# EuroGP programme

# Wednesday 30 March 2016

## 11:10-13:00 Session 1

### chairs: Malcolm Heywood & James McDermott

Full presentations: *Models of Evolution*
(***format: 18 minutes per paper + 5 min. questions***)

***Genetic Programming based Hyper-heuristics for Dynamic Job Shop Scheduling: Cooperative Coevolutionary Approaches***
John Park, Yi Mei, Su Nguyen, Gang Chen, Mengjie Zhang
Job shop scheduling (JSS) problems are optimisation problems that have been studied extensively due to their computational complexity and application in manufacturing systems. This paper focuses on a dynamic JSS problem to minimise the total weighted tardiness. In dynamic JSS, jobs' attributes are only revealed after they arrive at the shop floor. Dispatching rule heuristics are prominent approaches to dynamic JSS problems, and Genetic Programming based Hyper-heuristic (GP-HH) approaches have been proposed to automatically generate effective dispatching rules for dynamic JSS problems. Research on static JSS problems shows that high quality ensembles of dispatching rules can be evolved by a GP-HH that uses cooperative coevolution. Therefore, we compare two coevolutionary GP approaches to evolve ensembles of dispatching rules for dynamic JSS problem. First, we adapt the Multilevel Genetic Programming (MLGP) approach, which has never been applied to JSS problems. Second, we extend an existing approach for static JSS problem, called Ensemble Genetic Programming for Job Shop Scheduling (EGP-JSS), by adding “less-myopic” terminals that take job and machine attributes outside of the scope of the attributes commonly used in the literature. The results show that MLGP for JSS evolves ensembles that are significantly better than single “less-myopic” rules evolved using GP with only little difference in computation time. In addition, the rules evolved using EGP-JSS perform better than the MLGP-JSS rules, but MLGP-JSS evolves rules significantly faster than EGP-JSS.

***A Genetic Programming Approach for the Traffic Signal Control Problem with Epigenetic Modifications***
Esteban Ricalde, Wolfgang Banzhaf
This paper presents a proof-of-concept for an Epigenetics-based modification of Genetic Programming (GP). The modification is tested with a traffic signal control problem under dynamic traffic conditions. We describe the new algorithm and show first results. Experiments reveal that GP benefits from properties such as phenotype differentiation, memory consolidation within generations and environmentally-induced change in behavior provided by the epigenetic mechanism. The method can be extended to other dynamic environments.

***Plastic Fitness Predictors Coevolved with Cartesian Programs***
Michal Wiglasz, Michaela Drahosova
Coevolution of fitness predictors, which are a small sample of all training data for a particular task, was successfully used to reduce the computational cost of the design performed by cartesian genetic programming. However, it is necessary to specify the most advantageous number of fitness cases in predictors, which differs from task to task. This paper introduces a new type of directly encoded fitness predictors inspired by the principles of phenotypic plasticity. The size of the coevolved fitness predictor is adapted in response to the learning phase that the program evolution goes through. It is shown in 5 symbolic regression tasks that the proposed algorithm is able to adapt the number of fitness cases in predictors in response to the solved task and the program evolution flow.

Short presentations: *Applications*
(***format: 5 minute paper pitch****)

***Search-Based SQL Injection Attacks Testing using Genetic Programming***
Benjamin Aziz, Mohamed Bader, Cerana Hippolyte
Software testing is a key phase of many development methodologies as it provides a natural opportunity for integrating security early in the software development lifecycle. However despite the known importance of software testing, this phase is often overlooked as it is quite difficult and labour-intensive to obtain test datasets to effectively test an application. This lack of adequate automatic software testing renders software applications vulnerable to malicious attacks after they are deployed as detected software vulnerabilities start having an impact during the production phase. Among such attacks are SQL injection attacks. Exploitation of SQL injection vulnerabilities by malicious programs could result in severe consequences such as breaches of confidentiality and false authentication. It is therefore important that an application is adequately tested with a high volume of test data to ensure that it can withstand such attacks before it is deployed into the production phase. We present a search-based software testing technique to detect SQL injection vulnerabilities in software applications. This approach uses genetic programming as a means of generating our test datasets, which are then used to test applications for SQL injection-based vulnerabilities.

***Towards Automated Strategies in Satisfiability Modulo Theory***
Nicolás Gálvez Ramírez, Youssef Hamadi, Eric Monfroy, Frédéric Saubion
SMT solvers include many heuristic components in order to ease the theorem proving process for different logics and problems. Handling these heuristics is a non-trivial task requiring specific knowledge of many theories that even a SMT solver developer may be unaware of. This is the first barrier to break in order to allow end-users to control heuristics aspects of any SMT solver and to successfully build a strategy for their own purposes. We present a first attempt for generating an automatic selection of heuristics in order to improve SMT solver efficiency and to allow end-users to take better advantage of solvers when unknown problems are faced. Evidence of improvement is shown and the basis for future works with evolutionary and/or learning-based algorithms are raised.

***Patterns for Constructing Mutation Operators: Limiting the Search Space in a Software Engineering Application***
Thomas Kühne, Heiko Hamann, Svetlana Arifulina, Gregor Engels
We apply methods of genetic programming to a general problem from software engineering, namely example-based generation of specifications. In particular, we focus on model transformation by example. The definition and implementation of model transformations is a task frequently carried out by domain experts, hence, a (semi-)automatic approach is desirable. This application is challenging because the underlying search space has rich semantics, is high-dimensional, and unstructured. Hence, a computationally brute-force approach would be unscalable and potentially infeasible. To address that problem, we develop a sophisticated approach of designing complex mutation operators. We define “patterns” for constructing mutation operators and report a successful case study. Furthermore, the code of the evolved model transformation is required to have high maintainability and extensibility, that is, the code should be easily readable by domain experts. We report an evaluation of this approach in a software engineering case study.

***Iterative Cartesian Genetic Programming: Creating general algorithms for solving Travelling Salesman Problems***
Patricia Ryser-Welch, Julian F. Miller, Jerry Swan, Martin A. Trefzer
Evolutionary algorithms have been widely used to optimise or design search algorithms, however, very few have considered evolving iterative algorithms. In this paper, we introduce a novel extension to Cartesian Genetic Programming that allows it to encode iterative algorithms. We apply this technique to the Traveling Salesman Problem to produce human-readable solvers which can be then be independently implemented. Our experimental results demonstrate that the evolved solvers scale well to much larger TSP instances than those used for training.

* Short presentations must have a corresponding poster in the poster session. Full presentations are encouraged to do likewise.

## 14:00-15:50 Session 2

### chair: Jerry Swan

Full presentations: *Classification*
***(format: 18 minutes per paper + 5 min. questions)***

***One-class Classification for Anomaly Detection with Kernel Density Estimation and Genetic Programming***
Van Loi Cao, Miguel Nicolau, James McDermott
A novel approach is proposed for fast anomaly detection by one-class classification. Standard kernel density estimation is first used to obtain an estimate of the input probability density function, based on the one-class input data. This can be used for anomaly detection: query points are classed as anomalies if their density is below some threshold. The disadvantage is that kernel density estimation is lazy, that is the bulk of the computation is performed at query time. For large datasets it can be slow. Therefore it is proposed to approximate the density function using genetic programming symbolic regression, before imposing the threshold. The runtime of the resulting genetic programming trees does not depend on the size of the training data. The method is tested on datasets including in the domain of network security. Results show that the genetic programming approximation is generally very good, and hence classification accuracy approaches or equals that when using kernel density estimation to carry out one-class classification directly. Results are also generally superior to another standard approach, one-class support vector machines.

***On the Impact of Class Imbalance in GP Streaming Classification with Label Budgets***
Sara Khanchi, Malcolm I. Heywood, Nur Zincir-Heywood
Streaming data scenarios introduce a set of requirements that do not exist under supervised learning paradigms typically employed for classification. Specific examples include, anytime operation, non-stationary processes, and limited label budgets. From the perspective of class imbalance, this implies that it is not even possible to guarantee that all classes are present in the samples of data used to construct a model. Moreover, when decisions are made regarding what subset of data to sample, no label information is available. Only after sampling is label information provided. This represents a more challenging task than encountered under non-streaming (offline) scenarios because the training partition contains label information. In this work, we investigate the utility of different protocols for sampling from the stream under the above constraints. Adopting a uniform sampling protocol was previously shown to be reasonably effective under both evolutionary and non-evolutionary streaming classifiers. In this work, we introduce a scheme for using the current “champion” classifier to bias the sampling of training instances during the course of the stream. The resulting streaming framework for genetic programming is more effective at sampling minor classes and therefore reacting to changes in the underlying process responsible for generating the data stream.

***Genetic Programming for Region Detection, Feature Extraction, Feature Construction and Classification in Image Data***
Andrew Lensen, Harith Al-Sahaf, Mengjie Zhang, Bing Xue
Image analysis is a key area in the computer vision domain that has many applications. Genetic Programming (GP) has been successfully applied to this area extensively, with promising results. High-level features extracted from methods such as Speeded Up Robust Features (SURF) and Histogram of Oriented Gradients (HoG) are commonly used for object detection with machine learning techniques. However, GP techniques are not often used with these methods, despite being applied extensively to image analysis problems. Combining the training process of GP with the powerful features extracted by SURF or HoG has the potential to improve the performance by generating high-level, domain-tailored features. This paper proposes a new GP method that automatically detects different regions of an image, extracts HoG features from those regions, and simultaneously evolves a classifier for image classification. By extending an existing GP region selection approach to incorporate the HoG algorithm, we present a novel way of using high-level features with GP for image classification. The ability of GP to explore a large search space in an efficient manner allows all stages of the new method to be optimised simultaneously,

unlike in existing approaches. The new approach is applied across a range of datasets, with promising results when compared to a variety of well-known machine learning techniques. Some high-performing GP individuals are analysed to give insight into how GP can effectively be used with high-level features for image classification.

***A Genetic Programming-based Imputation Method for Classification with Missing Data***

Cao Truong Tran, Mengjie Zhang, Peter Andreae

Many industrial and real-world datasets suffer from an unavoidable problem of missing values. The ability to deal with missing values is an essential requirement for classification because inadequate treatment of missing values may lead to large errors on classification. The problem of missing data has been addressed extensively in the statistics literature, and also, but to a lesser extent in the classification literature. One of the most popular approaches to deal with missing data is to use imputation methods to fill missing values with plausible values. Some powerful imputation methods such as regression-based imputations in MICE are often suitable for batch imputation tasks. However, they are often expensive to impute missing values for every single incomplete instance in the unseen set for classification. This paper proposes a genetic programming-based imputation (GPI) method for classification with missing data that uses genetic programming as a regression method to impute missing values. The experiments on six benchmark datasets and five popular classifiers compare GPI with five other popular and advanced regression-based imputation methods in MICE on two measures: classification accuracy and computation time. The results showed that, in most cases, GPI achieves classification accuracy at least as good as the other imputation methods, and sometimes significantly better. However, using GPI to impute missing values for every single incomplete instance is dramatically faster than the other imputation methods.

## Short presentations: *Foundations*

***(format: 5 minute paper pitch*)***

***Modelling Evolvability in Genetic Programming***

Benjamin Fowler, Wolfgang Banzhaf

We develop a tree-based genetic programming system capable of modelling evolvability during evolution through machine learning algorithms, and exploiting those models to increase the efficiency and final fitness. Existing methods of determining evolvability require too much computational time to be effective in any practical sense. By being able to model evolvability instead, computational time may be reduced. This will be done first by demonstrating the effectiveness of modelling these properties *a priori*, before expanding the system to show its effectiveness as evolution occurs.

***Grammar Design for Derivation Tree Based Genetic Programming Systems***

Stefan Forstenlechner, Miguel Nicolau, David Fagan, Michael O'Neill

Grammar-based genetic programming systems have gained interest in recent decades and are widely used nowadays. Although researchers normally present the grammar used to solve a certain problem, they seldom write about processes used to construct the grammar. This paper sheds some light on how to design a grammar that not only covers the search space, but also supports the search process in finding good solutions. The focus lies on context free grammar guided systems using derivation tree crossover and mutation, in contrast to linearised grammar based systems. Several grammars are presented encompassing the search space of sorting networks and show concepts which apply to general grammar design. An analysis of the search operators on different grammar is undertaken and performance examined on the sorting network problem. The results show that the overall structure for derivation trees created by the grammar has little effect on the performance, but still affects the genetic material changed by search operators.

***Geometric Semantic Genetic Programming is Overkill***

Tomasz P. Pawlak

Recently, a new notion of Geometric Semantic Genetic Programming emerged in the field of automatic program induction from examples. Given that the induction problem is stated by means of function learning and a fitness function is a metric, GSGP uses geometry of solution space to search for the optimal program. We demonstrate that a program constructed by GSGP is indeed a linear combination of random parts. We also show that this type of program can be constructed

in a predetermined time by much simpler algorithm and with guarantee of solving the induction problem optimally. We experimentally compare the proposed algorithm to GSGP on a set of symbolic regression, Boolean function synthesis and classifier induction problems. The proposed algorithm is superior to GSGP in terms of training-set fitness, size of produced programs and computational cost, and generalizes on test-set similarly to GSGP.

***Semantic Geometric Initialization***
Tomasz P. Pawlak, Krzysztof Krawiec
A common approach in Geometric Semantic Genetic Programming (GSGP) is to seed initial populations using conventional, semantic-unaware methods like Ramped Half-and-Half. We formally demonstrate that this may limit GSGP's ability to find a program with the sought semantics. To overcome this issue, we determine the desired properties of geometric-aware semantic initialization and implement them in Semantic Geometric Initialization (SGI) algorithm, which we instantiate for symbolic regression and Boolean function synthesis problems. Properties of SGI and its impact on GSGP search are verified experimentally on nine symbolic regression and nine Boolean function synthesis benchmarks. When assessed experimentally, SGI leads to superior performance of GSGP search: better best-of-run fitness and higher probability of finding the optimal program.

* Short presentations must have a corresponding poster in the poster session. Full presentations are encouraged to do likewise.

## 17:00-18:50 Session 3 : Best Papers

### chair: Mengjie Zhang

***(format: 20 minutes per paper + 5 min. questions)***

***Evolutionary Approximation of Edge Detection Circuits***
Petr Dvoracek, Lukas Sekanina
Approximate computing exploits the fact that many applications are inherently error resilient which means that some errors in their outputs can safely be exchanged for improving other parameters such as energy consumption or operation frequency. A new method based on evolutionary computing is proposed in this paper which enables to approximate edge detection circuits. Rather than evolving approximate edge detectors from scratch, key components of existing edge detector are replaced by their approximate versions obtained using Cartesian genetic programming (CGP). Various approximate edge detectors are then composed and their quality is evaluated using a database of images. The paper reports interesting edge detectors showing a good tradeoff between the quality of edge detection and implementation cost.

***Surrogate Fitness via Factorization of Interaction Matrix***
Paweł Liskowski, Krzysztof Krawiec
We propose SFIMX, a method that reduces the number of required interactions between programs and tests in genetic programming. SFIMX performs factorization of the matrix of the outcomes of interactions between the programs in a working population and the tests. Crucially, that factorization is applied to matrix that is only partially filled with interaction outcomes, i.e., sparse. The reconstructed approximate interaction matrix is then used to calculate the fitness of programs. In empirical comparison to several reference methods in categorical domains, SFIMX attains higher success rate of synthesizing correct programs within a given computational budget

***Scheduling in Heterogeneous Networks using Grammar-based Genetic Programming***
David Lynch, Michael Fenton, Stepan Kucera, Holger Claussen, Michael O'Neill
Effective scheduling in Heterogeneous Networks is key to realising the benefits from enhanced Inter-Cell Interference Coordination. In this paper we address the problem using Grammar-based Genetic Programming. Our solution executes on a millisecond timescale so it can track with changing network conditions. Furthermore, the system is trained using only those measurement statistics that are attainable in real networks. Finally, the solution generalises well with respect to

dynamic traffic and variable cell placement. Superior results are achieved relative to a benchmark scheme from the literature, illustrating an opportunity for the further use of Genetic Programming in software-defined autonomic wireless communications networks.

### *On the Analysis of Simple Genetic Programming for Evolving Boolean Functions*

Andrea Mambrini, Pietro S. Oliveto

This work presents a first step towards a systematic time and space complexity analysis of genetic programming (GP) for evolving functions with desired input/output behaviour. Two simple GP algorithms, called (1+1) GP and (1+1) GP*, equipped with minimal function (F) and terminal (L) sets are considered for evolving two standard classes of Boolean functions. It is rigorously proved that both algorithms are efficient for the easy problem of evolving conjunctions of Boolean variables with the minimal sets. However, if an extra function (i.e. NOT) is added to F, then the algorithms require at least exponential time to evolve the conjunction of $n$ variables. On the other hand, it is proved that both algorithms fail at evolving the difficult parity function in polynomial time with probability at least exponentially close to 1. Concerning generalisation, it is shown how the quality of the evolved conjunctions depends on the size of the training set s while the evolved exclusive disjunctions generalize equally badly independent of s.