SIGEVOlution

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

Volume 4
Issue 3

in this issue

Editorial

You might be puzzled for receiving a new issue this soon. But it's actually good news! It means that, as promised, I am finally catching up with the backlog accumulated last year. The plan is to close the fourth volume by March, so that, for the fifth SIGEVOlution birthday, in April 2010, the newsletter will be right on schedule!

Meanwhile, we have two new major deadlines approaching. By March 3, 2010 many of us have to deliver their reviews for GECCO-2010. But most importantly, March 25, 2010 is the deadline for the submission to the 13 GECCO-2010 workshops. Yes, you got it right. GECCO-2010 will host 13 workshops. That's a lot of submission opportunities.

This new issue offers another juicy preview of Arthur Kordon's book "Applying Computational Intelligence: How to Create Value" and some interesting software. The first article is in fact an extract of Arthur's book discussing the major issues in applying computational intelligence in practice. In the second article, Patrick Stalph and Martin Butz present their new Java implementation of the XCSF Classifier System. Then, the columns, as always, provide information about new PhD theses, new issues of journals, and the forthcoming events.

My due thanks to the people who made this issue possible: Arthur Kordon, Patrick Stalph, Martin Butz, Xavier Llorà, Kumara Sastry, Cristiana Bolchini, Mario Verdicchio, Viola Schiaffonati, and board members Dave Davis and Martin Pelikan.

And remember, 13 workshops are waiting for us at GECCO-2010. The deadline is March 25, 2010!

The cover is a shot of Portland skyline by David Gn.

Pier Luca
February 14th, 2010

SIGEVOlution Volume 4, Issue 3

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

SIGEVO Officers

Darrell Whitley, Chair
John Koza, Vice Chair
Una-May O'Reilly, Secretary
Wolfgang Banzhaf, Treasurer

SIGEVOlution Board

Pier Luca Lanzi (EIC)
Lawrence "David" Davis
Martin Pelikan

Contributors to this Issue

Arthur Kordon
Patrick O. Stalph
Martin V. Butz

Contents

ISSN: 1931-8499

Issues in Applying Computational Intelligence

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

Applying any emerging technology is not trivial and requires some level of risk-taking even when the competitive advantage is clear. In the case of computational intelligence the process is even harder due to the different nature of the comprising methods, the lack of marketing, affordable professional tools and application methodology. Another important factor slowing down computational intelligence applications is the wrong perception of the technology. To many potential users it looks like it's either too expensive or it's rocket science. The pendulum of expectations also swings from one extreme of anticipating a silver bullet to all problems to the other extreme of awaiting the next technology fiasco.

This article focuses on the most important application issues of computational intelligence, shown in the mind-map in Figure 1. Their understanding and resolution is critical for the success of applied computational intelligence.

Technology Risks

The first application issue is related to the generic factors which lead to unsuccessful introduction of technology. From the multitude of potential causes of technology failure we'll focus on four: (1) balance between the user crisis pushing for introduction of technology and the total perceived pain of adaptation of the new technology; (2) the prevailing technocentric culture; (3) increased complexity as a result of the introduction of technology; and (4) technology hype. These factors are shown in the mind-map in Figure 2.

The Change Function

In a recent book about introducing new technologies, Pip Coburn has proposed the idea about The Change Function as the key driving force for opening the door to novel solutions. According to The Change Function, people are only willing to change and accept new technologies when the pain of their current situation outweighs the perceived pain of trying something new[1], i.e.,

The Change Function = f(user crisis vs. total perceived pain of adoption)

The arguments for The Change Function are as follows. Potential users have some level of reaction from indifference to crisis whenever they encounter a new technology product. People are more willing to change the higher the level of crisis that they have in their current situation.

The total perceived pain of adaptation is the perception of how painful it will be to actually adopt the product and especially change a habit. The Change Function looks at both sides of the issue of emerging technologies. On the one hand, it defines the critical level of the services a new technology might offer. On the other hand, it evaluates the total perceived pain of adoption associated with that new service.

Applying The Change Function to computational intelligence suggests that the technology must minimize the total perceived pain of adoption of potential users by offering a user-friendly environment, simple solutions, and easy integration into the existing work processes.

[1] The Change Function, Penguin Group, 2006.

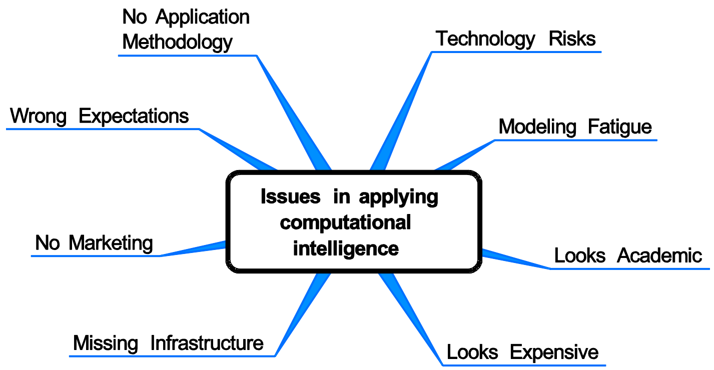


Fig. 1: Key issues in applying computational intelligence.

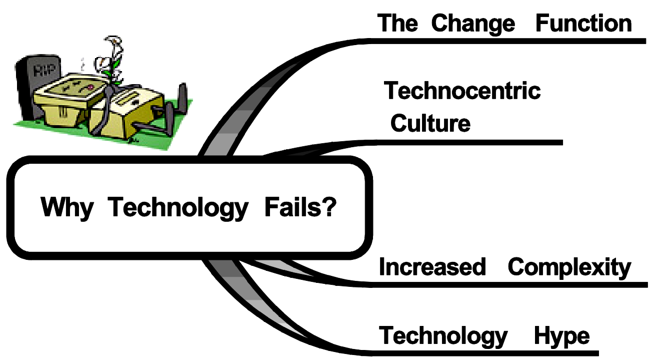


Fig. 2: Key factors for failure of new technology.

The other factor for success is the matching of the unique capabilities of computational intelligence to a critical problem of the user. Usually these are the cases when computational intelligence has a clear competitive advantage, i.e. a novel solution is needed in a dynamic environment of high complexity or uncertainty.

Technocentric Culture

According to Clayton Christensen – one of the leading gurus in technology innovations – three quarters of the money spent on product development investment results in products that do not succeed commercially[2]. It is observed that most new technologies are hated by users due to the high total pain of adoption. A possible cause for this unpleasant result is the prevailing technocentric culture in industry. It could be defined as an obsession with the critical role of the technology while neglecting the real customer needs, i.e., the famous Levitt's Law is totally ignored:

Levitt's Law: When people buy quarter-inch drill bits, it's not because they want the bits themselves. People don't want quarter-inch drill bits – they want quarter-inch holes. Ted Levitt

A typical result of the technocentric culture is pushing technology improvement at any cost by management. Often introducing new emerging technologies is part of this process and as such may become a potential issue for applied computational intelligence. Imposing the technology for purely technology's sake may lead to lost credibility, as we know from applied Artificial Intelligence (AI). In principle, computational intelligence development requires higher implementation efforts, i.e., high total perceived pain of adoption. It is critically important to justify the user needs before suggesting the introduction of technology. The new technology has to be transparent and the life of the user must be easier not harder after the technology is applied. An example of what the user does not need is the Microsoft Office advisor, based on computational intelligence and introduced in the late 1990s. The initial expectations of Microsoft to deliver a killer computational intelligence application evaporated soon through the unanimous cold response from the users. Absolutely boring and useless, it was removed by almost all users of the popular product.

Increased Complexity

Increased complexity of applied new solutions is the key root cause for the high total perceived pain of adoption. It is especially painful when the application requires changing a habit. Modifying an established work process is a potential disaster as well. Adding new equipment and pushing to learn new software helps the negative response of the user.

[2] C. Christensen and M. Raynor, The Innovator's Solution, Harvard Business School Press, 2003.

In many cases the imposed complexity is well balanced with the benefits of the applied computational intelligence system. A typical example is the nonlinear control systems based on neural networks. The advantages of using this type of system to control processes with difficult nonlinear behavior partially overcomes the pain of complex tuning. It has to be taken into account that tuning the neural network-based controllers requires 10-12 parameters in comparison to three in the prevailing PID controllers.

Technology Hype

Another factor that contributes to technology failure is eroding the credibility by overselling its capabilities. Unfortunately, technology hype is an inevitable evil in the introduction of emerging technologies which needs to be taken into account and counteracted with realistic expectations. We will focus on the following key sources of technology hype:

- Technology hype from vendors — Usually exaggerates the capabilities for direct value creation while it hides the application issues, especially the potential for growing maintenance cost. The ultimate example is one of the early slogans "We will turn your data into gold" of one of the leading vendors of neural networks-based systems.
- Technology hype from management — It is like the regular top-down push of corporate initiatives. In principle, it is slightly more realistic, since it is based on some practical experience. The potential topic of exaggeration is mostly the leveraging capacity of the technology.
- Technology hype from R&D — It may come from the top academics in the field, which usually are entirely convinced that their methods are The Ultimate Solution Machine to almost any problem in the Universe. Another source could be industrial R&D which compensates for the lack of professional marketing of the technology with typical oversell.
- Technology hype from the media — Applied AI in the past and recently different computational intelligence methods are periodically the focus of attention from the media. Sometimes the media-generated hype goes beyond any sensible limits. A typical case was the absurd story of using all the power of computational intelligence to create an intelligent litter box for cats.

Modeling Fatigue

Using modeling for new product design and process improvement has a long history in industry. Many profitable businesses did significant investments in developing and deploying different types of models for their critical operations. In some cases, the modeling efforts have already reached the saturation state when almost everything necessary for process optimization and advanced monitoring and control has been accomplished. As a result, the opportunities for introducing and applying new modeling approaches are limited.

The key features that are at the basis of this critical challenge for successful application of computational intelligence are captured in the mind-map in Figure 3 and discussed below.

The Invasion of First-Principles Models

The clearest feature of modeling saturation is the case of mass applications of models based on the laws of Nature. Here is a short list for when a business needs first-principles models:

- Understand process physics and chemistry;
- Design model-based optimization and control systems;
- Expect long-term profit from manufacturing;
- Process has manageable complexity and dimensionality;
- Expected return from modeling investment is high.

At the basis of the decision to initiate a costly program of fundamental modeling is the assumption that the future productivity gains will justify the expense. Another factor in support of first-principles modeling is the increased efficiency of the recent software environments from vendors like Aspen Technologies, Fluent, and Comsol Inc. In addition, fundamental modeling includes the best available domain experts in the specific modeling areas and that builds credibility in the implemented models.

On the negative side, however, first-principles modeling creates a barrier to other modeling approaches. One factor is the reluctance of management to invest more money after the high cost already spent on fundamental model development and deployment.

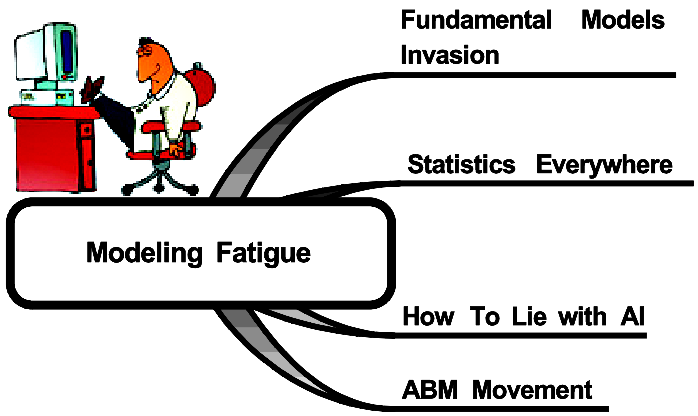


Fig. 3: Key symptoms of modeling fatigue.

The key factor, however, is the united front from the developers of mechanistic models against empirical solutions. While the principal argument of the advantages of the laws of Nature over empiricism is difficult to argue, going to the other extreme of ignoring data-driven solutions is also not acceptable.

Statistical Models Everywhere

The biggest generator of models in industry, however, is not the laws of Nature but the laws of Numbers. Thanks to Six Sigma and the increased role and availability of data the number of applied statistical models in industry is probably orders of magnitude higher than the number of first-principles models. While the advantages of statistical models are clear, mass-scale applications create some issues that deserve attention.

The key issue is creating the image of statistics as a universal solution to all data-related problems. Part of the problem is the opposition of many professional statisticians to some empirical methods that are not statistically blessed, such as neural networks and symbolic regression. As a result, the empirical modeling opportunities are implemented almost entirely as statistical models. This is a big challenge to computational intelligence since significant marketing efforts are needed to promote the technology, especially to Six Sigma black belts and professional statisticians.

A good starting point could be a realistic assessment of the performance of existing statistical models. There is a high probability that at least a part of them will need significant improvement. One of the potential alternative solutions is using symbolic regression models.

How to Lie with AI

A special category of models is heuristic-based applications, which emphasize the expert role at the expense of either first-principles or empirical approaches. Unfortunately, the average credibility of some of these models in industry is not very high (see [1, Chapter 1]).

Anything But Model (ABM) Movement

An unexpected result of modeling fatigue due to the mass-scale modeling efforts is the appearance of the so-called Anything But Model (ABM) movement. It includes the model users who are disappointed by the poor performance of applied models and tired of the administrative modeling campaigns. They are strong opponents to introducing new models and prefer direct human actions, based on their experience. Several factors contribute to this attitude. One of the factors is that the excess of models leads to confusion. At least some of the models do not predict reliably and the users gradually lose patience and confidence in using them. Another factor is the growing maintenance cost with the increased number and complexity of models. There is also a critical threshold of modeling intensity beyond which the saturation of the users becomes obvious.

Looks Too Academic

Another issue of applying computational intelligence is its academic image in industry. Part of the problem is the lack of popular references that would help the potential users to understand the diverse computational intelligence approaches. The dynamic growth of the technology makes it difficult to track the state of the art even for researchers, not to mention practitioners. These key factors are shown in the mind-map in Figure 4 and discussed in this section.

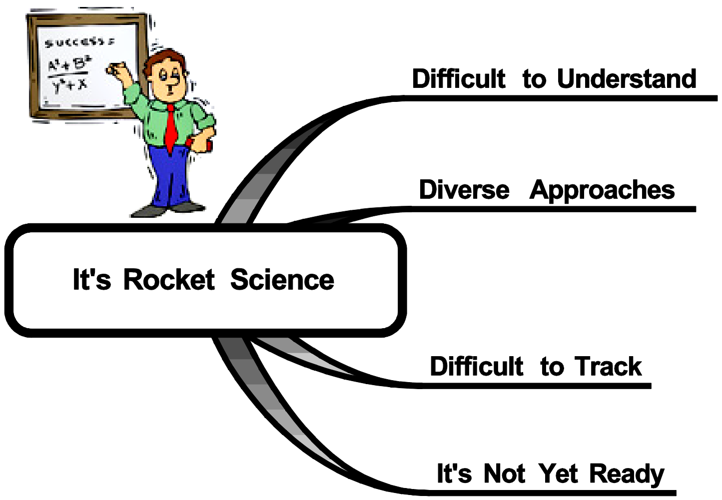


Fig. 4: A mind-map with the main reasons why computational intelligence looks too academic.

Difficult to Understand

One of the reasons for the academic image is the lack of understanding of the technology outside several closed research communities. A number of issues contribute to this situation. First, undergraduate and graduate courses on computational intelligence are offered in very few universities. The technology is virtually unknown to most students in related technical disciplines and this significantly narrows the application opportunities. It is extremely difficult for potential users to find popular explanation of the key computational intelligence approaches without scientific jargon and heavy math[3]. It has to be taken into account that some of the methods, especially support vector machines, are not easy to translate into plain English.

The other option of understanding and learning the capabilities of computational intelligence by playing with popular and user-friendly software is also virtually non-existent.

[3] One of the few exceptions is the book by V. Dhar and R. Stein, *Seven Methods for Transforming Corporate Data into Business Intelligence*, Prentice Hall, 1997.

Diverse Approaches

An additional factor contributing to the academic image of computational intelligence is the wide diversity of the comprising approaches. They differ in scientific principles, mathematical basis, and user interaction. It is extremely challenging even for a person with a sound technical and mathematical background to be easily introduced to all approaches. The confusion is also increased by the still divisive environment among the research communities developing the different methods. Usually each community tries to glorify the role of the specific approach as the ultimate technology, very often at the expense of the others. The spirit of competition prevails over the efforts of pursuing the synergetic benefits. There are very few popular references and practical guidelines on integration of these approaches, which is critical for the success of real-world applications.

Difficult to Track

The fast speed of growth of the existing computational intelligence methods and the emergence of numerous new approaches contributes to the academic image of the technology as well. This makes the task of tracking the scientific progress of computational intelligence very time-consuming and more complex than looking at state of the art of an average technology. For example, to cover the whole field it is necessary to keep abreast of at least five scientific journals, such as the IEEE Transaction on Neural Networks, IEEE Transaction on Fuzzy Systems, IEEE Transaction on Evolutionary Computation, IEEE Transaction on Systems, Man, and Cybernetics, and IEEE Intelligent Systems Journal, and the yearly proceedings of several top conferences, organized by the different scientific communities.

Even more difficult is keeping track of available software, and especially of successful real-world applications. The potential sources of information are very limited. Fortunately, there is a trend of a more sizable industrial presence in some of the conferences in special sessions and workshops devoted to practical issues of computational intelligence.

It's Not Yet Ready for Industry

A negative consequence from the fast scientific progress of computational intelligence is the perception of technology and incompleteness it creates among practitioners.

In principle, it is much more difficult to convince a potential user to apply a technology that is still in the high-risk dynamic development phase. In addition, management is reluctant to give support for a technology which may require continuous internal R&D development.

The limited knowledge about industrial success stories based on computational intelligence and the lack of affordable professional software contribute to the image of technology immaturity as well.

Perception of High Cost

The prevailing academic impression of computational intelligence in industry is combined with the perception of high total cost of ownership of the applied solutions. It is expected that research-intensive technology will require higher-than-average development cost. The investment in new infrastructure and the necessary training will increase the deployment cost as well. To potential users, it seems that the maintenance cost will also be high due to the specialized skills of the support teams. All of these application issues are presented in the mind-map in Figure 5 and discussed briefly below.

Growing R&D Cost

Unfortunately, the scientific expansion of computational intelligence creates the logical estimate not only of high development cost but even expectation for growing R&D expenses in the future. The technology is still treated by the management as research-intensive and the diverse methods on which it is based create the impression of a steep learning curve. The estimated high development cost is based on the assumption that probably external resources from academia or vendors will be needed. In some cases the development cost will be even higher if internal R&D efforts are necessary to improve the technology for important specific classes of applications. An additional factor in the high development cost of computational intelligence is that the opportunity analysis for potential applications is not trivial and requires more than average efforts for exploratory analysis.

Expensive Infrastructure

The assessment of high deployment cost is based on the assumption of necessary investment for the computational intelligence infrastructure. In some cases, such as intensive use of evolutionary computation, investment in more powerful computer hardware, such as clusters, is recommended. Unfortunately, the cost of available professional software is relatively high. Very often, it is necessary to allocate resources for internal software development and support, which also significantly increases the cost.

Expected Training Cost

A significant share of the expected high total cost of ownership of applied computational intelligence is related to training. Firstly, it is a nonstandard training for developers with an expected steep curve of understanding for the key approaches. Secondly, it could be very difficult training for the users, since most of them do not have the technical and mathematical skills of the developers. Thirdly, training tools explaining the computational intelligence approaches in plain English are practically unavailable.

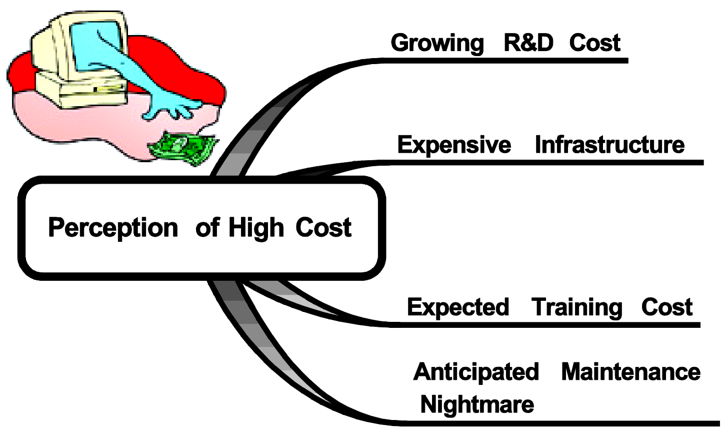


Fig. 5: Key factors that contribute to the image of high total cost of ownership of computational intelligence.

Anticipated Maintenance Nightmare

The fear of difficult maintenance and gradually reducing performance of applied computational intelligence solutions is the key concern to most potential users. The little experience from the known industrial applications of computational intelligence and the maintenance lessons from applied AI contribute to this perception. It is unclear to potential industrial users if it is possible to support the applied computational intelligence systems without Ph.D.-level skills. An additional source of concern is the uncertain long-term future of most of the computational intelligence vendors and their limited capabilities for global support.

Missing Infrastructure

Clarifying the requirements for the necessary infrastructure to support applied computational intelligence is of key concern to potential users of this emerging technology. Unfortunately, there are very few available sources to discuss this issue of critical importance for the final decision of whether or not to give the green light for the promotion of technology.

The objective of this section is to address the key concerns related to the computational intelligence infrastructure, represented in the mind-map in Figure 6.

Specialized Hardware

The first infrastructural question that needs to be clarified to the potential users of computational intelligence is the possible need for more powerful hardware than high-end personal computers. Fortunately, with the exception of evolutionary computation, there are no special requirements for high computational power. The needs for evolutionary computation are problem-dependent. In the case of high-dimensionality data or use of complex simulation packages (as is the situation with applying genetic programming for inventing electronic circuits) computer clusters or grid computing is a must. Most evolutionary computation algorithms are inherently parallel and benefit from distributed computing.

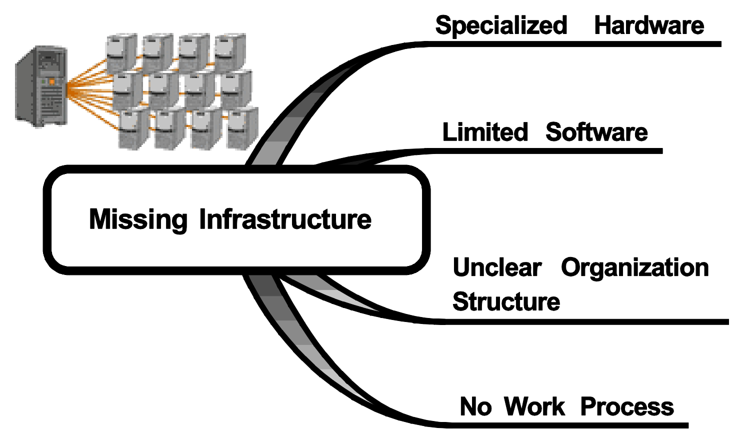


Fig. 6: A mind-map representing the key issues related to the required infrastructure for applied computational intelligence.

Limited Software

The situation is much more challenging with the other component of the applied computational intelligence infrastructure – the available software. Unfortunately, an affordable software platform which integrates all key approaches discussed in this book is not available. Different approaches have been integrated in some commercial packages, such as Intelligent Business Engines™ (NuTech Solutions), SAS Enterprise Miner (SAS Institute), and Gensym G2 (Gensym Corporation). However, the price of the software is relatively high and affordable for big corporate clients only.

A more realistic option is to build a software infrastructure with commercial products. Several vendors offer professional user-friendly software based on specific methods, mostly neural networks. Typical examples of off-line solutions are Neurodynamics (NeuroSolutions) and Neuroshell (Ward Systems Group). The leading products for on-line neural network applications in advanced process control and environmental compliance are ValueFirst™ (Pavilion Technologies) and Aspen IQModel (Aspen Technologies).

An interesting option is the commercial software which is embedded in popular products, such as Excel. The advantage of this solution is the minimal training for development and support. Examples are products like NeuralWorks Predict (Neuralware) and NeuralTools (Palisade).

The most realistic option to find software for most of the discussed approaches is on universal modeling platforms such as MATLAB (Mathworks) and Mathematica (Wolfram Research). In addition to commercially available toolboxes on these platforms, there are many free packages developed from academia. However, their user-friendliness is far from that desired for real-world applications.

Unclear Organization Structure

The third infrastructural question of interest to potential users of computational intelligence is the possible need for organizational changes. The answer is not trivial, especially for large corporations. In the case of expected high return from applying the technology, a special group of developers can be established. The objective of an applied computational intelligence group is to own the technology inside the company. In this most favorable scenario for introducing the technology, there is a clear focal point for internal marketing, opportunity analysis, potential technology improvement, application development, deployment and maintenance.

Another possible organizational option is to establish a small group within a modeling-type of R&D department with more limited tasks of focusing on opportunity analysis for specific computational intelligence applications with minimal development efforts. The third organizational scenario is when the individual computational intelligence experts are spread out in different groups and promote the technology through professional networks.

Work Process Not Defined

The fourth infrastructural question of interest to potential users of computational intelligence is the possible need to change existing work processes or define new ones, related to the specifics of the technology. In principle, solution development, implementation, and support is not significantly different from other technologies and probably doesn't require the introduction of new work processes.

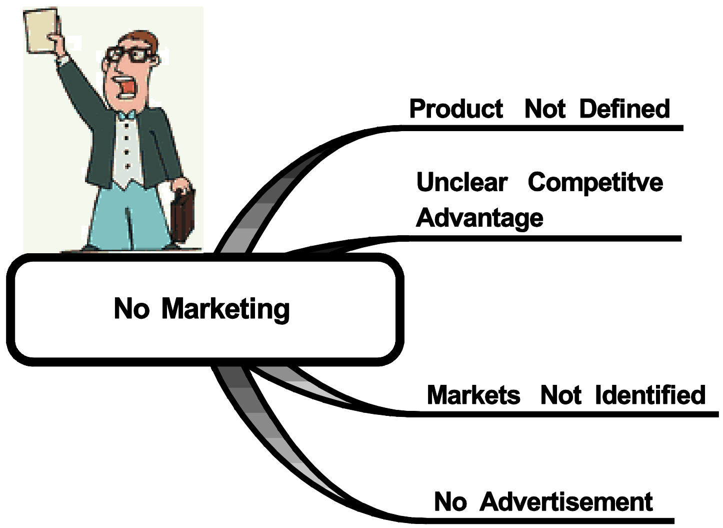


Fig. 7: A mind-map of the key issues to be resolved for professional promotion of computational intelligence in industry.

However, the need for nontrivial internal marketing of the unique computational intelligence capabilities will demand adequate work process. The ideal scenario is to link the computational intelligence application methodology into the existing standard work processes, like Six Sigma.

No Marketing

The most critical issue of applied computational intelligence from a practical point of view is the lack of professional marketing of this emerging technology. As a result, we have the paradox when the technology is introduced to potential users and markets through the "back door" by individual efforts of R&D enthusiasts. The need to open the field to industry through the "front door" with professional marketing is discussed briefly in this section. The key issues that need to be resolved are captured in the mind-map in Figure 7 and addressed below.

Product Not Clearly Defined

The first challenge in professional marketing of applied computational intelligence is the nontrivial definition of the final product. There are several sources of confusion, such as the wide diversity of methods and the broad application areas, to name a few. The deliverables are also extremely different. Here are some obvious examples of the main types of products, derived from applied computational intelligence:

- Predictive models;
- Problem classifiers;
- Complex optimizers;
- System simulators;
- Search engines.

They could be the basis of a broad definition of the expected products from this emerging technology. For marketing purposes it could be advertised that applied computational intelligence enhances the productivity of human intelligence through complex predictive models, problem classifiers, optimizers, system simulators, and search engines.

Unclear Competitive Advantages

This aspect is discussed at length in the ninth chapter of the book [1].

Key Markets not Identified

The broad view of the markets with most perspective is given in the second chapter of the book [1]. However, more specific efforts for identifying the opportunities and the size of individual markets are needed.

No Advertisement

A significant part of the marketing efforts is advertising the technology to a very broad nontechnical audience of potential users. The advertisement objective is to define a clear message for representing the technology and capturing attention. The key principle is focusing on the unique deliverables which give competitive advantage. Specific marketing slides and examples of elevator pitches are given in the corresponding chapters for each computational intelligence approach. In addition, examples of advertising applied computational intelligence to nontechnical and technical audiences are given in [1, Chapter 13].

Wrong Expectations

Probably the most difficult issue of applied computational intelligence is to help the final user in defining the proper expectations from the technology. Very often the dangerous combination of lack of knowledge, technology hype, and negative reception from some applied research communities creates incorrect anticipation about the real capabilities of computational intelligence. The two extremes of wrong expectations either by exaggeration or by underestimation of the computational intelligence capabilities cause almost equal damage to promotion of the technology in industry.

The key ways of expressing wrong expectations about applied computational intelligence are shown in the mind-map in Figure 8 and discussed below.

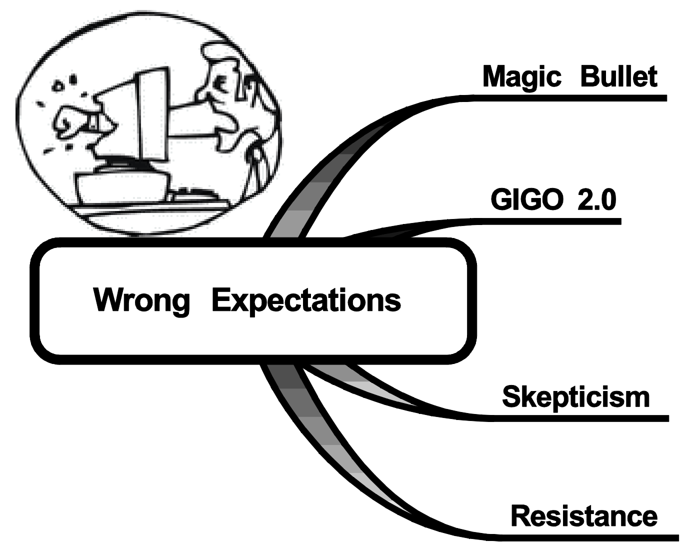


Fig. 8: Key ways of expressing wrong expectations from applied computational intelligence.

Magic Bullet

The expectation for technical magic is based on the unique capabilities of applied computational intelligence to handle uncertainty, complexity, and to generate novelty. The impressive features of the broad diversity of methods, such as fuzzy logic, machine learning, evolutionary computation, and swarm intelligence, contribute to such a Harry Potter-like image even when most of the users do not understand the principles behind them. Another factor adding to the silver bullet perception of computational intelligence is the technology hype from the vendors, the media, and some high-ranking managers.

As a result, potential users look at applied computational intelligence as the last hope to resolve very complex and difficult problems. Often, they begin looking at the technology after several failed attempts of using other methods. In some cases, however, the problems are ill-defined and not supported by data and expertise. In order to avoid the magic bullet trap, it is strongly recommended to identify the requirements, communicate the limitations of the appropriate methods, and to define realistic expectations in the very early phase of potential computational intelligence applications.

GIGO 2.0

The worst-case scenario of the magic bullet image is the GIGO 2.0 effect. In contrast to the classical meaning of GIGO 1.0 (Garbage-In-Garbage-Out), which represents the ignorant expectations of a potential solution, GIGO 2.0 embodies the next level of arrogant expectations defined as Garbage-In-Gold-Out. In essence, this is the false belief that low-quality data can be compensated for with sophisticated data analysis. Unfortunately, computational intelligence with its diverse capabilities to analyze data is one of the top-ranking technologies that create GIGO 2.0 arrogant expectations. It is observed that the bigger the disarray with data the higher the hope of exotic unknown technologies to clean up the mess. Usually this behavior is initiated by top management who are unaware of the nasty reality of the mess. It is strongly recommended to protect potential computational intelligence applications from the negative consequences of the GIGO 2.0 effect. The best winning strategy is to define the requirements and the expectations in advance and to communicate clearly to the user the limitations of the methods. Better to reject an impossible implementation than to poison the soil for many feasible computational intelligence applications in the future.

Skepticism

In contrast to the magic bullet optimistic euphoria, disbelief and lack of trust in the capabilities of applied computational intelligence is the other extreme of wrong expectation, but in this case on the negative side. Usually skepticism is the initial response of the final users of the technology on the business side. Several factors contribute to this behavior, such as lack of awareness of the technical capabilities and the application potential of computational intelligence, lessons from other over-hyped technology fiascos in the past, and caution from ambitious R&D initiatives pushed by management.

Skepticism is a normal attitude if risk is not rewarded. Introducing emerging technologies, like computational intelligence, requires a risk-taking culture from all participants in this difficult process. The recommended strategy for success and reducing skepticism is to offer incentives to the developers and the users of the technology.

Resistance

The most difficult form of wrong expectations from computational intelligence is actively opposing the technology due to scientific or political biases. In most cases the driving forces of resistance are part of the industrial research community which either does not accept the scientific basis of computational intelligence or feels threatened by its application potential.

Usually the resistance camp against computational intelligence is led by the first-principles modelers. They challenge almost any empirical method in general, and have severe criticisms against black-box models in particular. Their opposition strategy is to emphasize the advantages of first-principles models, which are well known, and to ignore the benefits of computational intelligence, which unfortunately are not familiar to a broad audience. Often first-principles modelers are very aggressive in convincing management to question the capabilities and the total-cost-of-ownership of computational intelligence. In most of the cases they silently reject the opportunities for collaboration with computational intelligence developers and to deliver an integrated solution.

The other part of the resistance movement against computational intelligence includes the professional statisticians and a fraction of the Six Sigma community.

Most professional statisticians do not accept the statistical validity of some of the computational intelligence methods, especially neural networks-based models. The key argument of these fighters for the purity of the statistical theory is the lack of a statistically sound confidence metric of the nonlinear empirical solutions derived by computational intelligence. They have support in the large application base of the Six Sigma community, since classical statistics is the key application method.

The third part of the resistance camp against computational intelligence includes the active members of the Anything But Model (ABM) movement who energetically oppose any attempt to introduce the technology. In this category we can also add the users expecting high perceived pain of adoption from computational intelligence.

No Application Methodology

Another application issue of computational intelligence is the confusion of potential users about how to implement the technology in practice. In contrast to the key competitive approaches, such as first-principles modeling, statistics, heuristics, and optimization, there is no well-known application methodology for computational intelligence. Potential users have difficulties selecting the proper methods for solving their problems. They don't know how to take advantage of integrating the different approaches. The specific application sequence is also unclear to users and there are very few references to answer the practical application questions related to computational intelligence.

These key issues related to the lack of computational intelligence application methodology are represented in the mind-map in Figure 9 and discussed briefly below.

Integration Advantages

Unfortunately, most potential users are unaware of the big benefits of integrating the different computational intelligence approaches. This gap is partially filled with the different integration options discussed in my book [1, Chapter 11]. An integrated methodology based on statistics, neural networks, support vector machines, genetic programming, and particle swarm optimization is presented, and illustrated with industrial examples.

Application Sequence

Potential computational intelligence users also need more clarity on the specific steps in applying the technology. This issue is covered in my book [1, Chapter 12] with two options for potential users. The first option is a generic computational intelligence application methodology as a separate work process at the application organization. The second option is directed to the potential users of Six Sigma and includes practical suggestions on how to integrate computational intelligence with this established work process.

Few References

Potential users of computational intelligence face another challenge due to the very few sources with practical information about the applicability of the different methods. Some vendors offer a development methodology in their manuals for a specific approach, mostly related to neural networks or genetic algorithms. However, this information is very limited and usually requires specialized training. In addition, parameter settings for the different approaches are not easy to find, which is very important from a practical point of view.

Fortunately, this gap is filled in the different chapters of [1]. The necessary information about the application sequence and parameter settings of the different approaches, as well as the relevant references of practical interest, are given in the corresponding chapters.

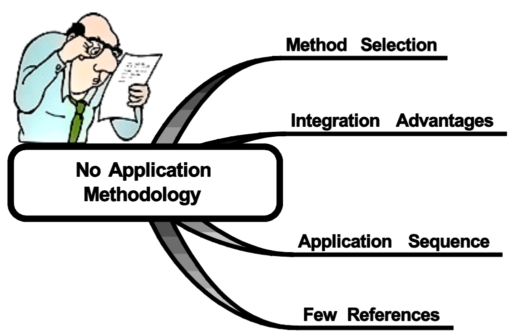


Fig. 9: A mind-map with the gaps in computational intelligence application methodology.

Key Messages

- A necessary condition for successful application of computational intelligence is that the benefits of the solution will outweigh the perceived pain of technology adoption.
- The academic image of computational intelligence creates a perception of high total-cost-of-ownership and alienates potential users.
- The missing computational intelligence infrastructure, due to limited professional software, unclear organizational structure, and undefined work processes, reduces the application opportunities of the technology.
- Marketing the competitive advantages and application capabilities of computational intelligence is the key step to open the door into industry.
- Wrong expectations about computational intelligence can either destroy the credibility of the technology or prevent its introduction.

The Bottom Line

The key application issues of computational intelligence are mostly related to insufficient knowledge of the real capabilities of the technology.

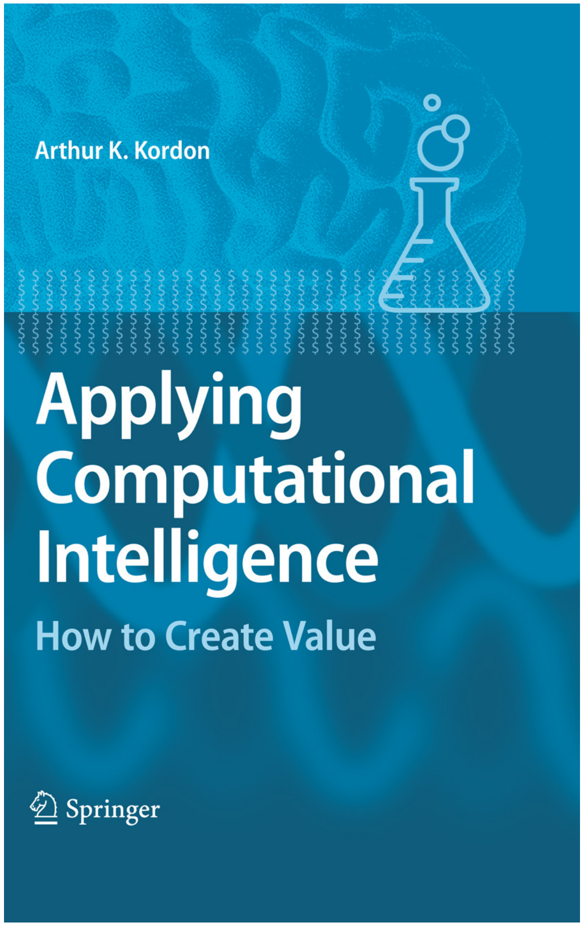
Suggested Reading

Three books with a good summary of effective innovation strategies are:

- C. Christensen and M. Raynor, The Innovator's Solution, Harvard Business School Press, 2003 (Amazon).
- P. Coburn, The Change Function, Penguin Group, 2006. (Amazon)
- M. George, J. Works, and K.Watson-Hemphill, Fast Innovation: Achieving Superior Differentiation, Speed to Market, and Increased Profitability, McGraw-Hill, 2005. (Amazon)

Bibliography

[1] Kordon, Arthur *Applying Computational Intelligence How to Create Value* Springer, 2010, ISBN: 978-3-540-69910-1



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

FOGA 2011 - Foundations of Genetic Algorithms

January 5-9, 2011, Schwarzenberg, Austria
http://www.sigevo.org/foga-2011
Enquiries and Submissions: foga@fhv.at
Deadline Monday July 5, 2010

We invite submissions of extended abstracts for the eleventh Foundations of Genetic Algorithms workshop. FOGA is only held every two years and focuses on theoretical foundations of all flavors of evolutionary computation. It will next be held in the Gasthof Hirschen hotel in Schwarzenberg in Austria from Wednesday, January 5 to Sunday January 9, 2011. Prof. Dr. Karl Sigmund has agreed to deliver a keynote lecture. Attendance is limited to people who submitted papers, or those requesting attendance in advance. Students are particularly encouraged to participate.

Submissions should address theoretical issues in evolutionary computation. Papers that consider foundational issues, place analysis in the wider context of theoretical computer science, or focus on bridging the gap between theory and practice are especially welcome. This does not prevent the acceptance of papers that use an experimental approach, but such work should be directed toward validation of suitable hypotheses concerning foundational matters.

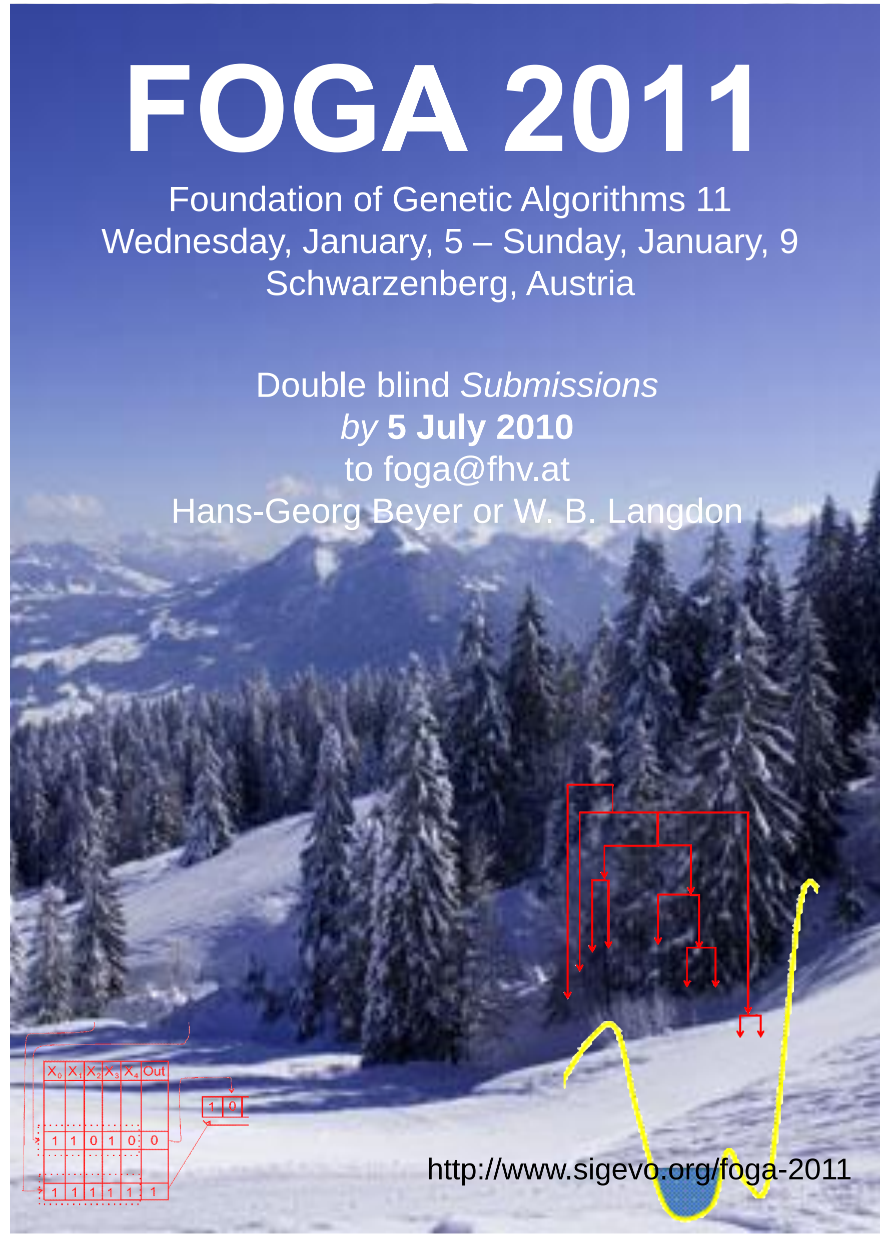
Extended abstracts should be between 10-12 pages long. To submit, please email a compressed postscript or a PDF file to foga@fhv.at no later than Monday, July 5, 2011. In your email, also include the title of the paper, and the name, address and affiliation of all the authors. To ensure the reviews are double-blind authors are asked to remove references to themselves from their paper.

Important Dates

Extended abstracts due	July 5, 2010
Notification to authors	September 13, 2010
Registration and room booking deadline	October 8, 2010
Pre-proceedings camera ready manuscript due	December 6, 2010
FOGA workshop	January 5–9, 2011
Post workshop proceedings	February 21, 2011

Organizers

Prof. Dr. habil. Hans-Georg Beyer	www2.staff.fh-vorarlberg.ac.at/ hgb/
Dr. W. B. Langdon	www.dcs.kcl.ac.uk/staff/W.Langdon/



JavaXCSF: The XCSF Learning Classifier System in Java

Patrick O. Stalph, COBOSLAB, University of Würzburg, Patrick.Stalph@psychologie.uni-wuerzburg.de
Martin V. Butz, COBOSLAB, University of Würzburg, butz@psychologie.uni-wuerzburg.de

Learning Classifier Systems were introduced by John H. Holland and constituted one of the first genetics-based machine learning techniques. The most prominent Learning Classifier System is XCS [3]. XCS can also be used for function approximation, then called XCSF [4]. JavaXCSF is an implementation of the XCSF Learning Classifier System. It is freely available from

www.coboslab.psychologie.uni-wuerzburg.de/code/

Based on previous implementations, the code was extended and heavily restructured, resulting in a flexible framework that helps newcomers to understand the basic structure, but also allows developers to realize their own ideas in the XCSF system. For a detailed documentation, please refer to [2].

XCSF Basics

In order to approximate an unknown function $f(\vec{x}) = \vec{y}$, XCSF evolves a population of classifiers, which structures the input space and learns local (simple) models of the function. As an example, one could approximate a sine function using piecewise linear models. A classifier consists of two important parts: condition and prediction. The condition specifies the locality of the classifier, that is, the subspace where the classifier is located. For example, this could be a rectangular or an ellipsoidal shape. The prediction specifies the function approximation in the subspace, e.g. a linear one.

Given a sample $\vec{x}_t, \vec{y}_t$ at iteration t, the basic workflow of XCSF is as follows.

Matching. Given the input $\vec{x}_t$, XCSF scans its population for matching classifiers.

Prediction. Each of those classifiers predicts the function output $\vec{p}_t$. A weighted sum yields the final prediction.

Update. The predictions are updated using the prediction error $\vec{\varepsilon}_t = \vec{p}_t - \vec{y}_t$.

Evolution. Two classifiers are reproduced – crossover and mutation are applied to the conditions.

Additionally, the present implementation of XCSF allows to register observers (in Java usually called *listeners*), which are informed about changes to the underlying XCSF instance. The observer design pattern allows for a clear separation of the core of XCSF from visualization and logging mechanisms. Figure 1 illustrates the workflow.

Features

Since the last release some new features were added, including new condition and prediction structures and parallelized matching. The source code was restructured to improve readability as well as usability and was profiled to improve its performance. The following paragraphs highlight some of the features.

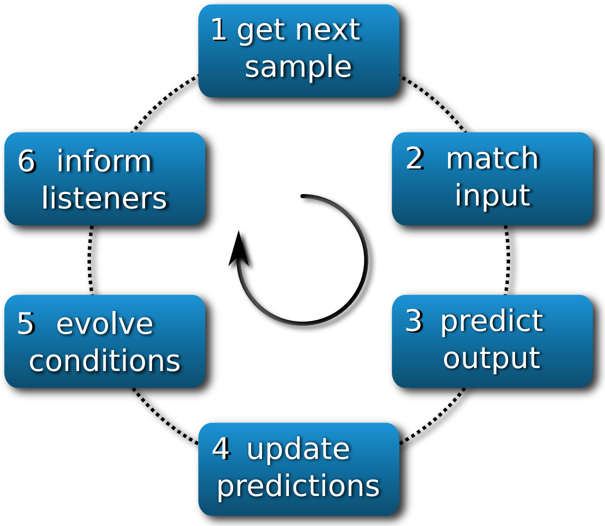


Fig. 1: Six steps define the basic workflow of JavaXCSF. While step 1 depends on the function implementation, steps 2-5 describe XCSF's internal work. Optionally, step 6 updates visualizations or log files.

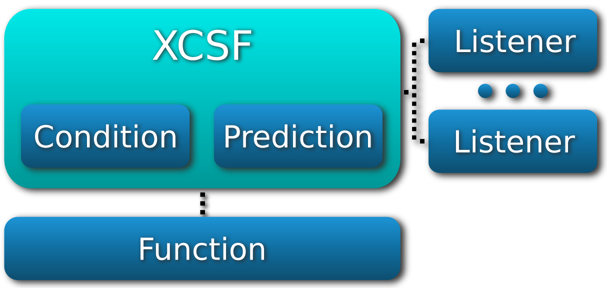


Fig. 2: Important modules in the XCSF framework are accessed via common interfaces, which allows for a fast implementation of new functions, conditions, and predictions. Moreover, a listener interface enables developers to easily plug in desired visualizations or data logging mechanisms.

Object-oriented. Due to the OOP design and intensive documentation, the code is nicely readable and even beginners quickly find their way through the code. Only a few parts of the code do not follow the OOP design for performance reasons.

Portable. Of course, the Java code is portable and can be executed on any operating system that runs a Java Runtime Environment (Version 1.5 or higher).

High Performance. The core of JavaXCSF was optimized using standard profiling methods: Heavily called low-performance methods were identified and rewritten. Visualization plug-ins might slow down the execution, since these optional plug-ins are neither optimized nor integrated into the core of JavaXCSF.

Parallelization. The most computational time is usually spent for the matching process. In order to adapt to current multi-core architectures, the code supports a parallelized matching method besides the usual serial one. A parameterless adaptation scheme assures a speedup equal to or greater than 1 by using serial code until the parallelized version is faster (as measured by computation time).

Extendable. Developers will find it easy to implement new functions, predictors, or condition types. Furthermore, a listener interface realizes the observer design pattern and thus enables to plug in visualizations or data logging mechanisms without modifying XCSF's internal workflow. Figure 2 illustrates the code structure with respect to the important components. Since there is only sparse communication with the function and eventual listeners, additional interfaces can be implemented easily.

Visualization Plug-ins. The code comes with several plug-ins to visualize what is actually happening. For example, the condition structure or the predicted function surface can be visualized during learning. The screenshot in Figure 3 shows three plug-ins.

Online Learning with Parallel Threads. It is possible to run XCSF in one thread, while another thread provides the function samples – real online learning in a synchronized fashion. Basically, this is a parallelized producer-consumer scheme. The example package contains a class that explains how to make use of this feature.

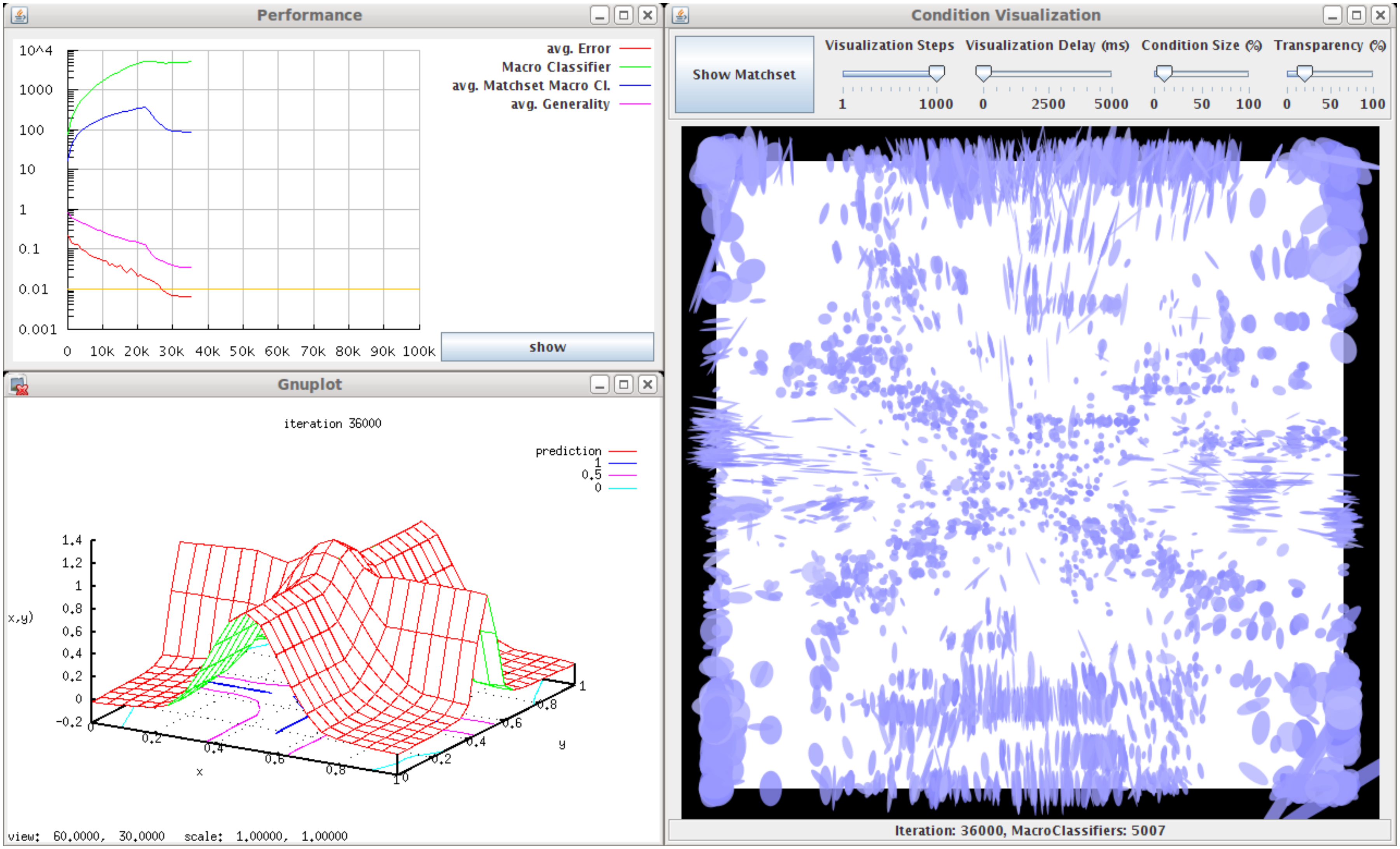


Fig. 3: JavaXCSF comes with several visualization plug-ins – three of them shown on this screenshot. The window at the top left contains graphs of selected elements, e.g. the prediction error. At the bottom left is a plot of the currently predicted function surface, while the corresponding condition structure is shown on the right. Although plug-ins are notified each iteration, the implementation decides, how often the GUI is updated (see *Visualization steps* slider at the top).

Implemented Additions. Besides the basic Learning Classifier System structure for real-valued inputs, the current version also contains condensation as well as compaction, and closest classifier matching mechanisms [1].

Caught Your Attention?

If you want to see XCSF approximate some exemplary functions, just download JavaXCSF and run `java -jar xcsf.jar` in a console. Questions, comments, and suggestions are of course welcome.

Bibliography

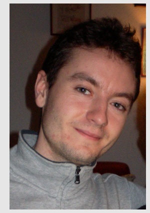
[1] M. V. Butz, P. L. Lanzi, and S. W. Wilson. Function approximation with XCS: Hyperellipsoidal conditions, recursive least squares, and compaction. *IEEE Transactions on Evolutionary Computation*, 12:355–376, 2008.

[2] P. O. Stalph and M. V. Butz. Documentation of JavaXCSF. Technical Report Y2009N001, COBOSLAB, Department of Psychology III, University of Würzburg, Röntgenring 11, 97070 Würzburg, Germany, October 2009.

[3] S. W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

[4] S. W. Wilson. Classifiers that approximate functions. *Natural Computing*, 1:211–234, 2002.

About the authors



Patrick O. Stalph received his diploma in computer science in 2009 from the University of Würzburg in Germany and currently works at the COBOSLAB, supervised by Martin V. Butz. He is interested in genetics-based machine learning, learning classifier systems in particular, as well as in modeling cognitive processes for flexible robot control, learning, and adaptation. Stalph has been working on the XCSF system for nearly three years now.

Email: Patrick.Stalph@psychologie.uni-wuerzburg.de

Dr. Butz received his PhD in computer science at the University of Illinois at Urbana-Champaign in October 2004 under the supervision of David E. Goldberg. His thesis “Rule-based evolutionary online learning systems: Learning Bounds, Classification, and Prediction” puts forward a modular, facet-wise system analysis for Learning Classifier Systems (LCSs) and analyzes and enhances the XCS classifier system. Until September 2007, Butz was working at the University of Würzburg at the Department of Cognitive Psychology III on the interdisciplinary cognitive systems project “MindRACES: From reactive to anticipatory cognitive embodied systems”. In October 2007 he founded his own cognitive systems laboratory: “Cognitive Bodyspaces: Learning and Behavior” (COBOSLAB), funded by the German research foundation under the Emmy Noether framework.

Email: butz@psychologie.uni-wuerzburg.de
COBOSLAB: http://www.coboslab.psychologie.uni-wuerzburg.de/

Dissertation Corner

DEEPSAM A New Hybrid Evolutionary Algorithm for Finding the Lowest Minima of Potential Surfaces: Approach and Application

Doctoral Thesis by Moshe Goldstein

A key factor in the properties of biological molecules is their structure. This research has focused on the problem of *protein structure prediction* due to the critical role of peptides and proteins in the normal biological functionality of cells and organisms. At sufficiently low temperatures, though not always, the structure of the minimal free energy corresponds to the global minimum of its force field (FF) – its Potential Energy Surface (PES), which is a function expressing the potential energy interactions inside the molecule. The problem of predicting the *native* (or *folded*) structure of a polypeptide given its sequence or some unfolded structure *is* the problem of finding that global minimum. This problem is computationally hard because we know that a polypeptide's PES has an estimated number of minima, exponential on its sequence length.

The approach taken in this research aimed at building a purely *ab-initio* global minimum search method. DEEPSAM (Diffusion Equation Evolutionary Programming Simulated Annealing Method), is a new evolutionary algorithm that has been developed hybridizing three well-established optimization methods, resulting in a more powerful tool. DEEPSAM makes effective integrative use of the advantages of Evolutionary Programming (EP), Simulated Annealing (SA) and the Diffusion Equation Method (DEM). Potential function *smoothing*, provided by DEM, contributes to *search space reduction*. Using simultaneously an *ensemble* of candidate solutions (a *population* of conformations), EP contributes to a wide *exploration* of the search space. Four special DEM-SA hybrid mutation operators contribute to long step size and small step size *exploitation* of the neighborhood of conformations in search space; they are dynamically chosen during the computation, allowing run-time *self-adaptability* of the evolutionary algorithm.

It is worth noting that because of the population-oriented approach, this algorithm provides us not only with the deepest minimum found, but also with an ensemble of deep lying minima structures that includes the deepest one. All those minima structures are close to each other energetically and perhaps geometrically as well.

DEEPSAM was tested in three stages:

1. DEEPSAM's "proof-of-concept" test case was a set of cyclic and non-cyclic small peptides. DEEPSAM proved to be a powerful and efficient *structure predictor*. Very good agreement with experimental data was shown. It is worth noting that because of the fact that DEEPSAM is a purely *ab-initio* method, its structure predictions necessarily depend on the validity of the force field in use.
2. DEEPSAM was run upon the protein *Ubiquitin*, charged +13, in the Gas Phase. Using the amber98 force field, an elongated unfolded final structure was found by DEEPSAM and it was in accord with experimental data — according to cross section values calculations, the structures found by DEEPSAM and those found experimentally were indistinguishable.
3. DEEPSAM was run upon the protein *Crambin*, a relatively small protein (46 amino acids long), which contains all the secondary structures present in more complex proteins. The purpose of these runs was to try DEEPSAM as a tool for testing the reliability (and validity) of three well-known forces fields. The results suggest that DEEPSAM may be used as a *force field tester*. Running DEEPSAM upon Crambin, we were able to decide which of the three FFs was the better model to describe the PES of our test protein.

Populations of relatively small size (in most cases, only five conformations) were enough to get these very good results; the parallel design of the algorithm contributed to its time efficiency. These two aspects together, contributed to relatively modest requirements of computer resources.

The success of our algorithm encourages a series of challenging future applications: peptides and proteins in solution and in the Gas Phase, Saccharides, Nucleic Acids, as well as the use of low resolution structures from the Rosetta package as initial structures for DEEPSAM.

Further code development and integration into well-known packages is underway, in cooperation with other groups.

In conclusion, a very promising evolutionary algorithm is now available for predicting the structure of bio-molecules: (a) it can be expected to be a good *structure predictor*, (b) due to its structure prediction capabilities, it has a very *important role to play in the Gas Phase context*, where there is neither X-ray crystallography data nor homological data for bio-molecules, and (c) it can be used as a *force field tester* against structural data.

Moshe Goldstein was born in Montevideo, Uruguay. He received his first degree in Computer Science from the Universidad de la Republica, Uruguay, his MSc in Computer Science from Ben-Gurion University of the Negev, Israel, and very recently his PhD in Computational Chemistry from the Hebrew University of Jerusalem, Israel. Moshe has been a Faculty member of the Computer Science Department of the Jerusalem College of Technology (JCT) during the last fifteen years. Moshe's fields of interest are bio-molecules' structure prediction algorithms, evolutionary computation, principles of programming languages and their application to scientific computation programming, computer science education, database design. Moshe is a Professional Member of the ACM, and a member of SIGEVO; he also is a member of IEEE, IEEE-CS and IEEE-CIS.

Email: goldmosh@g.jct.ac.il
Homepage: http://www.fh.huji.ac.il/˜goldmosh
Dissertation: www.fh.huji.ac.il/˜goldmosh/Thesis.pdf

New Issues of Journals

Evolutionary Computation 17(4) (www)
Special Issue: 12 Years of EC Research in Dortmund

- **Editorial Introduction**, Marc Schoenauer (pdf)
- **Editorial Introduction to the Special Issue: Evolutionary Computing in the Collaborative Research Centre on Computational Intelligence at Technische Universität (TU) Dortmund**, Hans-Paul Schwefel, Uwe Schwiegelshohn (pdf)
- **Analysis of Diversity-Preserving Mechanisms for Global Exploration**, Tobias Friedrich, Pietro S. Oliveto, Dirk Sudholt, Carsten Witt (pdf)
- **S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem**, Nicola Beume (pdf)
- **Statistical Methods for Convergence Detection of Multi-Objective Evolutionary Algorithms**, H. Trautmann, T. Wagner, B. Naujoks, M. Preuss, J. Mehnen (pdf)
- **Hybrid Evolutionary Optimization of Two-Stage Stochastic Integer Programming Problems: An Empirical Investigation**, Thomas Tometzki, Sebastian Engell (pdf)
- **On the Use of Problem-Specific Candidate Generators for the Hybrid Optimization of Multi-Objective Production Engineering Problems**, K. Weinert, A. Zabel, P. Kersting, T. Michelitsch, T. Wagner (pdf)
- **Competitive Coevolutionary Learning of Fuzzy Systems for Job Exchange in Computational Grids**, Alexander Fölling, Christian Grimme, Joachim Lepping, Alexander Papaspyrou, Uwe Schwiegelshohn (pdf)
- **Increasing the Production Accuracy of Profile Bending with Methods of Computational Intelligence**, Alessandro Selvaggio, Uwe Dirksen, A. Erman Tekkaya, Marco Schikorra, Matthias Kleiner (pdf)
- **Multi-Objective Optimization with Controlled Model Assisted Evolution Strategiesno access**, Jan Braun, Johannes Krettek, Frank Hoffmann, Torsten Bertram (pdf)
- **Regular Paper: Dependency Structure Matrix, Genetic Algorithms, and Effective Recombination**, Tian-Li Yu, David E. Goldberg, Kumara Sastry, Claudio F. Lima, Martin Pelikan (pdf)

Evolutionary Intelligence 2(4) (www)

- **Special issue on simulated evolution and learning**, Michael Kirley, Mengjie Zhang and Xiaodong Li, (pdf)
- **Numerical simplification for bloat control and analysis of building blocks in genetic programming**, David Kinzett, Mark Johnston and Mengjie Zhang, (pdf)
- **Improving the performance of evolutionary algorithms in grid-based puzzles resolution**, E. G. Ortiz-García, S. Salcedo-Sanz, Á. M. Pérez-Bellido, L. Carro-Calvo, A. Portilla-Figueras and X. Yao, (pdf)
- **Adaptive ε-Ranking on many-objective problems**, Hernán Aguirre and Kiyoshi Tanaka, (pdf)
- **The effects of time-varying rewards on the evolution of cooperation**, Golriz Rezaei and Michael Kirley (pdf)

Genetic Programming and Evolvable Machines 11(1) (www & blog)

- **Acknowledgment**, Lee Spector (pdf)
- **The influence of mutation on population dynamics in multiobjective genetic programming**, Khaled Badran and Peter I. Rockett (pdf)

- **Automated synthesis of resilient and tamper-evident analog circuits without a single point of failure**, Kyung-Joong Kim, Adrian Wong and Hod Lipson (pdf)
- **GP challenge: evolving energy function for protein structure prediction**, Paweł Widera, Jonathan M. Garibaldi and Natalio Krasnogor (pdf)
- **The identification and exploitation of dormancy in genetic programming**, David Jackson (pdf)
- **Book Review: Michael Affenzeller, Stefan Wagner, Stephan Winkler and Andreas Beham: Genetic algorithms and genetic programming modern concepts and practical applications**, CRC Press, 2009, 379 pp, ISBN-13: 978-1584886297 Gisele L. Pappa (pdf)
- **Book Review: Melanie Mitchell: Complexity a guided tour**, OUP, 2009, Hardback, 368 pp, ISBN13: 9780195124415 Felix Streichert (pdf)

Calls and Calendar

March 2010

4th Workshop on Theory of Randomized Search Heuristics (ThRaSH'2010)

March 24-25, 2010, Paris, France

Homepage: http://trsh2010.gforge.inria.fr/

Registration deadline: March 5, 2010

Following the workshops in Wroclaw, Poland, Dortmund, Germany, and Birmingham, UK, the 4th workshop on Theory of Randomized Search Heuristics (ThRaSH'2010) will take place in Paris on the 24th and 25th of March 2010. The purpose of the workshop is to address questions related to theory of randomized search heuristics such as evolutionary algorithms, ant colony optimization, or simulated annealing for both combinatorial and numerical optimization. The primary focus lies on discussing recent ideas and detecting challenging topics for future work, rather than on the presentation of final results.

Researchers working on theoretical aspects of randomized search heuristics are invited to submit a short abstract (one single page) by email to "thrash2010@lri.fr". No registration fee will be charged but participants are asked to register before the workshop.

April 2010

Evostar 2010 - EuroGP, EvoCOP, EvoBIO and EvoWorkshops

April 7-9, 2010, Istanbul Technical University, Istanbul, Turkey

Homepage: www.evostar.org

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
Workshop Deadline March 25, 2010
Late Breaking Papers Deadline April 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

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
- Notification of paper acceptance: March 15, 2010
- Camera ready submission: May 2, 2010
- IEEE WCCI 2010 Conference: July 18-23, 2010

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

August 2010

IEEE Conference on Computational Intelligence and Games (CIG-2010)

August 18-21, 2010, Copenhagen, Denmark

Homepage: http://game.itu.dk/cig2010

Submission deadline: March 15, 2010

Decision notification: May 15, 2010

Camera-ready submission: June 15, 2010

Conference: August 18-21, 2010

Aim and Scope

Games have proven to be an ideal domain for the study of computational intelligence as not only are they fun to play and interesting to observe, but they provide competitive and dynamic environments that model many real-world problems. Additionally, methods from computational intelligence promise to have a big impact on game technology and development, assisting designers and developers and enabling new types of computer games. The 2010 IEEE Conference on Computational Intelligence and Games brings together leading researchers and practitioners from academia and industry to discuss recent advances and explore future directions in this quickly moving field.

Topics of interest include, but are not limited to:

- Learning in games
- Coevolution in games
- Neural-based approaches for games
- Fuzzy-based approaches for games
- Player/Opponent modeling in games
- CI/AI-based game design
- Multi-agent and multi-strategy learning
- Applications of game theory
- CI for Player Affective Modeling
- Intelligent Interactive Narrative
- Imperfect information and non-deterministic games
- Player satisfaction and experience in games
- Theoretical or empirical analysis of CI techniques for games
- Comparative studies and game-based benchmarking
- Computational and artificial intelligence in:
  - Video games
  - Board and card games
  - Economic or mathematical games
  - Serious games
  - Augmented and mixed-reality games
  - Games for mobile platforms

The conference will consist of a single track of oral presentations, tutorial and workshop/special sessions, and live competitions. The proceedings will be placed in IEEE Xplore, and made freely available on the conference website after the conference.

Conference Committee

General Chairs:	Georgios N. Yannakakis and Julian Togelius
Program Chair:	Michael Mateas, Risto Miikkulainen, and Michael Young
Proceedings Chair:	Pier Luca Lanzi
Competition Chair:	Simon Lucas
Local Chairs:	Anders Drachen, Paolo Burelli, & Tobias Mahlmann

Important Dates

Tutorial proposals:	31st January 2010
Paper submission:	15th March 2010
Decision Notification:	15th May 2010
Camera-ready:	15th Jun 2010
Conference:	18-21 August 2010

For more information please visit: http://game.itu.dk/cig2010/

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

January 2011

FOGA 2011 - Foundations of Genetic Algorithms

January 5-9, 2011, Schwarzenberg, Austria

Homepage: http://www.sigevo.org/foga-2011

Enquiries and Submissions: foga@fhv.at

Deadline Monday July 5, 2010

We invite submissions of extended abstracts for the eleventh Foundations of Genetic Algorithms workshop. FOGA is only held every two years and focuses on theoretical foundations of all flavors of evolutionary computation. It will next be held in the Gasthof Hirschen hotel in Schwarzenberg in Austria from Wednesday, January 5 to Sunday January 9, 2011. Prof. Dr. Karl Sigmund has agreed to deliver a keynote lecture. Attendance is limited to people who submitted papers, or those requesting attendance in advance. Students are particularly encouraged to participate.

Submissions should address theoretical issues in evolutionary computation. Papers that consider foundational issues, place analysis in the wider context of theoretical computer science, or focus on bridging the gap between theory and practice are especially welcome. This does not prevent the acceptance of papers that use an experimental approach, but such work should be directed toward validation of suitable hypotheses concerning foundational matters.

Extended abstracts should be between 10-12 pages long. To submit, please email a compressed postscript or a PDF file to foga@fhv.at no later than Monday, July 5, 2011. In your email, also include the title of the paper, and the name, address and affiliation of all the authors. Submitted papers should use standard spacing and margins, with 11pt or 12pt font for the main text. Authors using LaTeX should either use the standard article style file or the FOGA style file which can be found at the conference web-site. To ensure the reviews are double-blind authors are asked to remove references to themselves from their paper.

Notification will be September 13, 2011 and drafts of the full paper will be needed by December 6, 2010. These drafts will be distributed as part of a preprint to participants at FOGA. Authors of papers presented at the FOGA workshop will be asked to contribute final versions of their papers (based on discussion/feedback at the meeting) as part of the final volume.

Important Dates

Extended abstracts due	July 5, 2010
Notification to authors	September 13, 2010
Registration and room booking deadline	October 8, 2010
Pre-proceedings camera ready manuscript due	December 6, 2010
FOGA workshop	January 5–9, 2011
Post workshop proceedings	February 21, 2011

Organizers

Prof. Dr. habil. Hans-Georg Beyer	www2.staff.fh-vorarlberg.ac.at/ hgb/
Dr. W. B. Langdon	www.dcs.kcl.ac.uk/staff/W.Langdon/

Further Information

Enquiries and submissions: foga@fhv.at

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.