Tutorials at PPSN 2016

Carola Doerr[1(✉)], Nicolas Bredeche[2], Enrique Alba[3], Thomas Bartz-Beielstein[4], Dimo Brockhoff[5], Benjamin Doerr[6], Gusz Eiben[7], Michael G. Epitropakis[8], Carlos M. Fonseca[9], Andreia Guerreiro[9], Evert Haasdijk[7], Jacqueline Heinerman[7], Julien Hubert[7], Per Kristian Lehre[10], Luigi Malagò[11], J.J. Merelo[12], Julian Miller[13], Boris Naujoks[4], Pietro Oliveto[14], Stjepan Picek[15,16], Nelishia Pillay[17], Mike Preuss[18], Patricia Ryser-Welch[13], Giovanni Squillero[19], Jörg Stork[4], Dirk Sudholt[14], Alberto Tonda[20], Darrell Whitley[21], and Martin Zaefferer[4]

[1] CNRS and Sorbonne Universités, Paris, France
Carola.Doerr@mpi-inf.mpg.de
[2] Sorbonne Universités, Paris, France
[3] University of Málaga, Málaga, Spain
[4] TH Köln, Cologne, Germany
[5] Inria Lille - Nord Europe, Villeneuve-d'ascq, France
[6] École Polytechnique, Palaiseau, France
[7] VU Amsterdam, Amsterdam, Netherlands
[8] Lancaster University, Bailrigg, UK
[9] University of Coimbra, Coimbra, Portugal
[10] University of Nottingham, Nottingham, UK
[11] Shinshu University, Matsumoto, Japan
[12] University of Granada, Granada, Spain
[13] University of York, York, UK
[14] University of Sheffield, Sheffield, UK
[15] KU Leuven, Leuven, Belgium
[16] University of Zagreb, Zagreb, Croatia
[17] University of KwaZulu-Natal, Durban, South Africa
[18] University of Dortmund, Dortmund, Germany
[19] Politecnico di Torino, Turin, Italy
[20] INRA, Paris, France
[21] Colorado State University, Fort Collins, USA

Abstract. PPSN 2016 hosts a total number of 16 tutorials covering a broad range of current research in evolutionary computation. The tutorials range from introductory to advanced and specialized but can all be attended without prior requirements. All PPSN attendees are cordially invited to take this opportunity to learn about ongoing research activities in our field!

1 Chairs' Welcome

Tutorials offer an efficient and interactive way of learning about ongoing research activities. While introductory tutorials are particularly targeted at researchers

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 1012–1022, 2016.
DOI: 10.1007/978-3-319-45823-6_95

who have just recently entered (or are about to enter) the multifaceted research field of evolutionary computation, more specialized tutorials address both junior and senior researchers intending to intensify or refresh their knowledge about various topics of interest.

In response to our call for tutorials we have received a large number of high-quality tutorial proposals out of which 16 have been selected for presentation at the conference. These 16 tutorials will be presented in two days at PPSN 2016, on September 17 and 18, which are exclusively reserved for tutorial and workshop presentations.

The topics of the tutorials cover introductions to evolutionary computation in cryptography, multi-modal optimization, and hyper-heuristics.

More specialized tutorials discuss gray-box optimization, graph-based and cartesian genetic programming, intelligent systems for smart cities, and the importance of diversity in evolutionary optimization.

A classic in the tutorial landscape is the introduction to evolutionary multi-objective optimization (EMO), a topic also addressed in the tutorials on using the attainment function as a tool for the performance evaluation of EMO algorithms.

Those researchers wishing to learn more on the role of theory in our field should not miss the tutorial on theory of evolutionary computation. This tutorial is followed up by a basic introduction to runtime analysis of evolutionary algorithms (EAs) and one on the theory of parallel EAs. A forth theory-flavored tutorial aims at bridging the gap between the optimization over manifolds and evolutionary computation.

In addition, a hands-on guide to experiment with real hardware is proposed for evolutionary robotics, it is discussed how to efficiently implement EAs in the cloud, and how to save time and cost through meta-model assisted optimization. We invite all PPSN participants to explore the wide range of topics discussed in the selected tutorials and wish you an enjoyable conference!
Nicolas Bredeche and Carola Doerr
PPSN 2016 Tutorial Chairs

2 Abstracts of the Tutorials

2.1 A Bridge Between Optimization over Manifolds and Evolutionary Computation

Tutorial Speaker: Luigi Malagò, *Shinshu University (Japan)*

Tutorial Abstract: The aim of this tutorial is to explore the promising connection between the well-consolidated field of optimization over manifolds and evolutionary computation. In mathematics, optimization over manifolds deals with the design and analysis of algorithms for the optimization over search spaces with admit a non-Euclidean geometry. One of the simplest examples is probably the sphere, where the shortest path between two points is given by a curve, and not a straight line. Manifolds may appear in evolutionary computation in at least two contexts. The simplest one is the case when an evolutionary algorithm is employed to optimize a fitness function defined over a manifold, such as in the case of the sphere, the cone of positive-definite matrices, the set of rotation matrices, and many others. The second one is more subtle, and is related to the stochastic relaxation of a fitness function. A common approach in model-based evolutionary computation is to search for the optimum of a function by sampling populations from a sequence of probability distributions. For instance, this is the case of evolutionary strategies, probabilistic model-building genetic algorithms, estimation of distribution algorithms and similar techniques, both in the continuous and in the discrete domain. A strictly related paradigm which can be used to describe the behavior of model-based search algorithms is that of stochastic relaxation, i.e., the optimization of the expected value of the original fitness function with respect to a probability distribution in a statistical model. From this perspective a model-based algorithm is solving a problem which is strictly related to the optimization of the stochastic relaxation over a statistical model. Notably, statistical models are well-known examples of manifolds, where the Fisher information plays the role of metric tensor. For this reason, it becomes of great interest to compare the standard techniques in the field of optimization over manifolds, with the mechanisms implemented by model-based algorithm in evolutionary computation. The tutorial will consist of two parts. In the first one, a unifying framework for the description of model-based algorithms will be introduced and some standard well-known algorithms will be presented from the perspective of the optimization over manifold. Particular attention will be devoted to first-order methods based on the Riemannian gradient over a manifold, which in the case of a statistical model is known as the natural gradient. In the second part, we will discuss how evolutionary algorithms can be adapted to solve optimization problems defined over manifold, which constitutes a novel and promising area of research in evolutionary computation.

2.2 Advances on Multi-modal Optimization

Tutorial Speaker: Mike Preuss, *University of Dortmund (Germany)*, and Michael G. Epitropakis, *Lancaster University (UK)*

Tutorial Abstract: Multimodal optimization is currently getting established as a research direction that collects approaches from various domains of operational research and evolutionary computation that strive for delivering multiple very good solutions at once. We discuss several scenarios and list currently employed and potentially available performance measures. Furthermore, many state-of-the-art as well as older methods are compared and put into a rough taxonomy. We also discuss recent relevant competitions and their results and outline the possible future developments in this area.

2.3 The Attainment Function Approach to Performance Evaluation in Evolutionary Multiobjective Optimization

Tutorial Speaker: Carlos M. Fonseca and Andreia P. Guerreiro, *University of Coimbra (Portugal)*

Tutorial Abstract: The development of improved optimization algorithms and their adoption by end users depend on the ability to evaluate their performance on the problem classes of interest. In the absence of theoretical guarantees, performance must be evaluated experimentally while taking into account both the experimental conditions and the nature of the data collected.

Evolutionary approaches to multiobjective optimization typically produce discrete Pareto-optimal front approximations in the form of sets of mutually non-dominated points in objective space. Since evolutionary algorithms are stochastic, such non-dominated point sets are random, and vary according to some probability distribution.

In contrast to quality indicators, which map non-dominated point sets to real values, and side-step the set nature of the data, the attainment-function approach addresses the non-dominated point set distribution directly. Distributional aspects such as location, variability, and dependence, can be estimated from the raw non-dominated point set data.

This tutorial will focus on the attainment function as a tool for the evaluation of the performance of evolutionary multiobjective optimization (EMO) algorithms. In addition to the theoretical foundations of the methodology, computational and visualization issues will be discussed. The application of the methodology will be demonstrated by interactively exploring example data sets with freely available software tools. To conclude, a selection of open questions and directions for further work will be identified.

2.4 Evolutionary Algorithms and Hyper-heuristics

Tutorial Speaker: Nelishia Pillay, *University of KwaZulu-Natal (South Africa)*

Tutorial Abstract: Hyper-heuristics is a rapidly developing domain which has proven to be effective at providing generalized solutions to problems and across problem domains. Evolutionary algorithms have played a pivotal role in the advancement of hyper-heuristics, especially generation hyper-heuristics. Evolutionary algorithm hyper-heuristics have been successful applied to solving problems in various domains including packing problems, educational timetabling, vehicle routing, permutation flowshop and financial forecasting amongst others. The aim of the tutorial is to firstly provide an introduction to evolutionary algorithm hyper-heuristics for researchers interested in working in this domain. An overview of hyper-heuristics will be provided. The tutorial will examine each of the four categories of hyper-heuristics, namely, selection constructive, selection perturbative, generation constructive and generation perturbative, showing how evolutionary algorithms can be used for each type of hyper-heuristic. A case study will be presented for each type of hyper-heuristic to provide researchers with a foundation to start their own research in this area. Challenges in the implementation of evolutionary algorithm hyper-heuristics will be highlighted. An emerging research direction is using hyper-heuristics for the automated design of computational intelligence techniques. The tutorial will look at the synergistic relationship between evolutionary algorithms and hyper-heuristics in this area. The use of hyper-heuristics for the automated design of evolutionary algorithms will be examined as well as the application of evolutionary algorithm hyper-heuristics for the design of computational intelligence techniques. The tutorial will end with a discussion session on future directions in evolutionary algorithms and hyper-heuristics.

2.5 Evolutionary Computation in Cryptography

Tutorial Speaker: Stjepan Picek, *KU Leuven (Belgium) and University of Zagreb (Croatia)*

Tutorial Abstract: Evolutionary Computation (EC) has been used with great success on various real-world problems. One domain abundant with numerous difficult problems is cryptology. Cryptology can be divided into cryptography and cryptanalysis where although not always in an obvious way, EC can be applied to problems from both domains. This tutorial will first give a brief introduction to cryptology intended for general audience. Afterwards, we concentrate on several topics from cryptography that are successfully tackled up to now with EC and discuss why those topics are suitable to apply EC. However, care must be taken since there exists a number of problems that seem to be impossible to solve with EC and one needs to realize the limitations of the heuristics. We will discuss the choice of appropriate EC techniques (GA, GP, CGP, ES, multi-objective optimization) for various problems and evaluate on the importance of that choice. Furthermore, we will discuss the gap between the cryptographic community and EC community and what does that mean for the results. By doing that, we give a special emphasis on the perspective that cryptography presents a source of benchmark problems for the EC community.

This tutorial will also present some live demos of EC in action when dealing with cryptographic problems.

2.6 Evolutionary Multiobjective Optimization

Tutorial Speaker: Dimo Brockhoff, *Inria Lille - Nord Europe (France)*

Tutorial Abstract: Many optimization problems are multiobjective, i.e., multiple, conflicting criteria need to be considered simultaneously. Due to conflicts between the objectives, usually no single optimum solution exists. Instead, a set of so-called Pareto-optimal solutions, for which no other solution has better function values in all objectives, does emerge.

In practice, Evolutionary Multiobjective Optimization (EMO) algorithms are widely used for solving multiobjective optimization problems. As stochastic blackbox optimizers, EMO approaches cope with nonlinear, nondifferentiable, or noisy objective functions. By inherently working on sets of solutions, they allow the Pareto-optimal set to be approximated in one algorithm run—opposed to classical techniques for multicriteria decision making (MCDM), which aim for single solutions.

Defining problems in a multiobjective way has two further advantages:

- The set of Pareto-optimal solutions may reveal shared design principles (innovization)
- Singleobjective problems may become easier to solve if auxiliary objectives are added (multiobjectivization).

Within this tutorial, we comprehensively introduce the field of EMO and present selected research results in more detail. More specifically, we

- explain the basic principles of EMO algorithms in comparison to classical approaches,
- show a few practical examples motivating the use of EMO, and
- present a general overview of state-of-the-art algorithms and selected recent research results.

2.7 Evolutionary Robotics—A Practical Guide to Experiment with Real Hardware

Tutorial Speaker: Jacqueline Heinerman and Gusz Eiben and Evert Haasdijk and Julien Hubert, *VU Amsterdam (Netherlands)*

Tutorial Abstract: Evolutionary robotics aims to evolve the controllers, the morphologies, or both, for real and/or simulated autonomous robots. Most research in evolutionary robotics is partly or completely carried in simulation. Although simulation has advantages, e.g., it is cheaper and it can be faster, it suffers from the notorious reality gap. Recently, affordable and reliable robots became commercially available. Hence, setting up a population of real robots is within reach for a large group of research groups today. This tutorial focuses on the know-how required to utilise such a population for running evolutionary experiments. To this end we use Thymio II robots with Raspberry Pi extensions (including a camera). The tutorial explains and demonstrates the work-flow from beginning to end, by going through a case study of a group of Thymio II robots evolving their neural network controllers to learn collecting objects on-the-fly. Besides the methodology and lessons learned, we spend time on how to code.

2.8 Graph-Based and Cartesian Genetic Programming

Tutorial Speaker: Julian Miller and Patricia Ryser-Welch, *University of York (UK)*

Tutorial Abstract: Genetic Programming is often associated with a tree representation for encoding expressions and algorithms. However, graphs are also very useful and flexible program representations which can be applied to many domains (e.g. electronic circuits, neural networks, algorithms).

Over the years a variety of representations of graphs have been explored such as: Parallel Distributed Genetic Programming (PDGP) , Linear-Graph Genetic Programming, Enzyme Genetic Programming, Graph Structured Program Evolution (GRAPE) and Cartesian Genetic Programming (CGP).

Cartesian Genetic Programming (CGP) is probably the best known form of graph-based Genetic Programming. It was developed by Julian Miller in 1999–2000. In its classic form, it uses a very simple integer address-based genetic representation of a program in the form of a directed graph. CGP has been adopted by a large number of researchers in many domains.

In a number of studies, CGP has been shown to be comparatively efficient to other GP techniques. It is also very simple to program. Since its original formulation, the classical form of CGP has also undergone a number of developments which have made it more useful, efficient and flexible in various ways. These include the addition of automatically defined functions (modular CGP), self-modification operators (self-modifying CGP), the encoding of artificial neural networks (GCPANNs) and evolving iterative programs (iterative CGP).

2.9 Gray Box Optimization in Theory and Practice

Tutorial Speaker: Darrell Whitley, *Colorado State University (USA)*

Tutorial Abstract: This tutorial will cover Gray Box Complexity and Gray Box Optimization for k-bounded pseudo-Boolean optimization. These problems can also be referred to at Mk Landscapes, and included problems such as MAX-kSAT, spin glass problems and NK Landscapes. Mk Landscape problems are a linear combination of M subfunctions, where each subfunction accepts at most k variables. Under Gray Box optimization, the optimizer is given access to the set of M subfunctions. If the set of subfunctions is k-bounded and separable, the Gray Box optimizer is guaranteed to return the global optimum with 1 evaluation. If a problem is not deceptive, the Gray Box optimizer also returns the global optimum after 1 evaluation. This means that simple test problems from ONEMAX to "Trap Functions" are solved in 1 evaluation in $O(n)$ time under Gray Box Optimization. If a tree decomposition exists with a fixed bounded tree width, then the problem can be solved using dynamic programming in $O(n)$ time. If the tree decomposition is bounded by $\lg(n)$, then the problem can be solved by dynamic programming in $O(n^2)$ time. Even for those problems that are not trivially solved, Gray Box optimization also makes it possible to exactly compute Hamming distance 1 improving moves in constant time. Thus, neither mutation nor enumeration of the Hamming neighborhood are necessary. Under many conditions it is possible to calculate the location of improving moves in a Hamming distance radius r neighborhood, thus selecting improving moves several moves ahead. This also can be done in constant time. There also exists deterministic forms of recombination that provably return the best possible offspring from a reachable set of offspring. Partition Crossover relies on localized problem decomposition, and is invariant to the order of the bits in the representation. The methods identify partitions of nonlinear interaction between variables. Variables within a partition must be inherited together. However, bits in different partitions can be linearly recombined. Given p partitions, recombination can be done in $O(n)$ time such that crossover returns the best solutions out of 2^p offspring. The offspring can also be proven to be locally optimal in the largest hyperplane subspace in which the two parents reside. Thus, Partition Crossover is capable of directly moving from known local optima to new, high quality local optima in $O(n)$ time. These innovations will fundamentally change both Local Search and Evolutionary Algorithms. Empirical results show that combining smart local search with Partition Crossover results in search algorithms that are capable of finding globally optimal solutions for nonlinear problems with a million variables in less than 1 min.

2.10 Implementing Evolutionary Algorithms in the Cloud

Tutorial Speaker: JJ Merelo, *University of Granada (Spain)*

Tutorial Abstract: Creating experiments that can be easily reproduced and converted in a straightforward way into a report involves knowing a series of techniques that are of widespread use in the open source and commercial software communities. This tutorial will introduce this techniques, including an introduction to cloud computing and DevOps for evolutionary algorithm practitioners, with reference to the tools and platforms that can make development of new algorithms and problem solutions fast and reproducible.

2.11 Intelligent Systems for Smart Cities

Tutorial Speaker: Enrique Alba, *University of Málaga (Spain)*

Tutorial Abstract: The concept of Smart Cities can be understood as a holistic approach to improve the level of development and management of the city in a broad range of services by using information and communication technologies.

It is common to recognize six axes of work in them: (i) Smart Economy, (ii) Smart People, (iii) Smart Governance, (iv) Smart Mobility, (v) Smart Environment, and (vi) Smart Living. In this tutorial we first focus on a capital issue: smart mobility. European citizens and economic actors need a transport system which provides them with seamless, high-quality door-to-door mobility. At the same time, the adverse effects of transport on the climate, the environment and human health need to be reduced. We will show many new systems based in the use of bio-inspired techniques to ease the road traffic flow in the city, as well as allowing a customized smooth experience for travelers (private and public transport).

This tutorial will then discuss on potential applications of intelligent systems for energy (like adaptive lighting in streets), environmental applications (like mobile sensors for air pollution), smart building (intelligent design), and several other applications linked to smart living, tourism, and smart municipal governance.

2.12 Meta-Model Assisted (Evolutionary) Optimization

Tutorial Speaker: Boris Naujoks and Jörg Stork and Martin Zaefferer and Thomas Bartz-Beielstein, *TH Köln (Germany)*

Tutorial Abstract: Meta-model assisted optimization is a well-recognized research area. When the evaluation of an objective function is expensive, meta-model assisted optimization yields huge improvements in optimization time or cost in a large number of different scenarios. Hence, it is extremely useful for numerous real-world applications. These include, but are not limited to, the optimization of designs like airfoils or ship propulsion systems, chemical processes, biogas plants, composite structures, and electromagnetic circuit design.

This tutorial is largely focused on evolutionary optimization assisted by meta-models, and has the following aims: Firstly, we will provide a detailed understanding of the established concepts and distinguished methods in meta-model assisted optimization. Therefore, we will present an overview of current research and open issues in this field. Moreover, we aim for a practical approach. The tutorial should enable the participants to apply up-to-date meta-modelling approaches to actual problems at hand. Afterwards, we will discuss typical problems and their solutions with the participants. Finally, the tutorial offers new perspectives by taking a look into areas where links to meta-modelling concepts have been established more recently, e.g., the application of meta-models in multi-objective optimization or in combinatorial search spaces.

2.13 Promoting Diversity in Evolutionary Optimization: Why and How

Tutorial Speaker: Giovanni Squillero, *Politecnico di Torino (Italy)*, and Alberto Tonda, *INRA (France)*

Tutorial Abstract: Divergence of character is a cornerstone of natural evolution. On the contrary, evolutionary optimization processes are plagued by an endemic lack of diversity: all candidate solutions eventually crowd the very same areas in the search space. Such a "lack of speciation" has been pointed out in the seminal work of Holland in 1975, and nowadays is well known among scholars. It has different effects on the different search algorithms, but almost all are quite deleterious. The problem is usually labeled with the oxymoron "premature convergence", that is, the tendency of an algorithm to convergence toward a point where it was not supposed to converge to in the first place. Scientific literature contains several efficient diversity-preservation methodologies that ranged from general techniques to problem-dependent heuristics. However, the fragmentation of the field and the difference in terminology led to a general dispersion of this important corpus of knowledge in many small, hard-to-track research lines.

Upon completion of this tutorial, attendees will understand the root causes and dangers of “premature convergence”. They will know the main research lines in the area of “diversity promotion”. They will be able to choose an effective solution from the literature, or design a new one more tailored to their specific needs.

2.14 Runtime Analysis of Evolutionary Algorithms: Basic Introduction

Tutorial Speaker: Per Kristian Lehre, *University of Nottingham (UK)*, and Pietro S. Oliveto, *University of Sheffield (UK)*

Tutorial Abstract: Evolutionary algorithm theory has studied the time complexity of evolutionary algorithms for more than 20 years. This tutorial presents the foundations of this field. We introduce the most important notions and definitions used in the field and consider different evolutionary algorithms on a number of well-known and important example problems. Through a careful and thorough introduction of important analytical tools and methods, including fitness- and level-based analysis, typical events and runs, and drift analysis. By the end of the tutorial the attendees will be able to apply these techniques to derive relevant runtime results for non-trivial evolutionary algorithms.

In addition to custom-tailored methods for the analysis of evolutionary algorithms we also introduce the relevant tools and notions from probability theory in an accessible form. This makes the tutorial appropriate for everyone with an interest in the theory of evolutionary algorithms without the need to have prior knowledge of probability theory and analysis of randomised algorithms.

Variants of this tutorial have been presented at GECCO 2013–2015, attracting well over 50 participants each time. The tutorial will be based on the ’Theoretical analysis of stochastic search heuristics’ chapter of the forthcoming Springer Handbook of Heuristics.

2.15 Theory of Evolutionary Computation

Tutorial Speaker: Benjamin Doerr, *École Polytechnique (France)*

Tutorial Abstract: Theoretical research has always accompanied the development and analysis of evolutionary algorithms, both by explaining observed phenomena in a very rigorous manner and by creating new ideas. Since the methodology of theory research is very different from experimental or applied research, non-theory researcher occasionally find it hard to understand and profit from theoretical research. Overcoming this gap in our research field is the target of this tutorial. Independent of particular theoretical subdisciplines or methods like runtime analysis or landscape theory, we aim at making theory accessible to researchers having little exposure to theory research previously. In particular,

- we describe what theory research in EC is, what it aims at, and showcase some of key findings of the last 15 years,
- we discuss the particular strengths and limitations of theory research,
- we show how to read, understand, interpret, and profit from theory results.

2.16 Theory of Parallel Evolutionary Algorithms

Tutorial Speaker: Dirk Sudholt, *University of Sheffield (UK)*

Tutorial Abstract: Evolutionary algorithms (EAs) have given rise to many parallel variants, fuelled by the rapidly increasing number of CPU cores and the ready availability of computation power through GPUs and cloud computing. A very popular approach is to parallelize evolution in island models, or coarse-grained EAs, by evolving different populations on different processors. These populations run independently most of the time, but they periodically communicate genetic information to coordinate search. Many applications have shown that island models can speed up computation time significantly, and that parallel populations can further increase solution diversity. However, there is little understanding of when and why island models perform well, and what impact fundamental parameters have on performance.

This tutorial will give an overview of recent theoretical results on the runtime of parallel evolutionary algorithms. These results give insight into the fundamental working principles of parallel EAs, assess the impact of parameters and design choices on performance, and contribute to the design of more effective parallel EAs.