Nelishia Pillay
Department of Computer Science, University of Pretoria, SOUTH AFRICA

Rong Qu
School of Computer Science, University of Nottingham, UNITED KINGDOM

Dipti Srinivasan
Department of Electrical and Computer Engineering, National University of Singapore, SINGAPORE

Barbara Hammer
CITEC Centre of Excellence, Bielefeld University, GERMANY

Kenneth Sörensen
Department of Engineering Management, University of Antwerp, BELGIUM

Automated Design of Machine Learning and Search Algorithms

Machine learning and search techniques play an important role in solving real-world complex optimization problems in areas such as transportation, data mining, computer vision, computer security and software development, amongst others. Given the growing complexity of optimization problems, the design of effective algorithms to solve these problems has become more challenging and time consuming. The design process is itself an optimization problem. Hence, there is a demand, especially from industry and business, to automate the design process, thereby to remove the heavy reliance on human experts and to reduce the man hours involved in designing machine learning and search algorithms.

There is a demand, especially from industry and business, to automate the design of machine learning and search algorithms, thereby removing the heavy reliance on human experts.

Automated design includes parameter tuning and control. Machine learning and search algorithms require parameters to be tuned, with the most appropriate parameter values being problem dependent. For example, genetic operator probabilities have to be decided for genetic algorithms. Similarly, hyperparameters, e.g. the learning rate in deep learning, needs to be chosen. There are usually many options for hyperparameters. Automating the selection of these values reduces the time and human expertise required for this. Automating parameter selection allows for the parameter values to be configured and adapted dynamically during execution of the algorithm, resulting in parameter-less algorithms. For some algorithms, e.g. evolutionary algorithms, it is also necessary to select which operators to use. Automating this process allows for different operators to be applied at different points in the algorithm.

Research into automated design of machine learning and search algorithms has also focused on generating new constructs used by these algorithms. These constructs have ranged from construction heuristics that are used to create initial solutions that these algorithms optimize further, creating operators used by search algorithms, generating machine learning workplans and architectures, to the induction of solution algorithms and software development.

Identifying an appropriate algorithm, or combination of algorithms, to solve the problem at hand has also been investigated as part of automated design of machine learning and search algorithms. One approach to select an algorithm to solve a particular problem is to identify a correlation between problem features and the algorithm most suitable for solving the problem. Rules representing a mapping of this relationship have been induced for this purpose. The second paper presented in this special issue examines applying transformation functions to the problem features to overcome the problems of stagnation and likeliness that arise in evolving and applying such rules.

Hyper-heuristics, specifically selection hyper-heuristics, and evolutionary algorithms have been examined for hybridizing algorithms to solve problems. For example, the first paper in this special issue reports on using a coevolutionary algorithm to combine support vector machines to solve multiclass classification problems.

Various techniques have been examined for automated design, however the

Digital Object Identifier 10.1109/MCI.2018.2806988
Date of publication: 10 April 2018

most effective techniques have proven to be the machine learning and search algorithms themselves. Evolutionary algorithms, specifically genetic programming and variations thereof, have played a pivotal role in inducing new constructs such as construction heuristics, operators and software. Evolutionary algorithms have also been used for designing architectures, e.g. neural network architectures, and parameter control and tuning. Hyper-heuristics, which explore a heuristic space rather than a solution space, have also been shown to be effective for parameter control, operator selection and hybridizing algorithms. The final paper presented in this special issue critically examines the role played by evolutionary algorithms, specifically genetic programming, and hyper-heuristics in software development and maps a way forward.

Evolutionary algorithms, specifically genetic programming and variations thereof, have played a pivotal role in inducing new constructs such as construction heuristics, operators and software.

The purpose of this special issue on "Automated Design of Machine Learning and Search Algorithms" is to report on current trends in the field. Sixteen high quality papers were submitted for consideration for the special issue. After a rigorous review process, three papers were selected for publication.

The first paper, "MC2ESVM: Multiclass Classification Based on Cooperative Evolution of Support Vector Machines" by Alejandro Rosales-Pérez, Salvador García, Hugo Terashima-Marín, Carlos A. Coello Coello and Francisco Herrera, presents a coevolutionary algorithm for designing a support vector machine approach for multiclass classification. Support vector machines (SVMs) have proven to be effective at binary classification. However, in solving multiclass classification problems the problem is decomposed into a number of binary classification problems and an SVM is used to solve each problem. This paper presents an approach that automates the process of combining SVMs to solve multiclass classification problems. The coevolutionary algorithm evolves separate support vector machine populations for each of the classes. The proposed approach is evaluated on 25 data sets from the KEEL repository. Two measures are used to assess the performance of the approach, namely, accuracy and Cohen's kappa. MC2ESVM outperformed both standard decomposition and single machine SVM methods and common approaches, such as random forests and neural networks, traditionally used for multiclass classification.

The second paper "Enhancing Selection Hyper-Heuristics via Feature Transformation" by Ivan Amaya, José C. Ortiz-Bayliss, Alejandro Rosales-Pérez, Andrés E. Gutiérrez-Rodríguez, Santiago E. Conant-Pablos, Hugo Terashima-Marín, explores the use of feature transformations to improve the performance of selection hyper-heuristics. Selection hyper-heuristics are used to select a heuristic for a given set of problem features. The set of features and the corresponding heuristic take the form of a rule. When applying a rule the features of the current state of the problem are compared to the features of each rule and the heuristic of the closest matching rule is applied. In the study presented in the paper a messy genetic algorithm is used to induce the rules. Two problems encountered when deriving and applying such rules are likeliness and stagnation. The paper presents transformation of features as a means of overcoming these problems. Two types of transformations, namely, explicit and implicit, are studied and tested on constraint satisfaction and knapsack problems. The study has shown that the use of explicit and implicit transformations improves the performance of selection hyper-heuristics, however, combining both types of transformations does not work well.

The third paper entitled "Is Evolutionary Computation Evolving Fast Enough?" is a position paper by Graham Kendall which critically evaluates the impact of evolutionary algorithms, specifically genetic programming and hyper-heuristics, on solving real world problems such as automated software development. The paper provides an overview of applications of evolutionary algorithms to problems in industry and emphasizes that there have not been many such applications. The use of evolutionary algorithms by industry has not developed as well as other artificial intelligence techniques. One of the reasons for this highlighted in the paper is that research has generally applied evolutionary algorithms to models of real world problems using benchmark sets rather than the real world problems themselves. Although genetic programming has provided the hope of "automatic programming", it has not advanced the field of software development to the point that it can easily be used by non-experts to develop software. Similarly, hyper-heuristics have not made the anticipated impact on industry. The paper highlights the slow adoption of these techniques for software development outside of academia and proposes a way forward to bridge this gap between academia and industry.

We would like to thank all the authors for submissions of high quality papers for the special issue. We would also like to thank the reviewers for their invaluable contribution in assessing the submissions to the special issue. Our final thanks are to the editor-in-chief Professor Hisao Ishubuchi for the opportunity to publish the special issue and his support and advice throughout the process.