman Problem running on the Nokia N73

- **Automatic function set detection**: the framework automatically detects the function set just by defining the function name prefix;
- **Automatic function arguments detection**: the framework automatically detects the number of arguments of the functions;
- **Compilation of the tree into bytecode**: instead of simulating the execution of the program tree for each fitness evaluation, Pyevolve compiles individuals into Python bytecode and runs it;

Using the dynamic typing of Python, users can specify highly flexible GP functions; for example, it is possible to write a function “mul(*)”, to multiply variables of different types, e.g., to multiply a list with an integer or an integer with another integer using the same function.

As an example, we report below a simple example for the symbolic regression of

$$f(a,b) = \sqrt{a^2+b^2}$$

```
import math
from pyevolve import *

error_accum = Util.ErrorAccumulator()

def gp_add(a, b): return a+b
def gp_sub(a, b): return a-b
def gp_mul(a, b): return a*b
def gp_sqrt(a):   return math.sqrt(abs(a))

def eval_func(chromosome):
   global error_accum
   error_accum.reset()
   code_comp = chromosome.getCompiledCode()

   for a in xrange(0, 5):
      for b in xrange(0, 5):
         evaluated     = eval(code_comp)
         target        = math.sqrt((a*a)+(b*b))
         error_accum   += (target, evaluated)

   return error_accum.getRMSE()

if __name__ == "__main__":
   genome = GTree.GTreeGP()
   genome.setParams(max_depth=4, method="ramped")
   genome.evaluator += eval_func

   ga = GSimpleGA.GSimpleGA(genome)
   ga.setParams(gp_terminals       = ['a', 'b'],
                gp_function_prefix = "gp")

   ga.setMinimax(Consts.minimaxType["minimize"])
   ga.setPopulationSize(500)
   ga.evolve(freq_stats=10)
   print ga.bestIndividual()
```

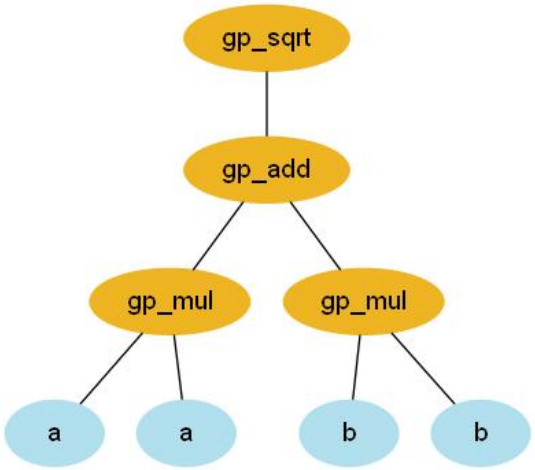


Fig. 11: The best individual of the population.

Visualization of Trees

With a simple function call, it is possible to visualize the individuals in the population:

```
best = ga.bestIndividual()
best.writeDotImage("best_individual.jpg")
```

This code takes the best individual of the population and creates the visualization in the Figure 11. The user can also visualize more individuals on the same image (see Figure 12).

Using Python decorators on the functions of the function set, the user can change the graphical representation of the function. Here is an example:

```
@GTree.gpdec(representation="+", color="red")
def gp_add(a, b): return a+b
```

In the example, the node text of the function “gp_add” is changed to “+” and the color of the node is set to red. The output of this change is shown in Figure 13.

Current Development

The current developmental version of Pyevolve implements additional choromosome types, genetic operators, and selection methods. It also supports easy-to-use distributed evolution over LAN/WAN. Finally, some work has also been undertaken to improve the performance of the genetic operators of the GA and GP.

Acknowlegments

I wish to thank some of the open-source projects that have made Pyevolve possible: Matplotlib[7] (used in the graphical plots), VPython[8] (used in the real-time visualization adapter) and Pydot[9] (used in the creation of Dot graph files for GP trees).

Bibliography

[1] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.

[2] Python Software Foundation. Python success stories; also available in http://www.python.org/about/success/, 2009.

[3] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: The MIT Press, 1992.

[4] William B. Langdon Riccardo Poli and Nicholas Freitag McPhee. *A field guide to genetic programming*. Published via lulu.com and freely available at www.gp-field-guide.org.uk, with contributions by J. R. Koza, 2008.

[5] Daniel G. Shafer. Python streamlines space shuttle mission design; also available in http://www.python.org/about/success/usa/, 2003.

[6] D. Whitley, K. Mathias, S. Rana, and J. Dzubera. Building better test functions. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 239–246. Morgan Kaufmann, 1995.

[7] http://matplotlib.sourceforge.net/
[8] http://vpython.org/
[9] http://code.google.com/p/pydot/

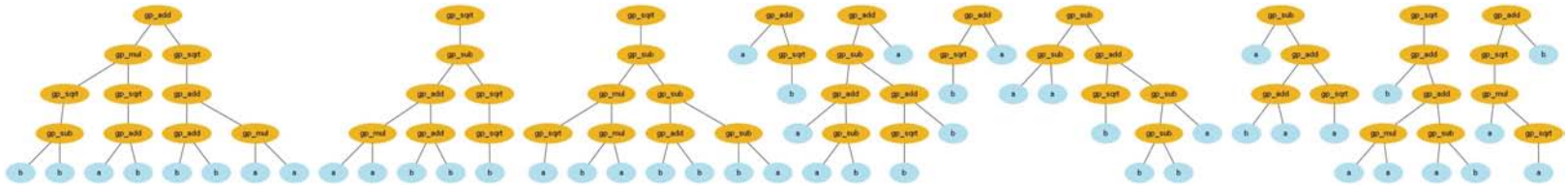


Fig. 12: The best ten individuals in the population.

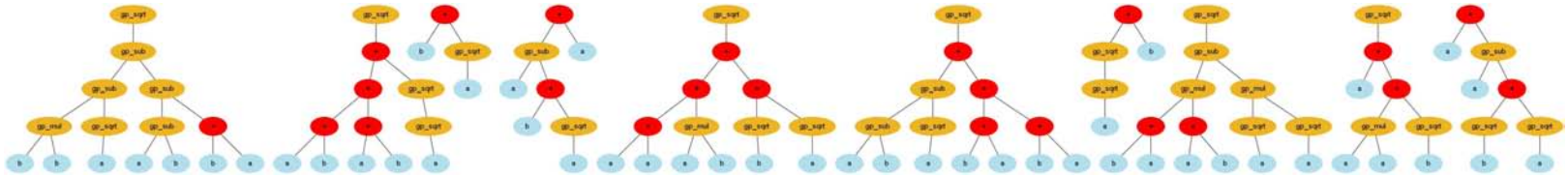


Fig. 13: Changing the node color and text.

About the author

Christian S. Perone (christian.perone@gmail.com) was born in Sarandi/RS (Brazil) in 1984 and currently lives in Porto Alegre/RS. He received his B.S. in Computer Science from University of Passo Fundo in Passo Fundo/RS and works for Compasso in Porto Alegre/RS.

Calls and Calendar

April 2010

Evostar 2010 - EuroGP, EvoCOP, EvoBIO and EvoWorkshops
April 7-9, 2010, Istanbul Technical University, Istanbul, Turkey
Homepage: www.evostar.org
Deadline November 11, 2009

The EuroGP, EvoCOP, EvoBIO and EvoApplications conferences compose EVO*: Europe's premier co-located events in the field of Evolutionary Computing.

Featuring the latest in theoretical and applied research, EVO* topics include recent genetic programming challenges, evolutionary and other meta-heuristic approaches for combinatorial optimisation, evolutionary algorithms, machine learning and data mining techniques in the biosciences, in numerical optimisation, in music and art domains, in image analysis and signal processing, in hardware optimisation and in a wide range of applications to scientific, industrial, financial and other real-world problems.

EVO* Poster

You can download the EVO* poster advertisement in PDF format here (Image: Pelegrina Galathea, by Stayko Chalakov (2009))

EVO* Call for Papers

You can download the EVO* CfP in PDF format here.

EuroGP

13th European Conference on Genetic Programming

EvoCOP

10th European Conference on Evolutionary Computation in Combinatorial Optimisation

EvoBIO

8th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics

EvoApplications 2010

European Conference on the Applications of Evolutionary Computation

- EvoCOMNET: 7th European Event on the Application of Nature-inspired Techniques for Telecommunication Networks and other Parallel and Distributed Systems
- EvoCOMPLEX (new): Evolutionary Algorithms and Complex Systems
- EvoENVIRONMENT: Nature Inspired Methods for Environmental Issues
- EvoFIN: 4th European Event on Evolutionary and Natural Computation in Finance and Economics
- EvoGAMES: 2nd European event on Bio-inspired Algorithms in Games
- EvoIASP: EC in Image Analysis and Signal Processing
- EvoINTELLIGENCE: Nature Inspired Methods for Intelligent Systems
- EvoMUSART: 8th European event on Evolutionary and Biologically Inspired Music, Sound, Art and Design
- EvoNUM: 3rd European event on Bio-inspired algorithms for continuous parameter optimisation
- EvoSTOC: 7th European event on Evolutionary Algorithms in Stochastic and Dynamic Environments
- EvoTRANSLOG: 4th European Event on Evolutionary Computation in Transportation and Logistics

EvoPHD

5th European Graduate Student Workshop on Evolutionary Computation

Evo* Coordinator: Jennifer Willies, Napier University, United Kingdom
j.willies@napier.ac.uk

Local Chair: Şima Uyar, Istanbul Technical University, Turkey
etaner@itu.edu.tr

Publicity Chair: Stephen Dignum, University of Essex, United Kingdom
sandig@essex.ac.uk

July 2010

GECCO 2010 - Genetic and Evolutionary Computation Conference
July 7-10, 2010, Portland, Oregon, USA
Homepage: http://www.sigevo.org/gecco-2010
Deadline January 13, 2010
Author notification: March 10, 2010
Camera-ready: April 5, 2010

The Genetic and Evolutionary Computation Conference (GECCO-2010) 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.

Organizers

General Chair:	Martin Pelikan
Editor-in-Chief:	Jürgen Branke
Local Chair:	Kumara Sastry
Publicity Chair:	Pier Luca Lanzi
Tutorials Chair:	Una-May O'Reilly
Workshops Chair:	Jaume Bacardit
Competitions Chairs:	Christian Gagné
Late Breaking Papers Chair:	Daniel Tauritz
Graduate Student Workshop	Riccardo Poli
Business Committee:	Erik Goodman
	Una-May O'Reilly
EC in Practice Chairs:	Jörn Mehnen
	Thomas Bartz-Beielstein,
	David Davis

Important Dates

Paper Submission Deadline	January 13, 2010
Decision Notification	March 10, 2010
Camera-ready Submission	April 5, 2010

Venue

The Portland Marriott Downtown Waterfront Hotel, located in downtown Portland, is near the Portland Riverplace Marina, restaurants, shopping & performing arts venues. Hotel room conference rate $179 includes complimentary in-room high-speed Internet access.

More Information

Visit www.sigevo.org/gecco-2010 for information about electronic submission procedures, formatting details, student travel grants, the latest list of tutorials and workshop, late-breaking papers, and more.

For technical matters, contact Conference Chair Martin Pelikan at pelikan@cs.umsl.edu.

For conference administration matters contact Primary Support Staff at gecco-admin@tigerscience.com.

GECCO is sponsored by the Association for Computing Machinery Special Interest Group for Genetic and Evolutionary Computation.

2010 IEEE World Congress on Computational Intelligence

July 18-23, 2010, Barcelona, Spain

Homepage: WWW

Deadline January 31, 2010

The 2010 IEEE World Congress on Computational Intelligence (IEEE WCCI 2010) is the largest technical event in the field of computational intelligence. It will host three conferences: the 2010 International Joint Conference on Neural Networks (IJCNN 2010), the 2010 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010), and the 2010 IEEE Congress on Evolutionary Computation (IEEE CEC 2010). IEEE WCCI 2010 will be held in Barcelona, a Mediterranean city located in a privileged position on the northeastern coast of Spain. Barcelona combines history, art, architecture, and charm within a pleasant, and efficient urban environment where meet old friends, and make new ones. The congress will provide a stimulating forum for scientists, engineers, educators, and students from all over the world to discuss and present their research findings on computational intelligence.

Important Due Dates

- Submission deadline: January 31, 2010
- Competition proposals: November 15, 2009
- Special sessions proposals: December 13, 2009
- Notification of special session acceptance: December 22, 2009
- Paper submission: January 31, 2010
- Tutorial and workshop proposal: February 14, 2010
- Notification of tutorial and workshop acceptance: February 22, 2010
- Notification of paper acceptance: March 15, 2010
- Final paper submission: May 2, 2010
- Early registration: May 23, 2010
- Tutorial and Workshops: July 18, 2010
- IEEE WCCI 2010 Conference: July 19, 2010

For more information visit http://www.wcci2010.org/call-for-papers

September 2010

PPSN 2010 – International Conference on Parallel Problem Solving From Nature

September 11-15, 2010, Cracow, Poland

Homepage: http://home.agh.edu.pl/ppsn

Deadline: April 6, 2010

The Eleventh International Conference on Parallel Problem Solving from Nature (PPSN XI) will be held at the AGH University of Science and Technology in Cracow, Poland on 11-15 September 2010. This biennial meeting aims to bring together researchers and practitioners in the field of natural computing. Natural Computing is the study of computational systems, which use ideas and get inspiration from natural systems, including biological, ecological, physical, chemical, and social systems. It is a fast-growing interdisciplinary field, in which a range of techniques and methods are studied for dealing with large, complex, and dynamic problems with various sources of potential uncertainties.

PPSN XI will be a showcase of a wide range of topics in Natural Computing including, but not restricted to: Evolutionary Computation, Neural Computation, Molecular Computation, Quantum Computation, Artificial Life, Swarm Intelligence, Artificial Ant Systems, Artificial Immune Systems, Self-Organizing Systems, Emergent Behaviors, and Applications to Real-World Problems. PPSN XI will also feature workshops and tutorials covering advanced and fundamental topics in the field of natural computation.

All accepted papers will be presented during poster sessions and will be included in the proceedings. Following the tradition of PPSN, proceedings will be published in the Series Lecture Notes in Computer Science (LNCS) by Springer.

Paper Presentation Following the now well-established tradition of PPSN conferences, all accepted papers will be presented during small poster sessions of about 16 papers. Each session will contain papers from a wide variety of topics, and will begin by a plenary quick overview of all papers in that session by a major researcher in the field. Past experiences have shown that such presentation format led to more interactions between participants and to a deeper understanding of the papers. All accepted papers will be published in the Proceedings.

General Chair

Robert Schaefer (AGH, Cracow, PL)

Honorary Chair

Hans-Paul Schwefel (Tech. Universität Dortmund, DE)

Program Co-Chairs

Carlos Cotta (University of Malaga, ES)
Joanna Kolodziej (University of Bielsko-Biala, PL)
Günter Rudolph (Tech. Universität Dortmund, DE)

Tutorials Chair

Krzysztof Cetnarowicz (AGH, Cracow, PL)

Workshop Chair

Aleksander Byrski (AGH, Cracow, PL)

Important dates

Workshop Proposals Submission	January 3, 2010
Workshop Proposals Notification	February 19, 2010
Paper Submission	April 6, 2010
Author Notification	May 21, 2010
Papers Camera Ready Submission	June 11, 2010
Early Registration	June 11, 2010
Conference	September, 11-15, 2010

Seventh International Conference on Swarm Intelligence
September 8-10, 2010. Brussels, Belgium
Homepage: http://iridia.ulb.ac.be/ants2010
Deadline February 28, 2010

Swarm intelligence is a relatively new discipline that deals with the study of self-organizing processes both in nature and in artificial systems. Researchers in ethology and animal behavior have proposed many models to explain interesting aspects of social insect behavior such as self-organization and shape-formation. Recently, algorithms and methods inspired by these models have been proposed to solve difficult problems in many domains.

An example of a particularly successful research direction in swarm intelligence is ant colony optimization, the main focus of which is on discrete optimization problems. Ant colony optimization has been applied successfully to a large number of difficult discrete optimization problems including the traveling salesman problem, the quadratic assignment problem, scheduling, vehicle routing, etc., as well as to routing in telecommunication networks.

Another interesting approach is that of particle swarm optimization, that focuses on continuous optimization problems. Here too, a number of successful applications can be found in the recent literature. Swarm robotics is another relevant field. Here, the focus is on applying swarm intelligence techniques to the control of large groups of cooperating autonomous robots.

ANTS 2010 will give researchers in swarm intelligence the opportunity to meet, to present their latest research, and to discuss current developments and applications.

The three-day conference will be held in Brussels, Belgium, on September 8-10, 2010. Tutorial sessions will be held in the mornings before the conference program.

Relevant Research Areas

ANTS 2010 solicits contributions dealing with any aspect of swarm intelligence. Typical, but not exclusive, topics of interest are:

- Behavioral models of social insects or other animal societies that can stimulate new algorithmic approaches.
- Empirical and theoretical research in swarm intelligence.
- Application of swarm intelligence methods, such as ant colony optimization or particle swarm optimization, to real-world problems.
- Theoretical and experimental research in swarm robotics systems.

Publication Details As for previous editions of the ANTS conference, proceedings will be published by Springer in the LNCS series (to be confirmed). The journal Swarm Intelligence will publish a special issue dedicated to ANTS 2010 that will contain extended versions of the best research works presented at the conference.

Best Paper Award

A best paper award will be presented at the conference.

Further Information

Up-to-date information will be published on the web site http://iridia.ulb.ac.be/ants2010/. For information about local arrangements, registration forms, etc., please refer to the above-mentioned web site or contact the local organizers at the address below.

Conference Address

ANTS 2010
IRIDIA CP 194/6 — Tel +32-2-6502729
Université Libre de Bruxelles — Fax +32-2-6502715
Av. F. D. Roosevelt 50 — http://iridia.ulb.ac.be/ants2010
1050 Bruxelles, Belgium — email: ants@iridia.ulb.ac.be

Important Dates

Submission deadline	March 28, 2010
Notification of acceptance	April 30, 2010
Camera ready copy	May 14, 2010
Conference	September 8–10, 2010

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 solicit 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.