IEEE Computational Intelligence MAGAZINE

MAY 2018
VOLUME 13 NUMBER 2
WWW.IEEE-CIS.ORG

IEEE
Computational
Intelligence
Society

IEEE

2018 IEEE Symposium Series on Computational Intelligence

IEEE SSCI 2018

18 Sun – 21 Wed Nov 2018, Bengaluru, India

Call for Papers http://www.ieee-ssci2018.org/

The 2018 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2018) is a flagship annual international conference sponsored by the IEEE Computational Intelligence Society promoting all aspects of computational intelligence. The IEEE SSCI 2018 co-locates several symposia in one roof, each dedicated to a specific topic in the computational intelligence domain, thereby encouraging cross-fertilization of ideas and providing a unique platform for top researchers, professionals, and students from all around the world to discuss and present their findings. The IEEE SSCI meeting features a large number of keynote addresses, tutorials, panel discussions and special sessions all of which are open to all the participants. Each symposium will award the best student paper and a best overall paper award. The IEEE SSCI 2018 will be held in Bengaluru, India, a garden city, also known as the Silicon Valley of India.

Symposia and Workshops

ADPRL, IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning

CIASG, IEEE Symposium on Computational Intelligence Applications in Smart Grid

CIBD, IEEE Symposium on Computational Intelligence in Big Data

CICA, IEEE Symposium on Computational Intelligence in Control and Automation

CICARE, IEEE Symposium on Computational Intelligence in Healthcare and E-health

CIComms, IEEE Symposium on Computational Intelligence for Communication Systems

CICS, IEEE Symposium on Computational Intelligence in Cyber Security

CIDM, IEEE Symposium on Computational Intelligence and Data Mining

CIDUE, IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments

CIEG, IEEE Symposium on Computational Intelligence in E-governance

CIEL, IEEE Symposium on Computational Intelligence and Ensemble Learning

CIES, IEEE Symposium on Computational Intelligence for Engineering Solutions

CIFEr, IEEE Symposium on Computational Intelligence for Financial Engineering and Economics

CIHLI, IEEE Symposium on Computational Intelligence in Human-like Intelligence Industry

CIMSIVP, IEEE Symposium on Computational Intelligence for Multimedia Signal and Vision Processing

CIR2AT, IEEE Symposium on Computational Intelligence in Robotics Rehabilitation and Assistive Technologies

CISDA, IEEE Symposium on Computational Intelligence for Security and Defense Applications

CISND, IEEE Symposium on Computational Intelligence in Scheduling and Network Design

CIVTS, IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems

DL, IEEE Symposium on Deep Learning

EALS, IEEE Symposium on Evolving and Autonomous Learning Systems

FASLIP, IEEE Symposium on Computational Intelligence in Feature Analysis, Selection, and Learning in Image and Pattern Recognition

FOCI, IEEE Symposium on Foundations of Computational Intelligence

IA, IEEE Symposium on Intelligent Agents

ICES, IEEE Conference on Evolvable Systems

IntECS, IEEE Symposium on Computational Intelligence for Embedded and Cyberphysical Systems

MBEA, IEEE Symposium on Model-Based Evolutionary Algorithms

MCDM, IEEE Symposium on Multi-criteria Decision-Making

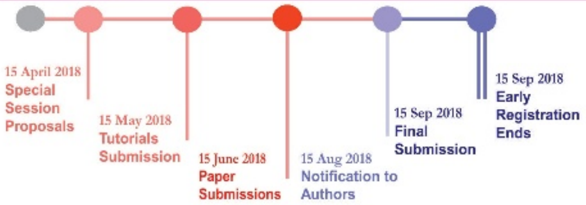
RiiSS, IEEE Workshop on Robotic Intelligence in Informationally Structured Space

SDE, IEEE Symposium on Differential Evolution

SIS, IEEE Swarm Intelligence Symposium

SNCC, IEEE Symposium on Neuromorphic Cognitive Computing

Important Dates



15 April 2018 Special Session Proposals

15 May 2018 Tutorials Submission

15 June 2018 Paper Submissions

15 Aug 2018 Notification to Authors

15 Sep 2018 Final Submission

15 Sep 2018 Early Registration Ends

Sponsored by

IEEE

IEEE Computational Intelligence Society

Organizing Committee

Honorary Chair:
N R Pal, India

General Co-Chairs:
S Suresh, Singapore
Koshy George, India

Program Chair:
B K Panigrahi, India
Julia Chung, Taiwan

Finance Chair:
P N Suganthan, Singapore

Organizing Chair:
J Senthilnath, Singapore

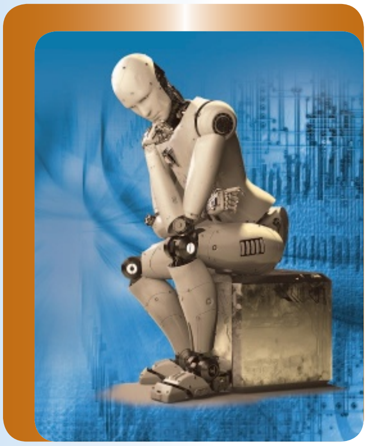
Contact Us:
ssci2018@gmail.com

Digital Object Identifier 10.1109/MCI.2018.2807078

IEEE Computational Intelligence MAGAZINE

Volume 13 Number 2 ❑ May 2018
www.ieee-cis.org

Features



on the cover

ROBOT—©ISTOCKPHOTO.COM/PHONLAMAIPHOTO,
BACKGROUND—FOOTAGE FIRM, INC.

Columns

Departments

IEEE Computational Intelligence Magazine (ISSN 1556-603X) is published quarterly by The Institute of Electrical and Electronics Engineers, Inc. **Headquarters:** 3 Park Avenue, 17th Floor, New York, NY 10016-5997, U.S.A. +1 212 419 7900. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society, or its members. The magazine is a membership benefit of the IEEE Computational Intelligence Society, and subscriptions are included in Society fee. Replacement copies for members are available for US$20 (one copy only). Nonmembers can purchase individual copies for US$201.00. Nonmember subscription prices are available on request. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of the U.S. Copyright law for private use of patrons: 1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01970, U.S.A.; and 2) pre-1978 articles without fee. For other copying, reprint, or republication permission, write to: Copyrights and Permissions Department, IEEE Service Center, 445 Hoes Lane, Piscataway NJ 08854 U.S.A. Copyright © 2018 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Periodicals postage paid at New York, NY and at additional mailing offices. Postmaster: Send address changes to IEEE Computational Intelligence Magazine, IEEE, 445 Hoes Lane, Piscataway, NJ 08854-1331 U.S.A. PRINTED IN U.S.A. Canadian GST #125634188.

Digital Object Identifier 10.1109/MCI.2017.2770272

Digital Object Identifier 10.1109/MCI.2017.2770273

Hisao Ishibuchi
Southern University of Science and Technology, CHINA
Osaka Prefecture University, JAPAN

IEEE CIM Survey Results

Last October, the CIM conducted a web-based survey. An e-mail about the survey was sent to 4000 CIS members randomly selected from all IEEE member grades. There were 662 member responses to the survey. Thank you very much for your help. The survey results were great. For example, 90% of the respondents think that the overall quality of the CIM is high, and 77% of them agree with the following statement: "The articles in the publication are relevant to my work or interests". Most members read the CIM electronically. The rest read the printed version of the magazine (24%) or print out a downloaded PDF file (15%). Under the Departments section, the Publication Spotlight had the highest readership (52%) followed by the Book Review (47%), Journal Call-for-Papers (47%), and Conference Call-for-Papers (42%). I am very happy that the CIM is highly valued by the CIS members and is playing an important role as a communication channel between the society and its members through those non-technical articles. Also, the readership of the Editor's Remarks was 30%, which is slightly higher than the President's Message at 24%.

The survey also included a question about the respondent's interest in various research areas. The question was "Which of the following topical areas, if any, should be present in the magazine? Please select all that apply." For this question, a list of 16 areas were selected. I tried to include all the popular research areas related to computational intelligence. The list included robotics, natural language processing, brain machine interface, and image processing, which were supported by 40-50% of the respondents. The most popular areas in the survey were artificial intelligence (86%), machine learning (83%), deep learning (76%), big data (61%), and optimization (55%). Since computational intelligence is central to many recent advances in artificial intelligence and machine learning, I hope that the CIM will receive more submissions on these upcoming topics. The CIM is planning to publish a special issue on "Deep Reinforcement Learning and Games" soon! (submission deadline: October 1, 2018).

The feature topic of the current CIM issue is "Automated Design of Machine Learning and Search Algorithms". Fully automated algorithm design may be an ultimate goal of algorithm researchers. It would be nice if a meta-algorithm could automatically design an appropriate algorithm for each problem at hand depending on its characteristics. If we have a high-performance meta-algorithm that is applicable to various application fields, we do not have to design a different algorithm for a different problem. However, it is extremely difficult to develop such a meta-algorithm since the design of even a single high-performance algorithm to a specific problem is very difficult. The core theme in this field is hyper-heuristics. Genetic programming is also closely related. You will find the latest developments in this field from the feature articles in the current CIM issue. I hope you will enjoy all articles in this issue.

Hisao Ishibuchi

Digital Object Identifier 10.1109/MCI.2018.2806981
Date of publication: 10 April 2018

Nikhil R. Pal
Indian Statistical Institute, INDIA

Random Thoughts: "Comprehensible & Sustainable" Computational Intelligence

My journey as the President of the IEEE Computational Intelligence Society (CIS) began on January 1, 2018 quite smoothly because of enormous help by my predecessor, Pablo (Estevez) and other friends at CIS including Jo-Ellen (Snyder) and Tom (Compton). So far it has been an enjoyable ride, but I have realized that there are challenges and it is fun to deal with them.

In my last message I discussed some of the important issues related to the present day "intelligent systems". Here I express my personal views on two other important issues: comprehensibility and sustainability. I think time has come for us to emphasize more on "comprehensible and sustainable" Computational Intelligence (CI)/Artificial intelligence (AI).

Good performance is definitely a requirement for any intelligent system. But if the system is not comprehensible/understandable at all, sometimes the system may fail with catastrophic consequences and we may not have any clue of that ahead of time. For example, a deep neural network, known for its unmatched performance, as of now, is a "black box" and we all know of some very simple situations where it miserably failed [1], [2].

Comprehensibility is a fuzzy concept with grades of membership beginning from zero for a black box system to one for a completely transparent system. For example, decision trees or crisp rules are highly comprehensible as along as the number of conditions involved is small. But with an increase in the depth of the tree or the length of the rules the level of comprehensibility reduces, yet they are comprehensible to some degree compared to, for example, a multilayer perceptron. In this context, a fuzzy rule-based system is highly comprehensible when the number of antecedent clauses in a rule is limited. Even when the number of antecedent clauses is high, because of the very nature of fuzzy reasoning, it is easy to visualize how fuzzy rules work and why it is unlikely to make an unexpected decision/generalization. But these systems, although can provide understandability, are usually poor performers compared to support vector machines, deep neural networks, or even multilayer perceptron networks. Thus, it would be good, if we could inject some level of comprehensibility into such systems to realize both comprehensibility and good performance. Fuzzy sets could be a possible vehicle for this. In fact, incorporation of fuzzy concepts may even help to deal with uncertainty. Some attempts have been made in this direction, but it deserves more attention – we need more emphasis on comprehensible CI.

Now I turn to the other issue, sustainable CI/AI: According to the Oxford Dictionary, the word sustainability means "The ability to be maintained at a certain rate or level". Sustainability demands efficient use of energy, use of renewable energy, and preservation of natural resources and our environment. To design an intelligent system for a given problem almost always we focus on maximizing/minimizing something that will help to satisfy our needs. Usually, these needs are our immediate needs. While

Digital Object Identifier 10.1109/MCI.2018.2806982
Date of publication: 10 April 2018

Digital Object Identifier 10.1109/MCI.2018.2824958

> **The goal of an integrated smart system for crop management should be to *Maximize* the yield using *Minimum* resources (human effort/cost - not necessarily money) with *Minimal* impact on the environment for serving the *Maximum* (inclusiveness, catering social needs) (M^4).**

designing a system, primarily we focus on achieving the best accuracy for the assigned task. It is more often than not, we forget or ignore the long term environmental impacts of what we do. But why should we, who primarily develop learning algorithms for problem solving, care about it? The answer lies here. The carbon footprint of using just a common server is much more than what we can imagine. The total carbon footprint of a Typical Dell PowerEdge R710 rack server is 6360 kg CO_2eq assuming a four year life which is comparable to driving 21,500 km in an SUV [3]! The specific configuration of the machine used to arrive at the estimate of carbon footprint is: 2 processors (Intel Xeon); 12 GB of RAM; 4 × 146 GB hard drives (HDD); 2 high output power supplies; 1 DVD drive; and 4 fans. This is just an illustration, newer servers with similar computing power may have lower carbon overhead while others may have more. Computer hardware companies are trying to address this issue. We also have a role to play. Often our learning algorithms run for days/weeks on a much more powerful platform on a big data set. One can easily imagine how much impact computers, in particular data centers, can have on our environment in terms of carbon footprint. So when we design our next algorithm, we should take this factor into account.

The other important facet of sustainability is related to the solutions that we provide. To emphasize this issue I take the help of smart agriculture systems. On September 25 2015, United Nations set 17 goals to transform our world [4]. The second goal is: End hunger, achieve food security and improved nutrition and promote *sustainable agriculture* (the first goal is: End poverty in all its forms everywhere). It emphasizes that the goal of a smart agriculture system should be not just maximizing the yield but also to ensure the sustainability. Sustainable technologies should cater to the needs of the present generation without compromising the needs of the future generation. Just to clarify, suppose we want to develop an integrated smart system for crop management. Our goal should be to *Maximize* the yield using *Minimum* resources (human effort/cost - not necessarily money) with *Minimal* impact on the environment for serving the *Maximum* (inclusiveness, catering social needs) (M^4). Such systems should be able to guide farmers on the following: What to grow? What would be the optimal distribution of different crops (farmer level, state level, country level)? When and how much to irrigate? When, which and how much fertilizers to use? When, which and how much pesticides to use? And when to harvest? To realize sustainability, the system should assist farmers on all these issues imposing constraint, for example, on the use of nitrogen, pesticides, and water. Why? Nitrate may lead to better yield, but has serious environmental impacts. It pollutes water, kills plants that need low level of nitrogen, promotes growth of non-native grasses and kills lichens, and causes a decline in native species. It has also been linked with causes of many diseases including methemoglobinemia, cancer, birth defects, and hyperthyroidism [5], [6]. So, we need to ensure the use of the Right nutrient at the Right rate in the Right place, and at the Right time (R^4). Similarly, a smart system should help to minimize the usage of water and pesticides. All these demand complex modeling and optimization to develop systems for prediction of weather, prediction of nutrients' need (at the level of small units of land), control of drone-assisted delivery of precise dose of pesticides and so on. Just in the area of agriculture, there are many other environment-sensitive challenging problems. CI provides a set of very useful tools for these. As examples, for some of the problems, the objective functions to be optimized may not be differentiable and in that case evolutionary algorithms could be handy tools. Like anyone else, farmers will not like black-box type systems. Here use of fuzzy modeling, wherever possible and useful, could be attractive.

To conclude, I would like to emphasize that while designing intelligent/smart systems, we need to take the comprehensibility and the sustainability of the algorithms as well as the sustainability of the solutions they provide much more seriously than we have been doing. These are certainly very difficult and challenging tasks and we need to make an effort to address them.

References

[1] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Computer and Communications Security*, Apr. 2017, pp. 506–519.

[2] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[3] http://i.dell.com/sites/content/corporate/corp-comm/en/Documents/dell-server-carbon-footprint-whitepaper.pdf (Accessed on Feb. 1, 2018).

[4] http://www.un.org/sustainabledevelopment/sustainable-developmentgoals/ (Accessed on Feb. 1, 2018).

[5] L. Knobeloch, B. Salna, A. Hogan, J. Postle, and H. Anderson, "Blue babies and nitrate-contaminated well water," *Environ. Health Perspectives*, vol. 108, no. 7, pp. 675–678, 2000.

[6] M. H. Ward, B. A. Kilfoy, P. J. Weyer, K. E. Anderson, A. R. Folsom, and J. R. Cerhan, "Nitrate intake and the risk of thyroid cancer and thyroid disease," *Epidemiology*, vol. 21, no. 3, pp. 389–395, 2010.

Nikhil R. Pal
Indian Statistical Institute, INDIA

Newly Elected CIS Administrative Committee Members (2018–2020)

Janusz Kacprzyk
Polish Academy of Sciences, POLAND

Janusz Kacprzyk received the M.Sc. in Automatic Control and Computer Science, Ph.D. in Systems Analysis, and D.Sc. in Computer Science. He is a Professor of Computer Science at the Systems Research Institute, Polish Academy of Sciences, and at WIT—Warsaw School of Information Technology, Professor of Automatic Control at PIAP—Industrial Institute of Automation and Measurements, Professor at Chogqing Three Gorges University in China, and Honorary Professor at the Mathematics Department, Yli Normal University in China. He is Full Member of the Polish Academy of Sciences, Academia Europaea (Informatics), European Academy of Sciences and Arts (Technical Sciences), Foreign Member of the Spanish Royal Academy of Economic and Financial Sciences (RACEF), and Bulgarian Academy of Sciences. He is Fellow of IEEE, IET, IFSA, EurAI (ECCAI), SMIA. He received four honorary doctorates (doctor honoris causa) from universities in Hungary, Finland and Bulgaria.

His main research interests include CI and AI tools, notably fuzzy logic, in decisions, optimization, control, data analysis and data mining, with applications in databases, IT/ICT, mobile robotics, systems modeling etc. He authored 6 books, (co)edited more than 100 volumes, (co)authored about 550 papers, and about 80 in JCR journals with 22461 citations and h-index 67 (Google Scholar), 6754 citations and h-index 37 (Scopus), and 4796 citations and h-index 31 (WoS). He is the editor in chief of 7 book series at Springer, and of 2 journals, and on the editorial boards of about 40 journals. He is on the IEEE CIS Fellows Committee, was Chair of 2016 IEEE CIS Award Committee, a member of IEEE CIS Adcom in 2011-2016, and a Distinguished Lecturer of IEEE CIS in 2011-2014.

He received 2006 IEEE CIS Pioneer Award in Fuzzy Systems, 2006 Sixth Kaufmann Prize and Gold Medal for pioneering works on soft computing in economics and management, 2007 Pioneer Award of the Silicon Valley Section of IEEE CIS for contribution in granular computing and computing in words, 2010 Award of the Polish Neural Network Society for exceptional contributions to the Polish CI community, IFSA 2013 Award for his lifetime achievements in fuzzy systems, 2014 World Automation Congress Lifetime Award, 2016 Award of the INNS—Indian Chapter for Outstanding Contributions to CI, 2017 HAFSA (Hispanic American Fuzzy Systems Society) Recognition for pioneering works in fuzzy logic and systems. He is President of the Polish Operational and Systems Research Society and Past President of International Fuzzy Systems Association.

Sanaz Mostaghim
Otto von Guericke University of Magdeburg, GERMANY

Sanaz Mostaghim is a Professor of Computer Science at the Otto von Guericke University Magdeburg (OvGU), Germany and holds a Ph.D. degree (2004) in electrical engineering and computer science from the University of Paderborn, Germany. She worked as a postdoctoral fellow at ETH Zurich in Switzerland (2004-2006) and as a lecturer at Karlsruhe Institute of technology (KIT), Germany (2006-2013), where she received her habilitation degree in applied computer science in 2012. Her research interests are in the area of swarm intelligence, evolutionary multi-objective optimization and decision making as well as their applications in robotics, science and industry.

Sanaz is an active member at IEEE CIS. She has been a member of AdCom 2015-2017, served as the Chair of Women in Computational Intelligence (2015-2016) and the Chair of two IEEE CIS task forces: Evolutionary Multi-objective Optimization (2013-2017) and Optimization Methods in Bioinformatics and Bioengineering (2013-2015). Currently she is the Chair of Young Professional Committee at IEEE CIS. She is an Associate Editor of *IEEE Trans. on Evolutionary Computation*, *IEEE Trans. on Emerging Topics in Computational Intelligence*, *IEEE Trans. on Cybernetics*, *IEEE Trans. on Systems, Man and Cybernetics:*

Digital Object Identifier 10.1109/MCI.2018.2806983
Date of publication: 10 April 2018

Systems, and a member of the Editorial Board of springer journal on *Complex and Intelligent Systems*.

Christian Wagner
University of Nottingham, UK

Christian Wagner is an Associate Professor in Computer Science at the University of Nottingham, UK and a Visiting Professor in Cybersystems at Michigan Technological University, USA. His research focuses on modelling and handling of uncertain data arising both from qualitative (people) and quantitative sources (e.g., sensors, processes), decision support systems and data-driven policy design; frequently in an inter-disciplinary setting.

He has published around 100 peer-reviewed articles, including prize-winning papers in international journals and conferences. He is Director of the Lab for Uncertainty in Data and Decision Making (LUCID: http://lucidresearch.org) with which he and his collaborators recently became runners-up for both the best regular and best student papers at the IEEE International Conference on Fuzzy Systems 2016 in Vancouver, Canada. He has attracted over US$10 million in research funding as principal and co-investigator in the last 6 years. He is an Associate Editor of the *IEEE Trans. on Fuzzy Systems*, is Chair of the IEEE CIS Task Force on Cyber Security and an elected member of the IEEE Computational Intelligence Society (CIS) Administrative Committee for 2018-2010.

Dr Wagner has co/developed multiple open source software frameworks, making cutting edge research accessible both to peer researchers as well as to different research communities beyond computer science, including an R toolkit for type-2 fuzzy systems and a new Java based framework for the object oriented implementation of general type-2 fuzzy sets and systems. His current research projects focus on the development, adaptation, deployment and evaluation of artificial intelligence techniques in inter-disciplinary projects bringing together heterogeneous data from stakeholders and quantitative measurements to support informed and transparent decision making in cyber security, environmental management and manufacturing.

Ronald R. Yager
Iona College, USA

Ronald R. Yager has worked in the area of computational intelligence for over twenty-five years. He is Director of the Machine Intelligence Institute and Professor of Information Systems at Iona College. He is the editor-in-chief of the *International Journal of Intelligent Systems*. He has published over 500 papers and edited over 30 books in areas related to fuzzy sets, human behavioral modeling, decision-making under uncertainty and the fusion of information. He is among the world's top 1% most highly cited researchers with over 58,000 citations in Google Scholar. He was the recipient of the IEEE Computational Intelligence Society's highly prestigious Frank Rosenblatt Award in 2016. He was also the recipient of the IEEE Computational Intelligence Society's Pioneer award in Fuzzy Systems in 2004. He received the special honorary medal of the 50-th Anniversary of the Polish Academy of Sciences. He received the Lifetime Outstanding Achievement Award from International Fuzzy Systems Association. He received honorary doctorate degrees, honoris causa, from the Azerbaijan Technical University and the State University of Information Technologies, Sofia Bulgaria. Dr. Yager is a fellow of the IEEE, the New York Academy of Sciences and the International Fuzzy Systems Association. He has served at the National Science Foundation as program director in the Information Sciences program. He was a NASA/Stanford visiting fellow and a research associate at the University of California, Berkeley. He has been a lecturer at NATO Advanced Study Institutes. He was a program director at the National Science Foundation. He was a visiting distinguished scientist at King Saud University, Riyadh, Saudi Arabia. He received his undergraduate degree from the City College of New York and his Ph.D. from the Polytechnic Institute New York University. He has been involved in a number of IEEE activities including serving as General Chair of the IEEE Conference on Financial Engineering and Intelligent Systems. He is also a member of a number of IEEE task forces.

Gary G. Yen
Oklahoma State University, USA

Gary G. Yen received his Ph.D. degree in Electrical and Computer Engineering from the University of Notre Dame in 1992. He is currently a Regents Professor in the School of Electrical and Computer Engineering, Oklahoma State University. His research interest includes intelligent control, evolutionary multiobjective optimization, conditional health monitoring, signal processing and their industrial/defense applications.

Gary was an Associate Editor of the *IEEE Trans. on Neural Networks* in 1994-1999. He is currently serving as an Associate Editor of the *IEEE Trans. on Evolutionary Computation* since 2005 and of the *IEEE Trans. on Emerging Topics on Computational Intelligence* since 2017 and was the founding Editor-in-Chief of the *IEEE Computational Intelligence Magazine* in 2006-2009. Gary served as the Chair of Neural Network Technical Committee in 2000-2002, Vice President for the Technical Activities in 2004-2005, and the President of the IEEE Computational Intelligence Society in 2010-2011. He also served as the Chair of CIS Awards Committee in 2008-2009 and 2014-2015, and the Chair of CIS Fellows Committee in 2016-2017. In 2006, Gary chaired the IEEE World Congress on Computational Intelligence (WCCI) held in Vancouver, Canada, and for the same conference, served as Finance Chair in Brisbane, Australia 2012, Plenary Session Chair in Beijing, China 2014, Co-General Chair in Vancouver, Canada again in 2016, and Conference Chair for IEEE CEC 2018 in Rio de Janeiro, Brazil. At the IEEE level, Gary served as a member of IEEE Technical Activities Board Award and Recognition Committee in 2010-2011, IEEE Technical Activities Board Product and Service Publication Committee in 2011-2012, IEEE Fellow Committee in 2012-2014, and TAB Periodicals Review Advisory Committee in 2015-2018.

He received Regents Distinguished Research Award from Oklahoma State University in 2009, 2011 Andrew P Sage Best Transactions Paper award from IEEE Systems, Man and Cybernetics Society, and 2013 Meritorious Service award from IEEE Computational Intelligence Society. In 2012 and again in 2017, he was elected as a Distinguished Lecturer for the Society. He is a Fellow of IEEE.

Gary Yen
IEEE CIS 2017 Fellow Committee Chair, USA

IEEE Fellows—Class of 2018

Yiu-ming Cheung
Hong Kong Baptist University, Hong Kong SAR, CHINA
For contributions to cluster analysis and visual computing.

Yiu-ming Cheung is currently a Full Professor at Department of Computer Science in Hong Kong Baptist University (HKBU). He received Ph.D. degree from Department of Computer Science and Engineering at The Chinese University of Hong Kong in 2000, and then joined the Department of Computer Science at HKBU in 2001. He is an IET/IEE Fellow, British Computer Society (BCS) Fellow, and Fellow of International Engineering and Technology Institute, Hong Kong (IETI), as well as the "Chu Tian Scholars" in China.

His research interests include Computational Intelligence, Statistical Learning, Intelligent Visual Computing, Pattern Recognition, Data Mining, and Watermarking. He has published over 200 research articles and has been granted three invention patents. In recognition of his innovative work, he was awarded two prestigious prizes: (1) the Gold Medal with Distinction (i.e. the highest grade in Gold Medals) and (2) Swiss Automobile Club Prize, both of which were selected from 1000 new inventions and products of 700+ competition teams from 40 countries, in the 45th International Exhibition of Invention, Geneva, in 2017. Furthermore, he was the Gold Award Winner of Hong Kong Innovative Invention Award in the Seventh Hong Kong Innovative Technologies Achievement Award 2017. In addition, he was the recipient of 2011 Best Research Award in Department of Computer Science, HKBU, and the recipient of Best Paper Awards in SEAL'2017, ISICA'2017, ICNC-FSKD'2014, and IWDVT'2005, respectively.

He is the Founding Chairman of IEEE (Hong Kong) Computational Intelligence Chapter and the Vice Chair of IEEE Computer Society Technical Committee on Intelligent Informatics (TCII). He has served in various capacities (e.g., Organizing Committee Chair, Program Committee Chair, Program Committee Area Chair, Financial Chair, etc.) in several top-tier international conferences, including WCCI 2016, ICDM 2006 & 2017, WI-IAT 2006 & 2012. Currently, he is an Associate Editor of *IEEE Trans. on Neural Networks and Learning Systems*, *Pattern Recognition*, *Knowledge and Information Systems* (*KAIS*), to name a few.

Oscar Cordón
University of Granada, SPAIN
For contributions to genetic and evolutionary fuzzy systems.

Oscar Cordón received his M.S. degree (1994) and his Ph.D. (1997) both in Computer Science from the University of Granada, Spain, where he is currently Professor at the Department of Computer Science and Artificial Intelligence. He was the founder and leader of this University's Virtual Learning Center (2001-2005) and is

Digital Object Identifier 10.1109/MCI.2018.2806985
Date of publication: 10 April 2018

the Vice-President for Digital University since 2015. He was founding Principal Researcher (2006-2011) and Distinguished Affiliated Researcher (2011-2015) of the European Centre for Soft Computing.

He has been, for more than 23 years, an internationally recognized contributor to R&D Programs in fundamentals and real-world applications of computational intelligence. He has published around 340 peer-reviewed scientific publications (including a research book on genetic fuzzy systems with more than 1230 citations in Google Scholar and 98 JCR-SCI-indexed journal papers, 55 in Q1), advised 18 Ph.D. dissertations, coordinated 30 research projects and contracts (with an overall amount of 7.4 M€), has a granted international patent on an intelligent system for forensic identification commercialized in Mexico and South Africa, and is currently or was an Associate Editor of 16 international journals. By December 2017, he is included in the 1% of most-cited researchers in the world (source: Thomson's Web of Knowledge, h-index = 32) and has received around 11700 citations in Scholar Google (h-index = 50).

He was awarded with the IEEE CIS Outstanding Early Career Award in its 2011 edition, the first such award conferred; the IFSA Award for Outstanding Applications of Fuzzy Technology in 2011, and the Spanish National Award on Computer Science ARITMEL by the Spanish Computer Science Scientific Society in 2014. He has taken many different representative positions with Eusflat and IEEE CIS. Among them, he was an elected member of IEEE CIS AdCom (2010-2012) as well as General Chair of FUZZ-IEEE 2016 and Technical Co-Chair of IEEE CEC 2015, 2017 and 2019, FUZZ-IEEE 2020, and IFSA-EUSFLAT 2015.

Xinping Guan
Shanghai Jiao Tong University, CHINA

For contributions to stability analysis for time-delay fuzzy systems and intelligent control of nonlinear systems.

Xinping Guan received the BSc degree in mathematics from Harbin Normal University, P. R. China in 1986, and the Ph.D. degree in Control Science and Engineering from Harbin Institute of Technology, P. R. China in 1999. He is currently a Chair Professor at the Department of Automation, Shanghai Jiao Tong University, P. R. China.

His research interests include fuzzy control, neural network based control and optimization of complex systems. He has published more than 200 journal papers and co-authored 6 books. According to Google Scholar, his publications have received more than 8400 citations with h-index 45. Dr. Guan has made seminal contributions to stability analysis and synthesis of time delay T-S fuzzy systems since 1999. The generalized parallel distributed control (GPDC) method proposed by Dr. Guan is recognized as the first delay-dependent stability analysis for time-delay T-S fuzzy systems. This technique has been seen as a milestone by establishing an analytical framework, and the stability criteria are generally acknowledged as benchmarks for conservatism comparison by peers. He received the "IEEE Transaction on Fuzzy Systems Outstanding Paper Award for 2005". He also received the Second Prize of National Natural Science Award of China in 2008, and the First Prize of Natural Science Award of Ministry of Education of China in 2006 and 2016, respectively.

He is/was on the editorial board of *IEEE Trans. on Systems, Man and Cybernetics-Part C* and five other international journals. He is an Executive Committee Member of Chinese Automation Association Council and the Chinese Artificial Intelligence Association Council. He also serves as Chair Technical Program Committee Member for more than 60 international conferences. He is a "National Outstanding Youth" awarded by National Natural Science Foundation of China, a distinguished professor of "Changjiang Scholar Program", and a "State-level Scholar" of "New Century Bai Qianwan Talent Program" of China.

Haibo He
University of Rhode Island, USA

For contributions to adaptive learning.

Haibo He received the B.S. and M.S. degrees in Electrical Engineering from Huazhong University of Science and Technology (Wuhan, China) in 1999 and 2002, respectively, and the Ph.D. degree in Electrical Engineering from Ohio University (Athens, USA) in 2006. From 2006 to 2009, he was an Assistant Professor at the Department of Electrical and Computer Engineering at Stevens Institute of Technology (Hoboken, USA). Currently, he is the Robert Haas Endowed Chair Professor at the Department of Electrical, Computer, and Biomedical Engineering at the University of Rhode Island (Kingston, USA).

His research focuses on adaptive learning and its wide applications in cyber-physical systems, such as smart grid, smart city, robotics, communication systems, and cyber security. He has published one sole-author research book, edited one book (Wiley-IEEE) and six conference proceedings (Springer), and authored/co-authored over 280 high profile journal and conference papers, including several highly cited papers, IEEE Transaction cover page paper, spotlight paper, and best papers. His classic paper entitled "Learning from Imbalanced Data" (*IEEE Trans. on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009) has received more than 2700 citations.

He has served the IEEE Computational Intelligence Society (CIS) at various capacities, including Chair of IEEE CIS Emergent Technologies Technical Committee (ETTC) (2015) and Chair of IEEE CIS Neural Networks Technical Committee (NNTC) (2013 and 2014). He was the Finance Chair of the IEEE World Congress on Computational Intelligence (IEEE WCCI'16), General Chair of the IEEE Symposium Series on Computational Intelligence (IEEE SSCI'14), Technical Program Co-Chair of the International Joint Conference on Neural Networks (IJCNN'15), among others. He has served as an Associate Editor for *IEEE Trans. on Smart Grid* (2010-2015) and *IEEE Computational Intelligence Magazine* (2015), among others. Currently, he is the Editor-in-Chief of *IEEE Trans. on Neural Networks and Learning Systems*. He was a recipient of the IEEE International Conference on Communications (IEEE ICC) "Best Paper Award" (2014), IEEE CIS "Outstanding Early Career Award" (2014), National Science Foundation "Faculty Early Career Development (CAREER) Award" (2011), and Providence Business News (PBN) "Rising Star Innovator" Award (2011).

Jiao Licheng
Xidian University, CHINA

For contributions to artificial neural networks and evolutionary computation.

Licheng Jiao received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982 and the M.S. and Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. Since 1992, he has been a Professor with the School of Electronic Engineering, Xidian University, Xi'an, where he is currently the Director of Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China.

In 1992, Dr. Jiao was awarded the Youth Science and Technology Award. In 1996, he was granted by the Cross-century Specialists Fund from the Ministry of Education of China. And he was selected as a member of the First level of Millions of Talents Project of China from 1996. In 2006, he was awarded the First Prize of Young Teacher Award of High School by the Fok Ying Tung Education Foundation. From 2006, he was selected as an Expert with the Special Contribution of Shaanxi Province. Dr. Jiao is a member of the IEEE Xi'an Section Execution Committee, the Chairman of the Awards and Recognition Committee and the Chairman of Computational Intelligence Society, the Chairman of IET Xi'an Section, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a committee member of the Chinese Committee of Neural Networks, an expert of the Academic Degrees Committee of the State Council, an Associate Editor of *IEEE Trans. on Geoscience and Remote Sensing*, and the Chairman of Xi'an Chapter of IEEE Geoscience and Remote Sensing Society. He has published more than 20 monographs and a hundred papers in international journals and conferences. His research interests include artificial neural networks, evolutionary computation, image processing, and intelligent information processing.

Jie Lu
University of Technology Sydney, AUSTRALIA

For contributions to fuzzy machine learning and decision support systems.

Jie Lu is a Distinguished Professor in the areas of fuzzy transfer learning, decision support systems, concept drift, and recommender systems. She is the Associate Dean in Research Excellence in the Faculty of Engineering and Information Technology at University of Technology Sydney (UTS), and the Director of Centre for Artificial Intelligence at UTS. She has published six research books and 400 papers in *Artificial Intelligence*, *IEEE Trans. on Fuzzy Systems*, *Decision Support Systems*, other refereed journals, and conference proceedings (H-index 44, Google Scholar). She has won more than 20 Australian Research Council (ARC) discovery and other research grants for over $4 million in the last 15years. She serves as Editor-in-Chief for *Knowledge-Based Systems* (Elsevier), Editor-in-Chief for *International Journal on Computational Intelligence Systems* (Atlantis), Associate Editor for *IEEE Trans. on Fuzzy Systems*, Editor for book series on Intelligent Information Systems (World Scientific), and has served as a guest editor of 12 special issues for IEEE transactions and other international journals. She has delivered 20 keynote speeches at international conferences, and has chaired 10 IEEE and other international conferences. She received the first UTS Research Excellence Medal for Teaching and Research Integration in 2010. She services as an ARC panel member (2016-2018). She is the Founding Chair of Australian NSW Computation Intelligence Chapter. She is a Fellow of IFSA (International Fuzzy Systems Association).

Jie Lu's outstanding and lasting contribution to computational intelligence focuses on integration of fuzzy techniques into machine learning and decision support systems. She has contributed to the development of theories and methods to cross-disciplinary research including fuzzy transfer learning, concept drift detection and adaptation, fuzzy classification, fuzzy recommender systems and fuzzy bi-level decision-making models and decision support systems. She has also pioneered real-world applications by applying computational intelligence techniques in e-government, e-business, logistics and customer retention.

Yew-Soon Ong
Nanyang Technological University, SINGAPORE

For contributions to memetic computation and applications.

Yew-Soon Ong is a Professor and Chair of the School of Computer Science and Engineering at Nanyang Technological University (NTU), Singapore. He is Founding Director of the Data Science and Artificial Intelligence Research Center (DSAIR), Founding Director of the A*Star SIMTECH-NTU Joint Lab on Complex Systems and Principal Investigator of the Data Analytics & Complex System Programme in the Rolls-Royce@NTU Corporate Lab. He received his Ph.D.

from University of Southampton, United Kingdom, and has held visiting appointments at Massachusetts Institute of Technology and Honda Research Institute Europe.

Dr. Ong is founding Editor-In-Chief of the *IEEE Trans. on Emerging Topics in Computational Intelligence*, founding Technical Editor-In-Chief of *Memetic Computing* and serves as Associate Editor of the *IEEE Trans. on Evolutionary Computation*, *IEEE Trans. on Neural Network and Learning Systems*, *IEEE Trans. on Cybernetics*, and *IEEE Trans. on Big Data*. His research interests in computational intelligence span across memetic computation, data-centric evolutionary optimization, and machine learning. Dr. Ong's research has contributed to the academic advancement of computational intelligence, particularly evolutionary memetic computation, earning him the recognition as a Thomson Reuters Highly Cited Researcher and cited amongst the World's Most-Influential-Scientific Minds in the field of Computer Science. He received the 2015 IEEE Computational Intelligence Magazine Outstanding Paper Award and the 2012 IEEE Transactions on Evolutionary Computation Outstanding Paper Award for his work in memetic computation. Several of his research technologies have been commercialized and licensed to companies and institutions. One of which led to the AI-enabled IOS 'Dark-Dots Game'. It emerged as the top action game in 48 countries including USA, China and Singapore; downloaded by well over 448,000 players worldwide when launched.

At the IEEE Computational Intelligence Society, he chaired the Intelligent Systems Applications Technical Committee (2013-2014) and the Emergent Technology Technical Committee (2011-2012). He has also served as Conference Chair of the IEEE Congress on Evolutionary Computation, Vancouver, Canada, 2016.

Hava Siegelmann
University of Massachusetts
Amherst, USA

For contributions to neural computation.

Dr. Hava Siegelmann is a Full Professor at the highly regarded College of Computer Science at the University of Massachusetts Amherst and director of the BINDS Lab. Her research focuses on advancing the state of neural networks and on the emerging field of Lifelong Learning, which is at the cutting edge of Machine Learning and Artificial Intelligence. She is currently on a leave of absence to initiate and direct DARPA's L2M lifelong learning program to develop computational systems capable of true learning, and applying prior learning to novel circumstances without retraining.

Dr. Siegelmann received her Ph.D. (Fellow of Excellence) in Computer Science from Rutgers (1993) with her ground-breaking thesis, "Foundation of Recurrent Neural Networks;" Master's (Cum Laude) Hebrew University (1992); BSc Technion (Summa Cum Laude) (1988). Siegelmann's seminal paper: Computation Beyond the Turing Limit, *Science* (1995), and subsequent book: *Neural Networks and Analog Computation: Beyond the Turing Limit* (1998), outlined her Super-Turing theory, an entirely new field of computation, the only known alternative to Turing computation, and now, a critical element in Lifelong Learning system development. Siegelmann, with Vladimir Vapnik, developed Support Vector Clustering—one of the most widely employed hierarchical clustering algorithms. Siegelmann is one of the few scientists to have successfully applied neural networks to complex, real-world applications as with her radar, and nuclear power plant control systems. Her findings are widely cited in textbooks and papers, and taught in curricula as foundational to the latest generation of AI and ML.

Dr. Siegelmann has held visiting appointments at MIT, Harvard, ETH Zurich, UC Berkeley, Cambridge University, Salk Institute, Bell Labs, NEC, the Weizmann Institute and more. In 2015 the NSF named Siegelmann one of 16 presidential BRAIN Initiative awardees; In 2016, the International Neural Network Society (INNS) named her the Donald O. Hebb Awardee, and IEEE named her a Distinguished Lecturer. Dr. Siegelmann has served extensively on the executive boards, and numerous committees for IEEE and INNS. She has chaired the 2011 IJCNN and others. She has given plenary and keynote talks in over 30 international conferences and has served as a longtime Editor at various journals including *Frontiers in Computational Neuroscience* (*Nature*), *Neural Networks*, and *Scholarpedia*. Dr. Siegelmann remains highly active in supporting young researchers and minorities. She also has years of experience consulting with industry, creating educational programs, fundraising, and in administration and organization.

David Fogel
Natural Selection, Inc., USA

Piero P. Bonissone
Piero P. Bonissone Analytics LLC, USA

Conference Report on 2017 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2017)

Honolulu, Hawaii (USA) was the host site for the 2017 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2017), held November 27 to December 1, 2017. This event marked the 10th anniversary of the first SSCI, also held in Honolulu (founded by David Fogel). Over the years, SSCI has grown into a flagship annual international conference on computational intelligence sponsored by the IEEE Computational Intelligence Society (IEEE CIS). The event differs from traditional IEEE CIS events in that multiple symposia across a wide variety of disciplines in computational intelligence are held concurrently. This allows a registrant to attend diverse technical sessions in many different topic areas.

SSCI 2017 received 775 submissions in total from 64 countries around the globe. After a peer review process, 494 papers were accepted for presentation. We were pleased to host more than 500 participants at the Hilton Hawaiian Village Resort as part of this very successful meeting, which included many student travel grant recipients, sponsored by the IEEE CIS.

In all, there were more than 30 symposia, with many featuring keynote speakers and special sessions. In addition, free tutorials were presented across all the main areas of computational intelligence, including neural networks, fuzzy logic, evolutionary computation, swarm

IEEE SSCI 2017 participants were treated to a beautiful rainbow to start the conference at the Hilton Hawaiian Village Resort.

The poster session included approximately 80 papers across all 30+ symposia.

Digital Object Identifier 10.1109/MCI.2018.2806986
Date of publication: 10 April 2018

Friday night fireworks concluded the IEEE SSCI 2017 event.

David Fogel (co-chair, left), Shumeet Baluja (one of two poster paper winners), and Piero Bonissone (co-chair, right).

David Fogel (co-chair, left), Christian Rodriguez (one of two poster paper winners), and Piero Bonissone (co-chair, right).

David Fogel (co-chair, left), Prof. Chang-Shing Lee (National University of Tainan, Taiwan, center), and Piero Bonissone (co-chair, right) take a break at the IEEE SSCI 2017 meeting.

intelligence, and complex adaptive systems, even including the physics of the mind. The SSCI event also featured a dynamic poster session, with two awards for the best poster papers, as well as a reception and banquet with Hawaiian music. In addition, the IEEE CIS Women in Computational Intelligence subcommittee organized a reception for women, students, and young professionals, which was attended by about 75 participants, including members of the SSCI organizing committee and IEEE CIS Administrative Committee who were happy to serve as mentors.

We'd like to thank each of the symposium organizers for their efforts to ensure a successful series of meetings. We'd also like to thank Carlos Coello Coello, Hussein Abbass, Haibo Hai, KC Tan, Simon Lucas, and Gary Yen for their assistance as members of the organizing committee. Mostly, we want to thank all of the participants for helping make it a great and fun event!

Haibo He, Jon Garibaldi, Kay Chen Tan,
Graham Kendall, Yaochu Jin, and
Yew Soon Ong

CIS Publication Spotlight

IEEE Transactions on Neural Networks and Learning Systems

LSTM: A Search Space Odyssey, by K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 10, October 2017, pp. 2222–2232.

Digital Object Identifier: 10.1109/TNNLS.2016.2582924

"Several variants of the long short-term memory (LSTM) architecture for recurrent neural networks have been proposed since its inception in 1995. In recent years, these networks have become the state-of-the-art models for a variety of machine learning problems. This has led to a renewed interest in understanding the role and utility of various computational components of typical LSTM variants. In this paper, we present the first large-scale analysis of eight LSTM variants on three representative tasks: speech recognition, handwriting recognition, and polyphonic music modeling. The hyperparameters of all LSTM variants for each task were optimized separately using random search, and their importance was assessed using the powerful functional ANalysis Of VAriance framework. In total, we summarize the results of 5400 experimental runs (≈15 years of CPU time), which makes our study the largest of its kind on LSTM networks. Our results show that none of the variants can improve upon the standard LSTM architecture significantly, and demonstrate the forget gate and the output activation function to be its most critical components. We further observe that the studied hyperparameters are virtually independent and derive guidelines for their efficient adjustment."

Broad Learning System: An Effective and Efficient Incremental Learning System Without the Need for Deep Architecture, by C. L. P. Chen and Z. Liu, *IEEE Transactions on Neural Networks and Learning Systems,* Vol. 29, No. 1, January 2018, pp. 10–24.

Digital Object Identifier: 10.1109/TNNLS.2017.2716952

"Broad Learning System (BLS) that aims to offer an alternative way of learning in deep structure is proposed in this paper. Deep structure and learning suffer from a time-consuming training process because of a large number of connecting parameters in filters and layers. Moreover, it encounters a complete retraining process if the structure is not sufficient to model the system. The BLS is established in the form of a flat network, where the original inputs are transferred and placed as "mapped features" in feature nodes and the structure is expanded in wide sense in the "enhancement nodes." The incremental learning algorithms are developed for fast remodeling in broad expansion without a retraining process if the network deems to be expanded. Two incremental learning algorithms are given for both the increment of the feature nodes (or filters in deep structure) and the increment of the enhancement nodes. The designed model and algorithms are very versatile for selecting a model rapidly. In addition, another incremental learning is developed for a system that has been modeled encounters a new incoming input. Specifically, the system can be remodeled in an incremental way without the entire retraining from the beginning. Satisfactory result for model reduction using singular value decomposition is conducted to simplify the final structure. Compared with existing deep neural networks, experimental results on the Modified National Institute of Standards and Technology database and NYU NORB object recognition dataset benchmark data demonstrate the effectiveness of the proposed BLS."

IEEE Transactions on Fuzzy Systems

Improving Supervised Learning Classification Methods Using Multigranular Linguistic Modeling and Fuzzy Entropy, by J. A. Morente-Molinera, J. Mezei, C. Carlsson, and E. Herrera-Viedma, *IEEE Transactions on Fuzzy Systems,* Vol. 25, No. 5, October 2017, pp. 1078–1089.

Digital Object Identifier: 10.1109/TFUZZ.2016.2594275

"Obtaining good classification results using supervised learning methods is critical if we want to obtain a high level of precision in the classification processes. The training data used for the learning

Digital Object Identifier 10.1109/MCI.2018.2806987
Date of publication: 10 April 2018

process play a very important role in achieving this objective. Therefore, it is important to represent the data in a way that best expresses its meaning. For this purpose, the authors propose to apply linguistic modeling methods in order to obtain a linguistic representation. With the help of multigranular linguistic modeling, data can be transformed and expressed using different (unbalanced) linguistic label sets. Expressing the data using linguistic expressions instead of numbers increases the readability and reduces the complexity of the problem, and data recovering methods allow us to manually control the level of precision. In this paper, several datasets are transformed and utilized for classification tasks using several supervised learning algorithms. For each combination of datasets and algorithms, the data has been expressed using several linguistic label sets that have different granularity values. After carrying out the testing processes, they can conclude that, in some cases, reducing data complexity leads to better classification results. Therefore, it is found that linguistic representation of the training data with just the necessary and sufficient precision can improve the reliability of the classification process."

Efficient Multiple Kernel Classification Using Feature and Decision Level Fusion, by A. J. Pinar, J. Rice, L. Hu, D. T. Anderson, and T. C. Havens, *IEEE Transactions on Fuzzy Systems,* Vol. 25, No. 6, December 2017, pp. 1403–1416.

Digital Object Identifier: 10.1109/TFUZZ.2016.2633372

"Kernel methods for classification is a well-studied area in which data are implicitly mapped from a lower-dimensional space to a higher dimensional space to improve classification accuracy. However, for most kernel methods, one must still choose a kernel to use for the problem. Since there is, in general, no way of knowing which kernel is the best, multiple kernel learning (MKL) is a technique used to learn the aggregation of a set of valid kernels into a single (ideally) superior kernel. The aggregation can be done using weighted sums of the precomputed kernels, but determining the summation weights is not a trivial task. Furthermore, MKL does not work well with large datasets because of limited storage space and prediction speed. In this paper, the authors address all three of these multiple kernel challenges. First, they introduce a new linear feature level fusion technique and learning algorithm, GAMKLp. Second, they put forth three new algorithms, DeFIMKL, DeGAMKL, and DeLSMKL, for nonlinear fusion of kernels at the decision level. To address MKL's storage and speed drawbacks, they apply the Nystrom approximation to the kernel matrices. The authors compare their methods to a successful and state-of-the-art technique called MKL-group lasso (MKLGL), and experiments on several benchmark datasets show that some of their proposed algorithms outperform MKLGL when applied to support vector machine (SVM)-based classification. However, to no surprise, there does not seem to be a global winner but instead different strategies that a user can employ. Experiments with their kernel approximation method show that they can routinely discard most of the training data and at least double prediction speed without sacrificing classification accuracy. These results suggest that MKL-based classification techniques can be applied to big data efficiently, which is confirmed by an experiment using a large dataset."

IEEE Transactions on Evolutionary Computation

DG2: A Faster and More Accurate Differential Grouping for Large-Scale Black-Box Optimization, by M. N. Omidvar, M. Yang, Y. Mei, X. Li, and X. Yao, *IEEE Transactions on Evolutionary Computation,* Vol. 21, No. 6, December 2017, pp. 929–942.

Digital Object Identifier: 10.1109/TEVC.2017.2694221

"Identification of variable interaction is essential for an efficient implementation of a divide-and-conquer algorithm for large-scale black-box optimization. In this paper, an improved variant of the differential grouping (DG) algorithm is proposed, which has a better efficiency and grouping accuracy. The proposed algorithm, DG2, finds a reliable threshold value by estimating the magnitude of roundoff errors. With respect to efficiency, DG2 reuses the sample points that are generated for detecting interactions and saves up to half of the computational resources on fully separable functions. It is mathematically showed that the new sampling technique achieves the lower bound with respect to the number of function evaluations. Unlike its predecessor, DG2 checks all possible pairs of variables for interactions and has the capacity to identify overlapping components of an objective function. On the accuracy aspect, DG2 outperforms the state-of-the-art decomposition methods on the latest large-scale continuous optimization benchmark suites. DG2 also performs reliably in the presence of imbalance among contribution of components in an objective function. Another major advantage of DG2 is the automatic calculation of its threshold parameter (ϵ), which makes it parameter-free. Finally, the experimental results show that when DG2 is used within a cooperative co-evolutionary framework, it can generate competitive results as compared to several state-of-the-art algorithms."

IEEE Transactions on Computational Intelligence and AI in Games

Creating AI Characters for Fighting Games Using Genetic Programming, by G. Martínez-Arellano, R. Cant, and D. Woods, *IEEE Transactions on Computational Intelligence and AI in Games,* Vol. 9, No. 4, December 2017, pp. 423–434.

Digital Object Identifier: 10.1109/TCIAIG.2016.2642158

"This paper proposes a character generation approach for the M.U.G.E.N. fighting game that can create engaging

AI characters using a computationally cheap process without the intervention of the expert developer. The approach uses a genetic programming algorithm that refines randomly generated character strategies into better ones using tournament selection. The generated AI characters were tested by 27 human players and were rated according to results, perceived difficulty and how engaging the gameplay was. The main advantages of this procedure are that no prior knowledge of how to code the strategies of the AI character is needed and there is no need to interact with the internal code of the game. In addition, the procedure is capable of creating a wide diversity of players with different strategic skills, which could be potentially used as a starting point to a further adaptive process."

IEEE Transactions on Cognitive and Developmental Systems

Deep Reinforcement Learning With Visual Attention for Vehicle Classification, by D. Zhao, Y. Chen, and L. Lv, *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 9, No. 4, December 2017, pp. 356–367.

Digital Object Identifier: 10.1109/TCDS.2016.2614675

"Automatic vehicle classification is crucial to intelligent transportation system, especially for vehicle-tracking by police. Due to the complex lighting and image capture conditions, image-based vehicle classification in real-world environments is still a challenging task and the performance is far from being satisfactory. However, owing to the mechanism of visual attention, the human vision system shows remarkable capability compared with the computer vision system, especially in distinguishing nuances processing. Inspired by this mechanism, we propose a convolutional neural network (CNN) model of visual attention for image classification. A visual attention-based image processing module is used to highlight one part of an image and weaken the others, generating a focused image. Then the focused image is input into the CNN to be classified. According to the classification probability distribution, we compute the information entropy to guide a reinforcement learning agent to achieve a better policy for image classification to select the key parts of an image. Systematic experiments on a surveillance-nature dataset which contains images captured by surveillance cameras in the front view, demonstrate that the proposed model is more competitive than the large-scale CNN in vehicle classification tasks."

IEEE Transactions on Emerging Topics in Computational Intelligence

A Strategy for Self-Organized Coordinated Motion of a Swarm of Minimalist Robots, by A. R. Shirazi and Y. Jin, *IEEE Transactions on Emerging Topics in Computational Intelligence,* Vol. 1, No. 5, October 2017, pp. 326–338.

Digital Object Identifier: 10.1109/TETCI.2017.2741505

"Minimalist robots are functionally highly restricted but well suited for swarm robotic applications because of their low costs and small sizes. Connectivity maintenance and collision avoidance are challenging in minimalist swarm robotic systems due to a short communication range and the lack of positional and directional sensing. In this paper, we introduce a self-organizing control strategy for collective flocking of a swarm of minimalist robots with an aim to improve swarm connectivity and to reduce the chance of collision between robots. Based on the relative positional information built up via collaborations, each robot determines a collision-free operational polygon. This scheduling scheme coordinates the motion of the robots by dividing them into one group of immobile and one group of mobile robots, such that each mobile robot is surrounded by immobile robots serving as beacons. In addition, we introduce a cohesive force into motion planning, which has been shown to play an important role in maintaining a swarm during flocking. A new quantitative metric is introduced for measuring the connectivity of a swarm of agents with local communications, thereby, evaluating the performance of the proposed control scheme. We run extensive simulations using simulated Kilobots to examine the influence of different sources of noise and the size of swarms on the connectivity in the swarm and the speed of flocking. Finally, we implement the proposed algorithm on a swarm of real Kilobots to compare the flocking performance with and without the proposed control strategy for coordinated and collective motion."

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.

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.

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.

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.

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.

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 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.

MC^{2}ESVM: Multiclass Classification Based on Cooperative Evolution of Support Vector Machines

Alejandro Rosales-Pérez and Hugo Terashima-Marin
School of Engineering and Sciences,
Tecnologico de Monterrey, MEXICO

Salvador García and Francisco Herrera
Department of Computer Science
and Artificial Intelligence,
University of Granada, SPAIN

Carlos A. Coello Coello
Department of Computer Science,
CINVESTAV-IPN, MEXICO

IMAGE LICENSED BY INGRAM PUBLISHING

Abstract—Support vector machines (SVMs) are one of the most powerful learning algorithms for solving classification problems. However, in their original formulation, they only deal with binary classification. Traditional extensions of the binary SVMs for multiclass problems are based either on decomposing the problem into a number of binary classification problems, which are then independently solved, or on reformulating the objective function by solving larger optimization problems. In this paper, we propose MC^{2}ESVM, an approach for multiclass classification based on the cooperative evolution of SVMs. Cooperative evolution allows us to decompose an *M*-class problem into *M* subproblems, which are simultaneously optimized in a cooperative fashion. We have reformulated the optimization problem such that it focuses on learning the support vectors for each class at the time that it takes into account the information from other classes. A comprehensive experimental study using common benchmark datasets is carried out to validate MC^{2}ESVM. The experimental results, supported by statistical tests, show the effectiveness of MC^{2}ESVM for solving multiclass classification problems, while keeping a reasonable number of support vectors.

Digital Object Identifier 10.1109/MCI.2018.2806997
Date of publication: 10 April 2018

Corresponding Author: Alejandro Rosales-Pérez (Email: arosalesp85@gmail.com)

1556-603X/18©2018IEEE

I. Introduction

Support Vector Machines (SVMs) [1] are powerful supervised learning algorithms with strong theoretical foundations that have shown a high performance over a wide range of problems [2]–[4]. The main idea behind SVMs is to find the hyperplane that maximizes the separation between two classes, which is defined through the so-called *support vectors*. In spite of the effectiveness of SVMs in solving binary classification problems, real world problems often require discriminating among more than two classes.

Over the last years, there has been interest in extending SVMs to multiclass problems. These approaches can be differentiated into two major groups: (1) decomposition strategies and (2) single machine methods. The first type of approach is based on decomposing the *M*-class problem into several binary classification problems. The most well-known decomposition techniques are the one-vs-one (OVO) and the one-vs-all (OVA) methods. These have been found to be quite effective in solving multiclass problems [5], [6]. However, they assume that each binary classification problem to be solved is independent of the rest.

On the other hand, single machine approaches are based on modifying the optimization problem, such that the multiclass SVM classifier is constructed based on solving a single optimization problem [7]. Nonetheless, they have the shortcoming of dealing with a more complex and larger optimization problem.

Evolutionary algorithms (EAs) encompass a family of algorithms that aim at solving complex optimization problems. EAs have been applied with success to the solution of different machine learning problems [8]–[10]. In recent years, several studies that hybridize EAs with SVMs have been reported [11]–[13]. Most of them deal with the hyper-parameter optimization problem. There are only a few attempts to deal with the parameter optimization problem, such as those reported in [14], [15]. They have, however, only focused on the classical binary classification problems.

This paper introduces MC^{2}ESVM (Multiclass Classification based on the Cooperative Evolution of SVMs). MC^{2}ESVM aims at taking advantage of the benefits of both decomposition and single machine approaches, by decomposing the multiclass problem and solving the resulting problems as single-objective optimization problems, optimizing the support vector for each class. This can be approached in a natural fashion with cooperative coevolutionary algorithms. Moreover, the inherent advantages of evolutionary algorithms allow MC^{2}ESVM to handle non-positive semidefinite kernels[1]. The main contributions of this paper are the following:

- ❑ The decomposition of the multiclass problem via coevolutionary optimization. This allows SVMs to be able to learn multiclass classifiers in a single optimization run by simultaneously solving a set of simpler problems. To the best of the authors' knowledge, this is the first attempt to combine coevolutionary algorithms with SVMs for multiclass problems.
- ❑ A derivation of the optimization problem that learns the class-specific support vectors, considering the information from other classes.

The performance of MC^{2}ESVM is assessed using a suite of 25 multiclass classification datasets. We first compare it with state-of-the-art SVMs extensions in terms of the prediction performance and common learning algorithms. Second, we compare with respect to the support vectors. Afterwards, we assess its scalability as either the number of instances or classes are increased. Finally, we assess the stability of the algorithm and the evolutionary parameters. Our experimental results show the effectiveness of MC^{2}ESVM for solving the classification task, while keeping a reasonable number of support vectors. These findings are supported by a set of non-parametric tests.

The remainder of this paper is organized as follows. Section II describes some preliminary concepts related to the main extension to multiclass SVM and coevolutionary optimization. Section III describes in detail our proposed MC^{2}ESVM. Next, Section IV outlines the experimental settings for our study, while Section V presents the experimental results and the statistical validation. Finally, Section VI provides our general conclusions.

II. Preliminaries

This section discusses the main preliminaries in which our contribution is based. Section II-A describes the main extensions proposed to solve multiclass problems using SVMs. Next, in Section II-B, we describe the main characteristics of coevolutionary algorithms.

A. Multiclass Extensions for SVMs

A number of approaches for extending SVMs so that they can handle multiclass problems have been proposed. Fig. 1 shows the proposed methods to approach multiclass problems with SVMs. They are briefly discussed next.

1) Decomposition Strategies

These approaches follow the idea of dividing the multiclass problems into several binary classification problems. The most common decomposition methods for multiclass SVMs are the following:

- ❑ ***One-vs-All*** (OVA) [16]: OVA decomposes the *M*-class problem into *M* subproblems. *M*-binary SVMs are constructed

[1] Non-positive semidefinite kernels can lead to a non-convex optimization for SVMs.

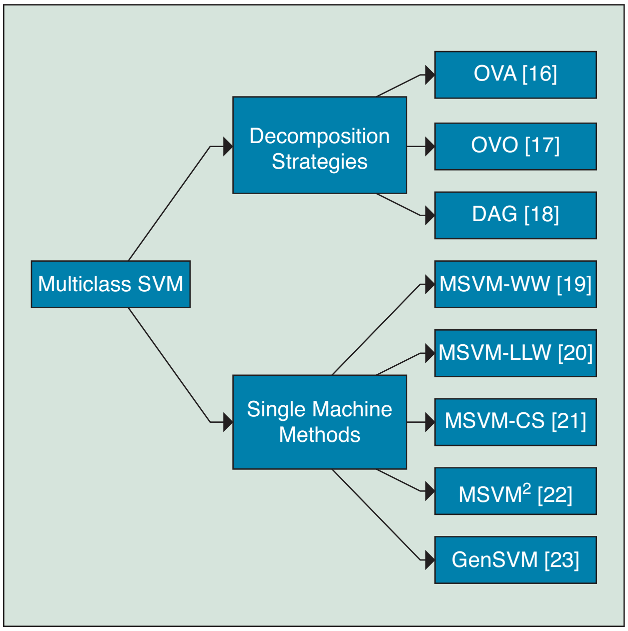


FIGURE 1 SVMs extensions for handling multiclass problems.

for each subproblem, such that the *i*th SVM is trained using the samples belonging to the *i*th class as positive samples and the remaining are treated as negative samples. A new sample is assigned to the class with the largest activation value. OVA introduces an artificial imbalance during the training. Thus, the higher the value of *M*, the higher the imbalance rate.

- ***One-vs-One*** (OVO) [17]: In OVO, an SVM is trained for each possible pair of classes, resulting in a total of $M(M-1)/2$ SVMs. This number is usually larger than the one of the OVA approach. In the prediction phase, a new sample is classified for each SVM and the class with the majority vote wins. The main criticism of OVO is that when M is large, the evaluation of the $M(M-1)/2$ SVMs can slow down the prediction stage of the resulting OVO.
- ***Directed Acyclic Graph*** (DAG) [18]: The training phase is similar to OVO, resulting in $M(M-1)/2$ SVMs constructed for each pair of classes. The difference relies on the prediction stage. DAG starts at the root, where an SVM is used to classify the test sample, and it moves either to the left or to the right path, depending on the predicted class given by the SVM. This process is repeated until a leaf node is reached, which indicates the predicted class. Note that, however, the performance of DAG depends on the SVM at the root node.

A comparison between these three strategies is performed in [24], finding that their accuracy is quite similar, with no statistical difference.

2) Single Machine Methods

These methods aim at solving directly the multiclass problem during the training phase. This is attained by modifying the SVM objective function, such that it simultaneously allows computing the multiclass classifier. For instance, in [19], authors propose MSVM-WW, where the single objective formulation for the multiclass SVM is given as follows:

$$\min_{\mathbf{w}_r,\xi^r,b_r}\frac{1}{2}\sum_{r=1}^{M}\|\mathbf{w}_r\|^2+\frac{C}{N}\sum_{i=1}^{N}\sum_{q}\xi_i^q$$

$$\text{subject to}\langle\mathbf{w}_r,\mathbf{x}_i\rangle+b_r\geq\langle\mathbf{w}_q,\mathbf{x}_i\rangle+b_q+2-\xi_i^q,\xi_i^r\geq 0, \qquad (1)$$

where $q=\{1,\ldots,M\}\backslash r$, N is the number of training samples, and C is a penalty parameter that controls the trade-off between accuracy and complexity.

This formulation, however, has to deal with a large number of slack variables. Other formulations of the objective function are MSVM-LLW [20], which reduces the dimensionality of the problem by means of a sum-to-zero constraint; MSVM-CS [21], which only takes into account the largest activation and the bias term is not considered during the training; MSVM^2 [22], which adds a quadratic function to the slack variables; and GenSVM [23], which uses a simplex encoding to reduce the dimensionality of the problem. These approaches have reported similar performance to those obtained by either OVA or OVO. Nonetheless, these methods have the disadvantage of dealing with larger optimization problems.

B. Coevolutionary Optimization

A coevolutionary algorithm is an evolutionary algorithm which is able to manage two or more populations simultaneously [25]. An important characteristic of these algorithms is that they allow to split the problem into different parts and assign a different population to each subproblem. Each population focuses its efforts on solving one specific part of the problem. Two different kinds of coevolutionary algorithms can be described:

- **Competitive coevolutionary algorithms** [26]. The individuals of each population compete against each other, such that the fitness value of an individual decreases as the result of an increment in the fitness value of its adversaries. Competitive coevolution is normally adopted for game-like problems.
- **Cooperative coevolutionary algorithms** [27]. Each population evolves individuals representing a part of the solution. A complete solution is composed by joining individuals from all the populations. Therefore, the fitness value of an individual is the result of its collaboration with other individuals from other populations.

In this work, our focus is on cooperative coevolution, due to the fact that it allows us to decompose the multiclass classification problem in a natural fashion, by assigning to each subproblem the task of learning the set of support vectors for each class.

III. MC^{2}ESVM: Multiclass Classification Based on Cooperative Evolution of Support Vector Machines

The proposed MC^{2}ESVM aims at training a multiclass SVM in a single step. The multiclass classifier is defined by the set of support vectors of each class. MC^{2}ESVM is based on the cooperative coevolutionary algorithm, in which each subpopulation optimizes the support vectors for each class at the same time that it considers the other subpopulations for solving the multiclass problem. Algorithm 1 describes MC^{2}ESVM. Generally, it follows these steps:

1) In line 1, for each class, a population is randomly created. The number of variables for each population depends on the number of samples in the training set for the given class.
2) In lines from 3 to 5, the fitness value is assigned for each individual of each class (population). For doing so, an individual from other classes is randomly selected to build the multiclass classifier. This is part of the cooperative coevolution.
3) Lines from 7 to 16 are the evolutionary process, as follows:
   a) Line 9 selects the best individual from other classes and the evolutionary operators are applied to create an offspring, in line 11.
   b) Line 12 computes the fitness value of the offspring solution by concatenating it with the best solutions from other classes.
   c) In line 13, the best solutions among the parents and offspring are selected to be included in the next generation.
4) Once the evolutionary process is over, the final solutions, i.e., the support vectors for the multiclass problem are obtained from the best individuals of each class; this is done in line 17.

The details of MC^{2}ESVM are given in the remainder of this section. First, in Section III-A, we explain the optimization problem for finding the support vectors for a multiclass problem. Next, Section III-B presents the representation adopted in the evolutionary optimization as well as the evolutionary operators. Section III-C describes the final steps to construct the multiclass SVM and in Section III-D, we discuss the extension to learn nonlinear functions.

A. Fitness Functions: Optimization Problem

In MC^{2}ESVM, each subproblem aims at learning the support vector for a given class label. Therefore, the optimization problem for the rth class is formulated as follows:

$$\min_{\mathbf{w}_r,\xi^r} \frac{1}{2}\|\mathbf{w}_r\|^2 + C\sum_{i=1}^{n_r}\xi_i^r$$
$$\text{subject to } \langle \mathbf{w}_r, \mathbf{x}_i\rangle \geq 1-\xi_i^r,\ \xi_i^r \geq 0 \tag{2}$$

where n_r is the number of samples for the rth class.

For the sake of simplicity, we have omitted the bias term in our formulation. Moreover, since a population is focused on learning the support vectors for a single class, the class label is not part of Equation 2. This leads to a simpler dual problem, with no additional constraints. For the dual formulation, the constraint is incorporated in the objective function by using the Lagrange multipliers:

$$\mathcal{L}(\mathbf{w},\xi^r,\alpha_i^r,\beta_i^r) = \frac{1}{2}\|\mathbf{w}_r\|^2 + C\sum_{i=1}^{n_r}\xi_i^r$$
$$-\sum_{i=1}^{n_r}\alpha_i^r[\langle \mathbf{w}_r,\mathbf{x}_i\rangle - 1 + \xi_i^r] - \sum_{i=1}^{n_r}\beta_i^r\xi_i^r$$
$$\text{subject to } \alpha_i^r, \beta_i^r \geq 0. \tag{3}$$

Algorithm 1 **MC^{2}ESVM.**

```
Require: $\mathcal{X}$, the set of samples,
         $\mathcal{Y}$, the set of classes labels,
         $C$, the regularization term,
         $P$, the population size,
         $E$, maximum number of evaluations.
Ensure: The set of support vectors
 1: Generate randomly an initial population, $\mathcal{P}_y$ for each
    class $y \in \mathcal{Y}$
 2: for each $y \in \mathcal{Y}$ do
 3:   Select randomly an individual from each class $y' \in \mathcal{Y}\backslash y$
 4:   Construct full solutions by combining the selected individuals of each class
 5:   Evaluate the full solutions using the fitness function
 6: end for
 7: while a stopping criterion is not met do
 8:   for each $y \in \mathcal{Y}$ do
 9:     Select the best individual from each population
        $y' \in \mathcal{Y}\backslash y$
10:     for each individual in the current class do
11:       Apply evolutionary operators to create an offspring
12:       Evaluate the offspring with the fitness function
13:       Add the offspring to the next generation if it
          improves its parents
14:     end for
15:   end for
16: end while
17: Construct the final solution based on the best individuals
    of each population
```

Setting the partial derivatives to zero gives:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_r} = 0 \rightarrow \mathbf{w}_r = \sum_{i=1}^{n_r}\alpha_i^r\mathbf{x}_i \tag{4}$$
$$\frac{\partial\mathcal{L}}{\partial\xi_i^r} = 0 \rightarrow \beta_i^r = C - \alpha_i^r. \tag{5}$$

Substituting Equations 4 and 5 in Equation 3, gives the following dual optimization problem:

$$\min_{\alpha_i^r} \frac{1}{2}\sum_{i,j=1}^{n_r}\alpha_i^r\alpha_j^r\langle \mathbf{x}_i,\mathbf{x}_j\rangle - \sum_{i=1}^{n_r}\alpha_i^r$$
$$\text{subject to } 0 \leq \alpha_i^r \leq C. \tag{6}$$

At this point, Equation 6 learns the support vectors for a single class label. Since the goal of MC^{2}ESVM is to gain benefit from the cooperative evolution of each class label, an additional term is added to Equation 6 that considers the information from other classes. Thus, the optimization problem in cooperative evolution is stated as:

$$\min_{\alpha_i^r} \frac{1}{2}\sum_{i,j=1}^{n_r}\alpha_i^r\alpha_j^r\langle \mathbf{x}_i,\mathbf{x}_j\rangle - \sum_{i=1}^{n_r}\alpha_i^r + \frac{1}{n_r}\sum_{i=1}^{n_r}\mathcal{P}(\mathbf{x}_i)$$
$$\text{subject to } 0 \leq \alpha_i^r \leq C \tag{7}$$

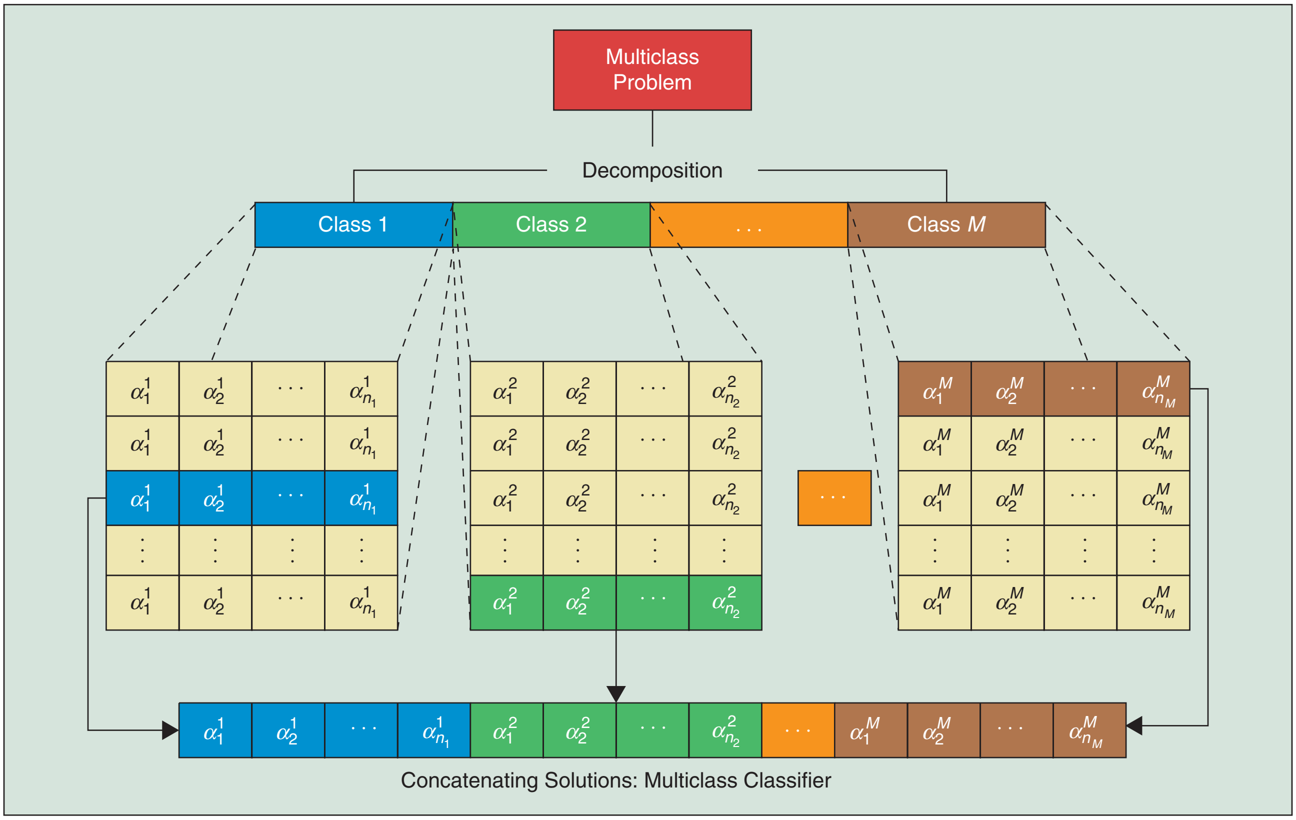

Multiclass Problem
Decomposition
Class 1
Class 2
. . .
Class M
α_1^1 α_2^1 · · · $\alpha_{n_1}^1$
α_1^2 α_2^2 · · · $\alpha_{n_2}^2$
α_1^M α_2^M · · · $\alpha_{n_M}^M$
Concatenating Solutions: Multiclass Classifier

FIGURE 2 Population scheme used in MC^{2}ESVM.

where

$$\mathcal{P}(\mathbf{x}) = \begin{cases} Z & \text{if } Z > 0 \\ 0 & \text{if } Z \le 0 \end{cases} : Z = 2 + \sum_{i=1}^{n_q} \alpha_i^q \langle \mathbf{x}_i, \mathbf{x} \rangle - \sum_{i=1}^{n_r} \alpha_i^r \langle \mathbf{x}_i, \mathbf{x} \rangle \qquad (8)$$

where $q \in \{1, \ldots, M\} \backslash r$ is the index class with the largest activation.

Thus, by adding this term, we penalize the errors that occur in the multiclass classification. In other words, MC^{2}ESVM punishes those solutions that do not work well together.

In the next section, we describe the representation of the solutions and the evolutionary operators used in MC^{2}ESVM.

B. Representation

In MC^{2}ESVM, each class works with the others in a cooperative fashion. As we have previously mentioned, each population manages the instances for a specific class. A representation scheme of the population is shown in Fig. 2.

Each population consists of P individuals. All populations share the same individual representation. Since the goal is to optimize the α vector from Equation 7, a real-valued representation is adopted. Moreover, the number of variables of each population depends on the number of instances available in the training set for each class[2]. By using this representation, the number of variables is not increased in the optimization task, as usually happens with other methods. Moreover, all populations are evolved simultaneously and each of them deals with simpler problems.

The α vector of each individual in each population is randomly initialized. For doing so, each variable of an individual has a probability of 0.5 to take a value in the range $(0, C]$; otherwise, it takes a value of 0.

The individuals in each population are evolved. This is attained by using the differential evolution operator [28], which generates a new child solution as follows:

$$\bar{\alpha}_i^{r(s)} = \begin{cases} \alpha_i^{r(t)} + F \times (\alpha_i^{r(u)} - \alpha_i^{r(v)}) & \text{with prob. } CR, \\ \alpha_i^{r(s)} & \text{Otherwise} \end{cases} \qquad (9)$$

where CR and F are two control parameters and s, t, u and v are the indexes for the current individual, which acts as the parent solution, and three randomly selected individuals from the rth population.

Finally, the child solution is added to the population for the next generation in the evolutionary process if and only if it improves the sth parent; otherwise, the current parent is kept.

C. Building the Multiclass SVM

Once the coevolutionary process is over, the next step is to build the multiclass classifier. This is done by selecting from each population the member that gets the highest score in the

[2] For example, in a three-class problem, with 50 instances of class 1, 80 instances of class 2, and 100 instances of class 3; the number of variables to optimize in subpopulations 1, 2, and 3 are equal to 50, 80, and 100, respectively.

TABLE 1 Description of the datasets used in our study. For each dataset, we show the number of instances, the number of attributes, and the number of classes.

ID	DATASET	ATTS.	INSTS.	IR	CLASSES	ID	DATASET	ATTS.	INSTS.	IR	CLASSES
1	AUTOMOBILE	25	203	5.69	6	14	NEWTHYROID	5	215	3.48	3
2	BALANCE	4	625	4.25	3	15	PENBASED	16	10,992	1.05	10
3	CLEVELAND	13	303	3.87	5	16	SATIMAGE	36	6,435	1.73	6
4	CONTRACEPTIVE	9	1,473	1.55	3	17	SEGMENT	19	2,310	1.00	7
5	DERMATOLOGY	34	366	2.17	6	18	SPLICE	60	3,190	1.77	3
6	ECOLI	7	336	15.27	8	19	TAE	5	151	1.04	3
7	GLASS	9	214	3.60	6	20	TEXTURE	40	5,500	1.00	11
8	HAYES-ROTH	4	132	1.47	3	21	VEHICLE	18	846	1.05	4
9	IRIS	4	150	1.00	3	22	VOWEL	13	990	1.00	11
10	LED7DIGIT	7	500	1.16	10	23	WINE	13	178	1.30	3
11	LYMPHOGRAPHY	18	148	18.30	4	24	YEAST	8	1,484	11.65	10
12	MARKETING	13	8,993	1.48	9	25	ZOO	16	101	3.20	7
13	MOVEMENT LIBRAS	90	360	1.00	15						

objective function and by concatenating the solutions. This can also be shown in Fig. 2.

It is worth noting that solutions with $\alpha_i^r > 0$ are considered the support vectors. These solutions represent the multiclass classifier learned by MC^{2}ESVM. A new instance $\mathbf{x}_t$ is classified as follows:

$$y_t = \underset{r}{\operatorname{argmax}} \sum_{i \in SV_r} \alpha_i^r \langle \mathbf{x}_i, \mathbf{x}_t \rangle \qquad (10)$$

where SV_r represents the set of support vectors from the rth class.

D. Learning Nonlinear SVMs

The optimization problem presented in Equation 7 learns a linear function from the training data. For learning nonlinear functions, the so-called *kernel trick* is used with SVMs. By using the kernel trick in MC^{2}ESVM, the inner product, $\langle \mathbf{x}_i, \mathbf{x} \rangle$, in Equations 7 and 10, is replaced by a Kernel function, $K(\mathbf{x}_i, \mathbf{x})$. Some commonly used kernel functions are the following [29]:

❏ Linear kernel: $K(\mathbf{x}_i, \mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x} \rangle$

❏ Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}) = (\langle \mathbf{x}_i, \mathbf{x} \rangle + 1)^d$

❏ Radial basis function kernel: $K(\mathbf{x}_i, \mathbf{x}) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2}$

where d, γ are adjustable parameters for the above kernel functions.

IV. Experimental Setup

In this section, we describe the experimental settings in our study. In Section IV-A, we present the datasets used in our experimental study. Section IV-B describes the algorithms that are used to compare the performance of MC^{2}ESVM. Finally, Section IV-C presents the performance measures and statistical tests used to assess each algorithm.

A. Datasets

A set of 25 datasets available in the KEEL repository [30], [31] are used in our experimental study. Since we want to assess the behavior of the proposed MC^{2}ESVM in multiclass problems, the datasets have been chosen based on the number of classes, i.e., those with more than two classes. Table 1 shows some characteristics of these datasets, such as the number of instances, the number of features, the imbalance rate (IR)[3] and the number of classes.

These datasets have been partitioned into 10 training/test subsets by using the k-fold cross validation technique. Furthermore, features have been pre-processed in order to have zero mean and unit standard deviation.

B. Considered Algorithms

Several multiclass extensions are used to compare the performance of MC^{2}ESVM with respect to them. The selection of these extensions is based on their availability on public frameworks, such as KEEL and MSVMpack [7]. Concretely, these extensions are the following:

❏ Decomposition strategies using the sequential minimal optimization (SMO) [32] implemented in KEEL:
 —OVO —OVA

❏ Single machine methods available at MSVMpack:
 —MSVM^2 —MSVM-CS
 —MSVM-LLW —MSVM-WW

For all cases, a radial basis function (RBF) kernel is used, because it is one of the most popular and effective kernels used with SVMs [33]. Regarding the hyper-parameter configuration of each method, and for the sake of allowing a fair comparison, the values of the hyper-parameters are tuned for each method in each dataset, as it is suggested in [33], [34]. For doing so, we used a random search for the hyper-parameters optimization [35], by randomly sampling 100 points for the kernel's

[3] The IR is computed as the average of the IR of all pairwise classes.

parameter in the range $\gamma = [2^{-10}, 2^{2}]$, and for the regularization parameter $C = [2^{-4}, 2^{10}]$ for all methods. Each configuration is tested and the one with the best score in accuracy is chosen for each dataset.

It is worth noting that during the experiments, some configurations for the single machine methods available at MSVMpack were unable to converge to an optimal solution. Thus, we have modified the code in order to allow a maximum number of evaluations of the objective function, which was fixed to 200,000. This limit is also set for all methods. Furthermore, in the case of MC^{2}ESVM, the convergence criterion is defined as having an improvement in the best solution after 10 iterations lower than 0.001. MC^{2}ESVM also requires some additional parameters, which are common in evolutionary algorithms. These parameters were set as follows:

- Population size = 20
- Crossover Rate (CR) = 0.8
- Differential weight (F) = 1.6

We have further considered the set of common learning algorithms: Random Forest (RF), Multi-Layer Perceptron (MLP), K-Nearest Neighbor (KNN), and Naïve Bayes (NB), for comparison. To allow a fair comparison, the hyper-parameters of these methods are tuned. In RF, the number of trees is adjusted in the range $[1, 100]$; in MLP, the learning rate is tuned in the range $[0, 1]$ and the number of neurons $[1, 100]$; KNN, the neighborhood size is determined in $[1, 10]$.

C. Performance Measures

We have considered two different metrics for measuring the performance of MC^{2}ESVM and the reference methods. The set of metrics are accuracy and Cohen's kappa, which are described next:

- **Accuracy** (Acc) is a common metric for assessing the performance of supervised learning algorithms. It indicates the ratio of samples that are correctly classified, i.e.,

$$Acc = \frac{1}{N}\sum_{i=1}^{M} TP_i \tag{11}$$

  where TP_i is the number of correctly classified samples from class i.

- **Cohen's kappa** ($\mathcal{K}$) measures the degree of agreement between two observations: the predicted class and the correct one. An easy way of computing Cohen's kappa is as follows:

$$\mathcal{K} = \frac{N\sum_{i=1}^{M} TP_i - \sum_{i=1}^{M} P_i T_i}{N^2 - \sum_{i=1}^{M} P_i T_i} \tag{12}$$

  where P_i is the number of predicted samples as class i and T_i is the number of samples from class i.

  Cohen's kappa ranges from -1, indicating total disagreement, through 0 (random classification), to 1, which indicates a perfect agreement.

TABLE 2 Comparison between MC^{2}ESVM and the SVM formulations for multiclass problems.

METHOD	*ACC*	$\mathcal{K}$
MC^{2}ESVM	**0.8167 ± 0.1787**	**0.7424 ± 0.2437**
OVA	0.7920 ± 0.1897	0.7044 ± 0.2683
OVO	0.8003 ± 0.1868	0.7031 ± 0.2880
MSVM^2	0.7682 ± 0.1886	0.6648 ± 0.2697
MSVM-CS	0.7958 ± 0.1970	0.7157 ± 0.2651
MSVM-LLW	0.7795 ± 0.1967	0.6715 ± 0.2988
MSVM-WW	0.7808 ± 0.1949	0.6875 ± 0.2705

In order to support the comparisons, a set of non-parametric statistical tests is used. Non-parametric tests are widely recommended for a safe and robust comparison of multiple classifiers over multiple datasets by [36]–[38]. In this study, we have used the Friedman Aligned Ranks test to compare among multiple algorithms, and the Holm's procedure is used to find out which algorithms are distinctive. In all cases, the significance level is set to $\alpha = 0.05$. A description of these tests can be found in [37], [38].

V. Experimental Results and Their Analysis

This section presents the results obtained in our experimental study and analyzes them. In Section V-A, we report the results of MC^{2}ESVM and the reference methods using the set of multiclass datasets. Section V-B analyzes and compares the support vectors found by each method. Section V-C, we assess the scalability of the proposed MC^{2}ESVM with respect to the number of classes and the number of samples. Finally, in Section V-D we analyze the stability of MC^{2}ESVM.

A. Classification Performance

The aim of this study is twofold. First, comparing the performance of MC^{2}ESVM with respect to other SVM formulations for multiclass problems. Second, comparing against standard learning algorithms.

1) Comparing with SVMs Multiclass Extensions

In this first part of our study, we assess the performance of MC^{2}ESVM when it is compared with several extensions of SVMs for multiclass problems. Table 2[4] shows the average results obtained from the 25 datasets.

Table 3 shows the ranking obtained by Friedman's Aligned Ranks both with accuracy score (Acc) and Cohen's Kappa ($\mathcal{K}$). We further show the adjusted p-value with the Holm's test (p_{Holm}). Note that MC^{2}ESVM is set as a control method because the purpose of our study is to compare the performance of our proposal against the rest.

Observing the results in Tables 2 and 3, we can highlight the following:

- The worst performance of MC^{2}ESVM is obtained in the Balance and the Vehicle datasets, which have three and four

[4] The detailed results on each dataset and the hyper-parameters values of each method are provided as supplementary material. The supplementary material and source code are available at http://ccc.inaoep.mx/~arosales/resources/MC^{2}ESVM.tar.gz.

classes, respectively. On the other hand, its best performance is shown in the Cleveland, Ecoli, and Zoo datasets, which have at least five classes each.

- ❑ It is worth noting that, in general, MC^{2}ESVM significantly performs better than reference methods in datasets with imbalance rates greater than 1.5 and with five or more classes. This may be explained due to the fact that OVO with a large number of classes, significantly increases the number of binary classifiers, leading to an ensemble with a more complex decision function. OVA, on the other hand, artificially makes higher this imbalance.
- ❑ On well-balanced problems, the performance of MC^{2}ESVM and reference methods are quite similar, regardless of the number of classes.
- ❑ MC^{2}ESVM statistically outperforms most of the SVM formulations for handling multiclass problems. In fact, it is statistically better in five out of six methods for the accuracy score and in four out of six methods for Cohen's Kappa statistic, under the considered level of $\alpha = 0.05$.
- ❑ MSVM-CS and the OVO decomposition are clearly the most competitive multiclass SVMs for the proposed MC^{2}ESVM. These competitive performances can also be noted in the lack of a statistically significant difference when Cohen's Kappa is considered.
- ❑ The difference between MC^{2}ESVM and OVO in accuracy is marginal, but in Cohen's Kappa, MC^{2}ESVM clearly outperforms OVO. This is an interesting point to observe because the hyper-parameters for each method were done by considering the accuracy as the main criterion. Thus, MC^{2}ESVM is not overfitted to this criterion.
- ❑ Except for MSVM-CS, the rest of the multiclass SVMs based on modifying the objective function showed a low performance. This may be due to the fact that they have to deal with a larger optimization problem than those based on decomposition.
- ❑ OVO can be highlighted as the best method based on decomposing the problem into multiple binary classification problems, while MSVM-CS is an outstanding method from those based on modifying the optimization problem.

The Holm's test has reported no statistically significant difference between MC^{2}ESVM and OVO neither between MC^{2}ESVM and MSVM-CS, when the multiple comparison is done by considering all methods. This may be due to the number of algorithms in the comparison and the fact that those algorithms have influence on the rank computation and also in the post-hoc [39]. Therefore, we have thoroughly inspected these three algorithms by comparing them. With this aim, Table 4 shows the statistical comparison when considering these methods. MC^{2}ESVM is again considered as a control method in the test. Based on this more focused test, we can note that indeed the differences between MC^{2}ESVM and the reference methods (OVO and MSVM-CS) are statistically significant at the considered level.

TABLE 3 Average rankings of the methods computed with Friedman Aligned Ranks (FAR) and Holm's adjusted p-values (p_{Holm}).

	ACC		$\mathcal{K}$	
METHOD	FAR	p_{Holm}	FAR	p_{Holm}
MC^{2}ESVM	50.28	–	51.20	–
OVO	70.36	0.1611	75.30	0.1128
MSVM-CS	83.82	**0.0385**	78.54	0.1128
OVA	92.78	**0.0091**	94.56	**0.0074**
MSVM^2	103.44	**0.0008**	102.94	**0.0015**
MSVM-WW	105.04	**0.0007**	101.12	**0.0020**
MSVM-LLW	110.28	**0.0002**	112.34	**0.0001**

TABLE 4 Average Friedman Aligned Ranks (FAR) and Holm's adjusted p-values (p_{Holm}) for the best methods.

	ACC		$\mathcal{K}$	
METHOD	FAR	p_{Holm}	FAR	p_{Holm}
MC^{2}ESVM	27.16	–	27.96	–
OVO	39.92	**0.0385**	41.74	**0.0254**
MSVM-CS	46.92	**0.0027**	44.33	**0.0161**

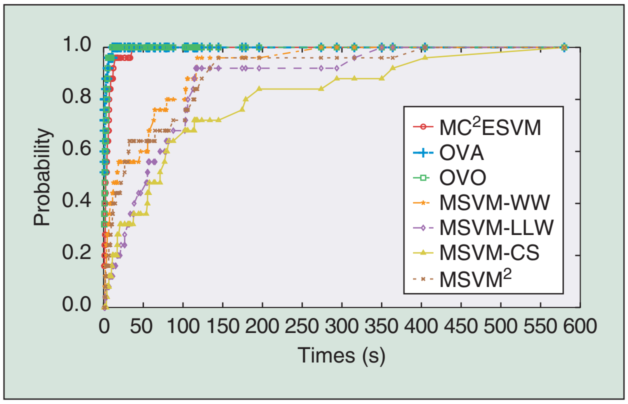


FIGURE 3 Probability of each method to learn a Multiclass SVM classifier in a given amount of time.

Fig. 3 graphically depicts a comparison of the computational time for each method. This figure represents the probability that a given method learns the multiclass SVM in a given amount of time. From it, we can note:

- ❑ Both OVO and OVA are the best ones, and have virtually the same performance.
- ❑ MC^{2}ESVM is the second best one, requiring at most, 30 seconds for solving each benchmark problem.
- ❑ Single machine methods are clearly the slowest ones. This is due to the fact that they deal with a larger optimization problem.
- ❑ In the best case, single machine methods required around 280 seconds to ensure solving each dataset.
- ❑ Among all SVMs multiclass extensions, MSVM-CS is the worst one in terms of computational time.

2) Comparing with other Learning Algorithms

In this section, our goal is to contrast the performance of MC^{2}ESVM with common learning algorithms. To this end, RF, MLP, KNN, and NB are chosen as reference methods.

Table 5 shows the reported results obtained by each method on each dataset and Table 6 shows the ranking for each algorithm computed with the Friedman's Aligned Ranks method and the adjusted *p*-values with the Holm's test. Observing these results, we stress the following:

- MC^{2}ESVM achieves, on average, the highest scores both on the accuracy and Kappa statistics metrics.
- Among the reference methods, RF is the most competitive, since it has no statistically significant difference with respect to MC^{2}ESVM.

TABLE 5 Comparison between MC^{2}ESVM and RF, MLP, KNN, and NB.

METHOD	*ACC*	$\mathcal{K}$
MC^{2}ESVM	**0.8167 ± 0.1787**	**0.7424 ± 0.2437**
RF	0.8048 ± 0.1742	0.7229 ± 0.2417
MLP	0.7932 ± 0.1837	0.7164 ± 0.2511
KNN	0.7607 ± 0.2020	0.6638 ± 0.2873
NB	0.7238 ± 0.1934	0.6319 ± 0.2577

TABLE 6 Average rankings of the methods computed with Friedman Aligned Ranks (FAR) and Holm's adjusted *p*-values (p_{Holm}).

	ACC		$\mathcal{K}$	
METHOD	FAR	p_{Holm}	FAR	p_{Holm}
MC^{2}ESVM	35.28	–	37.60	–
RF	53.76	0.0713	56.62	0.0634
MLP	61.04	**0.0239**	59.26	0.0634
KNN	77.22	**<0.010**	78.42	**<0.010**
NB	87.70	**<0.010**	83.10	**<0.010**

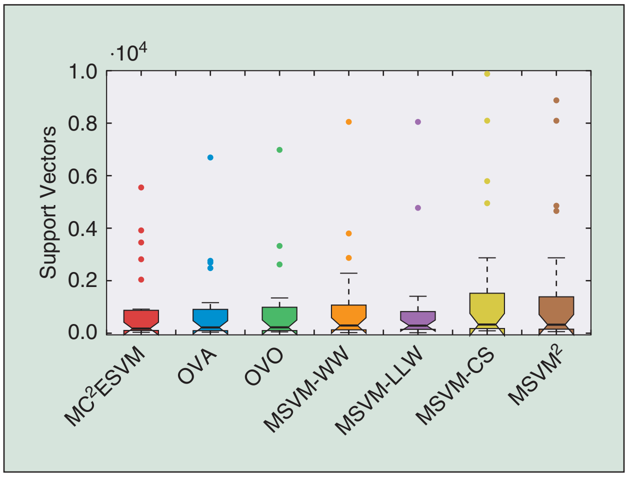


FIGURE 4 Boxplot for the number of support vectors generated by each method.

- NB is generally the worst one. The exception is on the Splice dataset, where it achieves the best performance. This may be explained due to the inductive bias of this algorithm, which assumes independence between attributes.

B. Analyzing the Support Vectors

The aim of this study is twofold. First, by comparing the number of support vectors generated by each method, this can give insights about the complexity of the learned model. Second, we show the effect in the decision function for each method.

For the first case, Fig. 4 graphically depicts the distribution of the number of support vectors resulting from each method. Based on it, we can stress the following for this first part:

- MC^{2}ESVM, OVO, OVA, MSVM-WW, and MSVM-LLW are quite similar in terms of the number of learned support vectors.
- MSVM-CS and MSVM^2 are the worst performers under this criterion. This can be explained due to the fact that these methods deal with a large number of variables during the optimization process. Furthermore, MSVM-CS does not consider the bias in its formulation, which can also explain this behavior.

The second goal of this study is to contrast the decision boundaries learned by each method. For illustrative purposes, we have generated an artificial dataset with two features and three classes. This artificial dataset is best known as the Madelon dataset and it is generated following the methodology proposed in [40]. A noise level of 10% is induced in this dataset.

Fig. 5 graphically depicts the decision regions that are learned by each method. Fig. 5(a) shows the training points used to fit the model's parameters. Figs. 5(b) to 5(h) show the region for each class generated by each method. From this, the following can be noted:

- MC^{2}ESVM seems to better capture the regions of each class, by exhibiting well-defined regions for each. Moreover, MC^{2}ESVM is able to learn a simpler function, which does not show a wiggle shape.
- OVO and OVA show overlapped regions. This may be explained due to the fact that both OVO and OVA work by solving several binary classification problems, independently, which leads to regions of uncertainty.
- In general, MSVM-CS and MSVM^2 generate more complex decision boundaries than the other methods. This is also consistent with the fact that they are the ones with the highest number of support vectors, which can be considered a measure of the model's complexity.
- MSVM-WW has the highest overlapped area. This is also consistent with its low performance in the classification task.

C. Analyzing the Scalability

The aim of this study is to assess the scalability of MC^{2}ESVM with respect to the number of classes and the number of training instances. In order to perform these studies, we have generated a set of artificial datasets, following the methodology proposed in [40], as in the previous study, but with the following considerations:

- ❏ For assessing the scalability with respect to the number of classes, the datasets are generated by varying the number of classes from 3 to 15. In all cases, a set of 25 features and 1,500 samples are fixed.
- ❏ For assessing the scalability with respect to the number of instances, the number of classes is fixed in 3 and the number of features is kept in 25. The number of instances for each dataset ranges from 800 to 6,000, with a step size of 400.

Fig. 6 depicts the behavior of MC^{2}ESVM under these conditions. Based on these figures, we can remark the following:

- ❏ The number of evaluations increases as either the number of instances or the number of classes is increased. This is an expected result, since MC^{2}ESVM deals with more complex problems.
- ❏ The number of classes seems to be more harmful in the scalability of MC^{2}ESVM. This may be due to the fact that, as the number of classes grows, the complexity of the decision boundaries is increased, making harder the recognition.

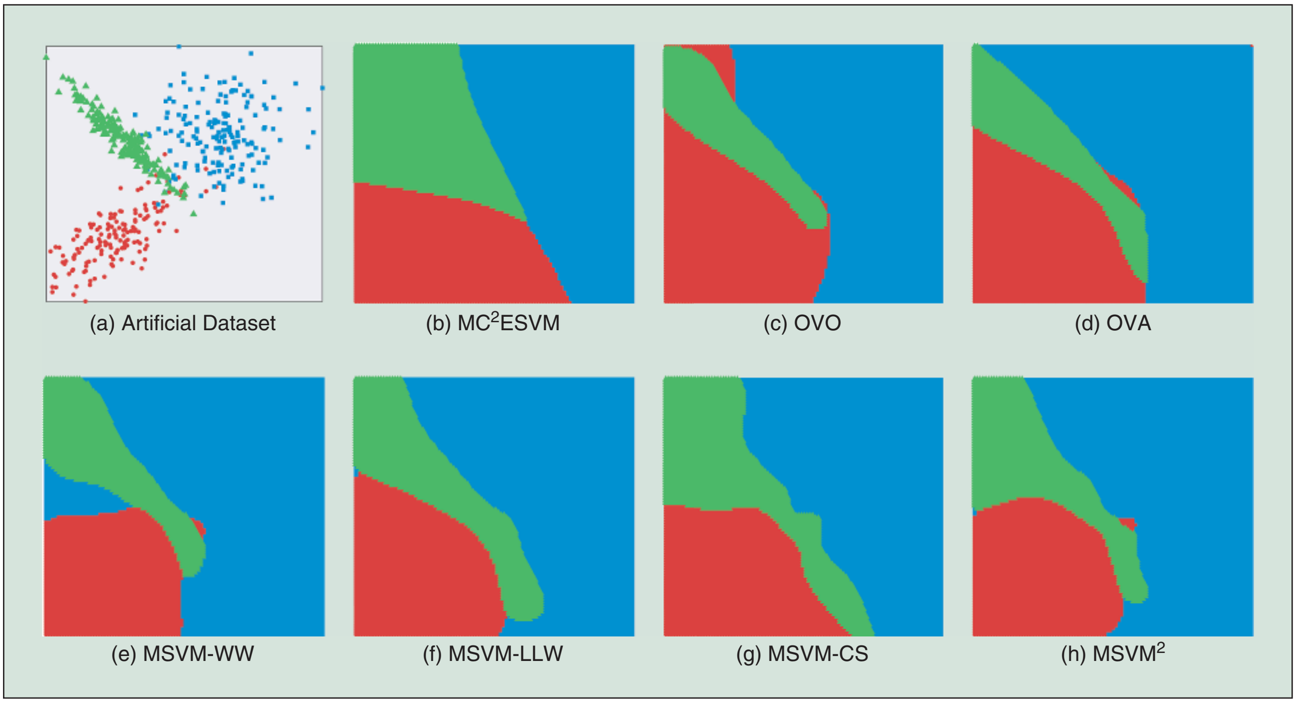


FIGURE 5 Decision boundary for each of the SVM multiclass extensions.

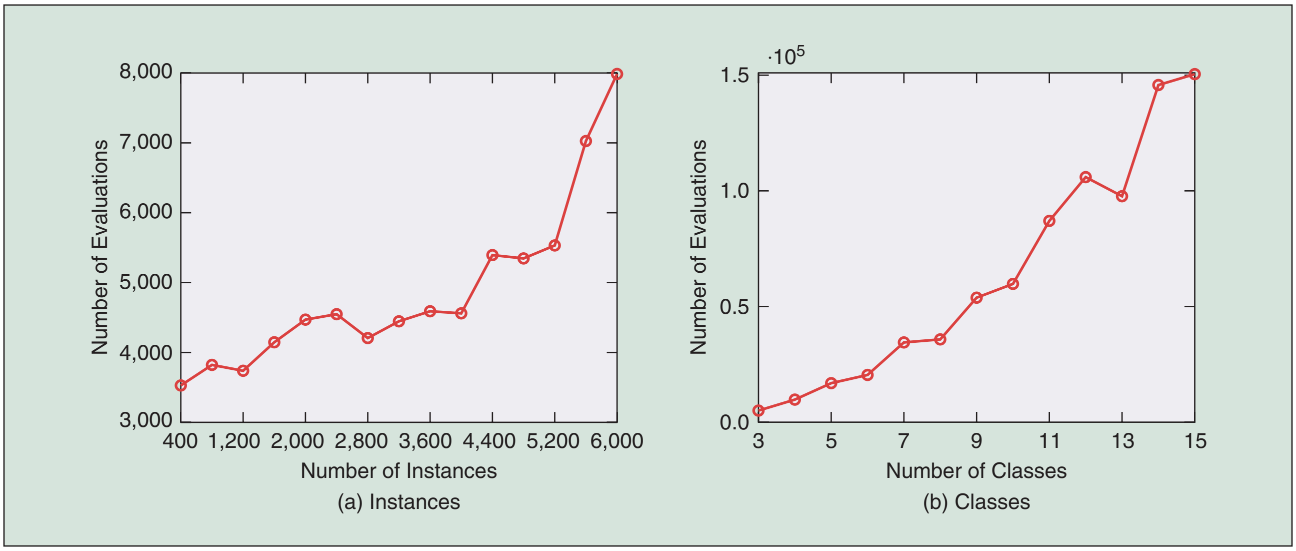


FIGURE 6 Scalability when either the number of classes or instances increases.

Another factor that also contributes to this is the cooperative approach, which decomposes the problem, having a population for each class. Thus, the number of evaluations of the objective function increases, at least, by a factor of the number classes. However, since each population works with the instances of the corresponding class, the computation requires a lower computational cost.

D. Analyzing the Stability

Due to the stochastic nature of the cooperative evolutionary algorithm in MC^{2}ESVM, another interesting issue is concerned with its stability. Thus, in order to assess it, we have carried out two analyzes. On the one hand, the classification performance over different replications is determined. On the other hand, we examine the parameters of the evolutionary algorithm.

For the first part, we have performed 15 replications of MC^{2}ESVM over each dataset. Fig. 7 shows the mean and standard deviation. From it, we can note how MC^{2}ESVM is able to get virtually the same performance in all replications. Thus, MC^{2}ESVM exhibits a stable behavior, showing a low variance in the classification performance.

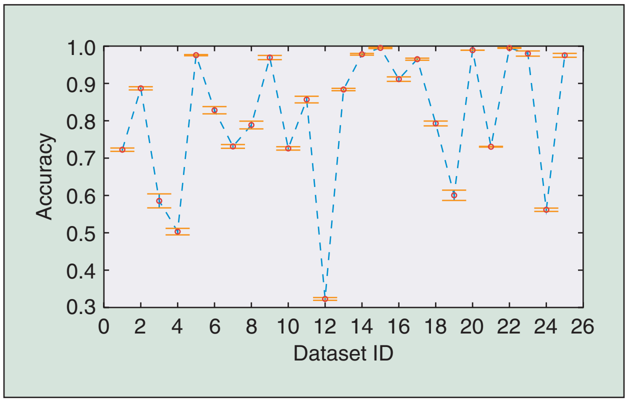


FIGURE 7 Mean and standard deviation of the 15 replications of MC^{2}ESVM for each dataset.

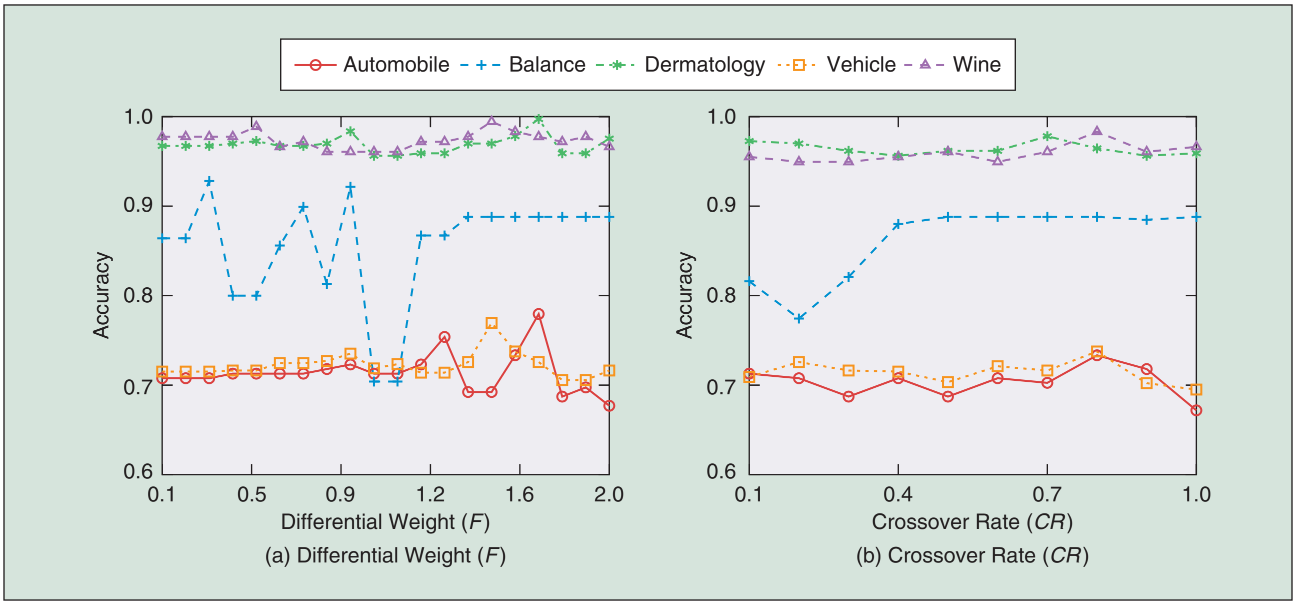
The remaining issue to analyze is the evolutionary parameters. Since MC^{2}ESVM uses differential evolution operators, we have tested it by varying the value of the differential weight (F) and the crossover rate (CR). Fig. 8 shows the performance on five representative datasets when these parameters are varied. Fig. 8(a) shows the effect of varying F in the range [0.1, 2.0] and Fig. 8(b) for CR in the range [0.1, 1.0]. Based on these results, a value in the range [1.5, 1.7] for the F parameter and in the range [0.6, 0.9] for the CR parameter, are good recommendations.

VI. Conclusions

This paper introduced a multiclass classification approach based on the cooperative evolution of support vector machines, called MC^{2}ESVM. The method is general in the sense that it subsumes the decomposition-based and the single-machine methods. MC^{2}ESVM is intuitive and takes advantages of the cooperative evolution to decompose the classification problem, such that each subproblem faces the learning of the support vectors to the specific class, but acting in a cooperative fashion by considering the information of the other classes.

Unlike decomposition-based extensions, MC^{2}ESVM is able to capture in a single model the multiclass classification problem, leading to a simpler decision function than the one obtained by fusing multiple binary classifiers. In contrast with single machine methods, MC^{2}ESVM does not increase the number of variables to be optimized, deriving in a simpler optimization problem with no additional constraints. These features make the proposed extension a more flexible approach.

The experimental evaluation over a set of 25 common benchmark datasets shows that MC^{2}ESVM is able to outperform most of the multiclass SVM extensions. This claim is supported by a set of non-parametric tests with a level of significance of $\alpha = 0.05$.



(a) Differential Weight (F)

(b) Crossover Rate (CR)

FIGURE 8 Performance of MC^{2}ESVM under different parameters values.

The experimental results have also revealed that MC^{2}ESVM performs better on problems with an imbalance rate higher than 1.5 and with a number of classes greater than five. MSVM-CS excels as the most competitive among the single machine methods. OVO stands as the best method from those based on binary decomposition. This finding is also confirmed in [5]. A focused analysis of these two prominent methods and MC^{2}ESVM revealed that the latter is able to statistically perform better. MC^{2}ESVM does not seem to be overfitted to the hyper-parameters optimized by the accuracy criterion. In fact, it shows a greater improvement on Cohen's Kappa when it is compared to the other methods. Furthermore, the computational time required by MC^{2}ESVM is not as high as that of existing single machine methods. In fact, it requires a reasonable time for learning a model, similar to that required by decomposition-based methods. Thus, it has benefitted from the advantages of each approach.

Another interesting conclusion is that MC^{2}ESVM showed to be able to learn simpler functions than most of the methods based on adapting the optimization problem. OVO and OVA, on the other hand, suffer from generating regions whit high uncertainty. Finally, the most important criterion for scalability of MC^{2}ESVM is the number of classes, which increases the number of evaluations in the optimization.

An advantage of EAs is that they will allow SVM to handle non-positive semidefinite kernels. This type of kernel leads to a non-convex optimization problem, narrowing their applicability to the quadratic solver of SVMs. As part of our future work, we will analyze the performance of MC^{2}ESVM on non-positive semidefinite kernels. Another interesting path for future research is to deepen into the interaction scheme of solutions of different populations. Finally, exploiting the parallelizable nature of EAs to handle the so-called Big Data problems is another interesting topic for future research.

Acknowledgments

This research has been supported by FEDER and by the Spanish National Research Project TIN2017-89517-P, by the Regional Andalusian project P11-TIC-7765 and by CONACyT under projects 221551 and 241461.

References

[1] V. N. Vapnik, *The Nature Statistical Learning Theory.* New York: Springer-Verlag, 1995.

[2] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, Oct. 2014.

[3] A. Demontis, M. Melis, B. Biggio, G. Fumera, and F. Roli, "Super-sparse learning in similarity spaces," *IEEE Comput. Intell. Mag.*, vol. 11, no. 4, pp. 36–45, Nov. 2016.

[4] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, May 2009.

[5] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognit.,* vol. 44, no. 8, pp. 1761–1776, Aug. 2011.

[6] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res*, vol. 5, pp. 101–141, Jan. 2004.

[7] F. Lauer and Y. Guermeur, "MSVMpack: A multi-class support vector machine package," *J. Mach. Learn. Res.*, vol. 12, pp. 2293–2296, July 2011.

[8] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, no. 3, pp. 397–415, May 2008.

[9] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 4–19, Feb. 2014.

[10] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. Coello, "Survey of multiobjective evolutionary algorithms for data mining: Part II," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 20–35, Feb. 2014.

[11] C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, and T. Paquet, "A multi-model selection framework for unknown and/or evolutive misclassification cost problems," *Pattern Recognit.*, vol. 43, no. 3, pp. 815–823, Mar. 2010.

[12] A. Rosales-Pérez, S. García, J. A. Gonzalez, C. A. Coello Coello, and F. Herrera, "An evolutionary multiobjective model and instance selection for support vector machines with pareto-based ensembles," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 863–877, Dec. 2017.

[13] N. Verbiest, J. Derrac, C. Cornelis, S. García, and F. Herrera, "Evolutionary wrapper approaches for training set selection as preprocessing mechanism for support vector machines: Experimental evaluation and support vector analysis," *Appl. Soft Comput.*, vol. 38, pp. 10–22, Jan. 2016.

[14] M. L. D. Dias and A. R. R. Neto, "Evolutionary support vector machines: A dual approach," in *Proc. IEEE Congr. Evolutionary Computation*, 2016, pp. 2185–2192.

[15] I. Mierswa, "Evolutionary learning with kernels: A generic solution for large margin problems," in *Proc. 8th Annu. Conf. Genetic Evolutionary Computation*, 2006, pp. 1553–1560.

[16] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of classifier methods: A case study in handwritten digit recognition," in *Proc. 12th Int. Association Pattern Recognition Conf.*, 1994, vol. 2, pp. 77–82.

[17] U. H. G. Kreßel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 255–268.

[18] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems*, Vol. 12, S. A. Solla, T. K. Leen, and K. Müller, Eds. Cambridge, MA: MIT Press, 2000, pp. 547–553.

[19] J. Weston and C. Watkins, "Multi-class support vector machines," Royal Holloway, Univ. London, London, Tech. Rep. CSD-TR-98-04, 1998.

[20] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data," *J. Amer. Stat. Assoc.*, vol. 99, pp. 67–81, June, 2004.

[21] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, Mar. 2002.

[22] Y. Guermeur and E. Monfrini, "A quadratic loss multi-class SVM for which a radius-margin bound applies," *Informatica*, vol. 22, no. 1, pp. 73–96, Jan. 2011.

[23] G. J. J. Van Den Burg and P. J. F. Groenen, "GenSVM: A generalized multiclass support vector machine," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 7964–8005, Jan. 2016.

[24] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[25] R. P. Wiegand, "An analysis of cooperative coevolutionary algorithms," Ph.D. dissertation, George Mason Univ., Fairfax, VA, 2004.

[26] C. D. Rosin and R. K. Belew, "New methods for competitive coevolution," *Evol. Comput.*, vol. 5, no. 1, pp. 1–29, Mar. 1997.

[27] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, Mar. 2000.

[28] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series).* Secaucus, NJ: Springer-Verlag, 2005.

[29] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2001.

[30] J. Alcalá, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Oct. 2008.

[31] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, no. 2–3, pp. 255–287, June, 2011.

[32] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.

[33] A. J. Bagnall and G. C. Cawley, "On the use of default parameter settings in the empirical evaluation of classification algorithms," *CoRR*, vol. abs/1703.06777, Mar. 2017.

[34] J. Wainer and G. Cawley, "Empirical evaluation of resampling procedures for optimising SVM hyperparameters," *J. Mach. Learn. Res.*, vol. 18, no. 15, pp. 1–35, Jan. 2017.

[35] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[36] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[37] S. García and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, Dec. 2008.

[38] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inform. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.

[39] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?" *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 152–161, Jan. 2016.

[40] I. Guyon, S. Gunn, A. B. Hur, and G. Dror, *Design and Analysis of the NIPS2003 Challenge*. Berlin: Springer, 2006, pp. 237–263.

Enhancing Selection Hyper-Heuristics via Feature Transformations

IMAGE LICENSED BY INGRAM PUBLISHING

Ivan Amaya, José C. Ortiz-Bayliss, Alejandro Rosales-Pérez, Andrés E. Gutiérrez-Rodríguez, Santiago E. Conant-Pablos, and Hugo Terashima-Marín
School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey, MEXICO

Carlos A. Coello Coello
Evolutionary Computation Group, CINVESTAV-IPN, Mexico City, MEXICO

Abstract—Hyper-heuristics are a novel tool. They deal with complex optimization problems where standalone solvers exhibit varied performance. Among such a tool reside selection hyper-heuristics. By combining the strengths of each solver, this kind of hyper-heuristic offers a more robust tool. However, their effectiveness is highly dependent on the 'features' used to link them with the problem that is being solved. Aiming at enhancing selection hyper-heuristics, in this paper we propose two types of transformation: explicit and implicit. The first one directly changes the distribution of critical points within the feature domain while using a Euclidean distance to measure proximity. The second one operates indirectly by preserving the distribution of critical points but changing the distance metric through a kernel function. We focus on analyzing the effect of each kind of transformation, and of their combinations. We test our ideas in the domain of constraint satisfaction problems because of their popularity and many practical applications. In this work, we compare the performance of our proposals against those of previously published data. Furthermore, we expand on previous research by increasing the number of analyzed features. We found that, by incorporating transformations into the model of selection hyper-heuristics, overall performance can be improved, yielding more stable results. However, combining implicit and explicit transformations was not as fruitful. Additionally, we ran some confirmatory tests on the domain of knapsack problems. Again, we observed improved stability, leading to the generation of hyper-heuristics whose profit had a standard deviation between 20% and 30% smaller.

Digital Object Identifier 10.1109/MCI.2018.2807018
Date of publication: 10 April 2018

Corresponding Author: Ivan Amaya (Email: iamaya2@itesm.mx)

1556-603X/18©2018IEEE

I. Introduction

Hyper-heuristics have emerged in recent years as a strategy for combining the strengths of different heuristics into a single method. Their aim is to provide more flexibility when solving a wider variety of optimization problems [1]. Initially, authors used the term hyper-heuristic to describe heuristics for choosing heuristics. Nowadays, this definition also includes the automatic generation of heuristics. Thus, hyper-heuristics operate at a higher level of generality by working with low-level heuristics rather than by solving a problem. Burke et al. present an in-depth survey about the topic in [1].

Selection hyper-heuristics represent a subset that relies on a mechanism for interpreting the problem state and deciding the most suitable heuristic to apply. One of the main challenges for automating this decision is how to properly characterize such a state, to allow a correct mapping to heuristics. There are different ways to get such a mapping. Examples include machine learning [2] and evolutionary algorithms [3]. Despite this, the effectiveness of the selection depends greatly on the predictive power of the feature set [4].

Feature preprocessing is a major step in data mining and it plays a central role in the generalization performance of the models. This step encompasses selection, generation and transformation [5]. Feature selection seeks to choose a subset of the most relevant features, for the problem at hand. Feature generation focuses on getting a new set of more discriminating features by combining primitive ones. Feature transformation aims at adapting the set of features to help learning algorithms improve their learning stage. Xue et al. offer a comprehensive review on feature selection and generation in [6], [7]. Similarly, other authors give an in-depth discussion on feature transformation in [5], [8].

In recent years, there has been an interest in studying feature preprocessing for hyper-heuristics. Outstanding applications include [9]–[11]. In [9], Smith-Miles and Lopes studied the relationships between critical features of problem instances and algorithm performance. Besides, Montazeri explored feature selection through genetic algorithms in [10]. Additionally, Hart et al. analyzed the effect of feature generation in hyper-heuristics [11]. Most early studies in hyper-heuristics have focused on feature selection or feature generation while neglecting an analysis of feature transformation to empower hyper-heuristics. The only work in this regard is [12], where several transforming functions were proposed. Such functions required a manual tuning of parameters. Moreover, the authors failed to assess their scalability beyond two dimensions, and its generalization for several problem domains. To the best of our knowledge, no previous study has adapted the transformation functions and explored kernel transformations [13]. Our hypothesis is that doing so may facilitate the rule-creation process (see Sect. V-A). This, in turn, may allow selection hyper-heuristics to work on higher dimensional spaces and for a broader set of domains, enhancing them.

Thus, this paper makes two main contributions. First, it proposes explicit transformations, based on linear and S-shaped functions, with parameters tailored to the distribution of each feature. Second, it exchanges the distance function for one using a radial basis function kernel. Our experimental results confirm the validity of the proposed methods for improving the performance of selection hyper-heuristics in two optimization domains.

The rest of the paper is organized as follows. Section II presents the preliminaries for properly understanding this work. We delve in the inner workings of a hyper-heuristic, and on the problems related to using features directly. Moreover, we present work previously done to explore the feasibility of using feature transformations. In this section we also present the main ideas regarding kernels and our test domain. Afterwards, we describe our proposed approach (Section III), focusing on the definition of the transformations. Section IV describes the experimental methodology adopted in this work, which was split into four stages. Section V presents the obtained data and their discussion. Striving to facilitate the meaning of our data, we split this section into the same four stages as Section IV. Finally, the conclusions and some possible paths for future work are laid out in Section VI.

II. Preliminaries

We begin this section by describing what hyper-heuristics are and how they operate. We focus on selection hyper-heuristics since it is the approach used in this work, although it is worth highlighting that other types of hyper-heuristics exist. After that, we explain two problems related to features and how transforming them may prove helpful. We then summarize the main idea behind a previously reported work dealing with feature transformations. We wrap this section up by discussing the main ideas behind a notion known as *Kernel* and the domain in which we carry out our experiments.

A. Hyper-Heuristics

The No-Free-Lunch (NFL) theorem [14] implies that there is no algorithm that best solves all kinds of problems. Thus, a recent and recurring alternative is to use a strategy for combining many feasible solvers. Each of the problems that requires being solved is usually referred to as an "instance". Even if this is a clever way to try and circumvent the restrictions posed by the NFL theorem, a recursive problem arises: the algorithm selection problem [15]. Here, focus migrates to finding a proper way to carry out the selection in such a way that turns out to be beneficial. Several approaches rely on this idea to improve the generality of their solution method [16], [17], but detailing them is beyond the scope of this manuscript.

Throughout this work we use an evolutionary hyper-heuristic model proposed in [18]. This model falls into the category of selection hyper-heuristics, and it is depicted in Fig. 1. The idea is to solve a given instance using a combination of algorithms, ruled by a "selector". The pillar for this idea is that a partially solved problem may not be as efficiently solved by the same algorithm and, thus, a change should be made. To identify the moment in which it is appropriate to switch the solver, this

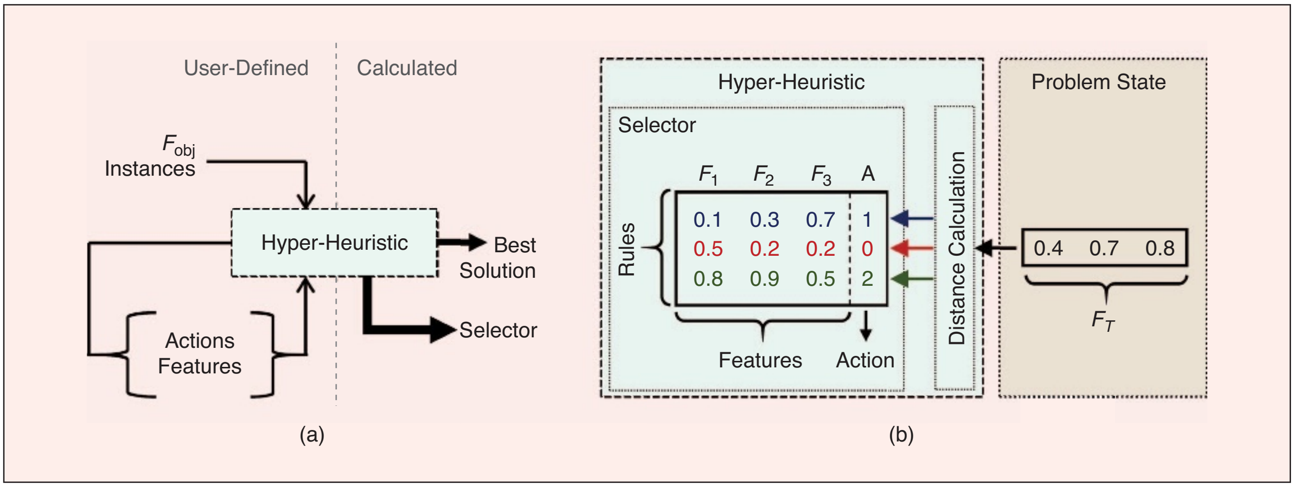


FIGURE 1 Overview of the hyper-heuristic model (a) and its inner workings (b). In this framework, the problem state is a vector that characterizes the instance being solved in its current stage. The hyper-heuristic contains a set of rules where the condition is represented also by a vector and their action corresponds, in this case, to specific heuristics.

model represents the problem by a set of features. These can be as simple as the size of the problem, or as complex as a relation between different values of each problem.

It is worth mentioning that the hyper-heuristic model used in this work closely resembles Learning Classifier Systems (LCS). Similarities focus on the way rules are generated and how they are applied based on the problem features. In fact, there are some works for selection hyper-heuristics that deepen into the way LCS can generate or be used as hyper-heuristics [19], [20].

As can be seen in Fig. 1 (left), the user provides four elements. The first one is the set of instances that will be solved. The second one is an objective function for measuring the quality of the solution. The remaining ones are the set of features and solvers. Using these information, the model randomly selects a subset of instances for training itself. For a given selector, the hyper-heuristic selects the first instance and calculates its features, F_T. With this starting point, it calculates the Euclidean distance to each rule, and applies the action (i.e., the heuristic) of the closest one. This process is depicted in Fig. 1 (right). After an action, the problem state changes. The process is repeated until the instance is solved. At this point, the hyper-heuristic moves to the next instance and the process continues until finishing all of them.

As mentioned before, the model used in this work has the ability to train itself. Here, this means an iterative procedure, where a messy genetic algorithm evolves a population of selectors. To do so, the process described above is carried out for each selector, and the objective function defined by the user guides the evolution. In this work, we used a steady-state configuration with a population size of 20, a crossover rate of 1.0 and a mutation rate of 0.1. Furthermore, we allowed the algorithm to run for 100 cycles (i.e., generations).

1) An Illustrative Example

Aiming to better clarify how selection hyper-heuristics work, we now present a simple, but useful, example. Imagine a problem where you have a set of items and need to split them into two subsets with the lowest possible difference. Therefore, the quality of a solution can be measured through Eq. 1, where $item_x^y$ represents element x from subset y, and N_y represents the number of elements in subset y.

$$Q = \left| \sum_{i=1}^{N_1} item_i^1 - \sum_{j=1}^{N_2} item_j^2 \right|. \tag{1}$$

One way of solving this problem requires two steps. First, assign all items to a single subset (i.e., subset 1). Then, choose some items for moving to the other subset (i.e., subset 2). Here, item transfer should stop if subset 2 represents, at least, half the items. Item selection can be done through two low-level heuristics: the Max load, and the Min load. Both can perform well or poorly, depending on the problem instance. For example, if the set of items is [10 1 1 1 1 1 1 1 1 1 1 1], both heuristics yield a perfect split ($Q = 0$). However, if the set is [10 9 8 1 1 2 2 1 1 1 1 1 1], the quality is 15 for the Max heuristic and 1 for the Min heuristic. Moreover, if the set comprises [10 3 4 2 10 10 1 1 1 1 1 1], the Max heuristic yields a solution with $Q = 14$ while the Min yields one with $Q = 6$.

A hyper-heuristic may represent a better approach, but it requires a way of mapping features to actions (i.e., the rules). Here, a single feature (F_1) can be defined as shown in Eq. 2, and one rule can be defined for each action.

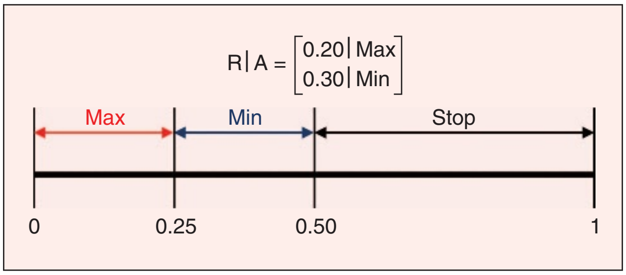
$$F_1 = \frac{\sum_{j=1}^{N_2} item_j^2}{\sum_{i=1}^{N_1} item_i^1 + \sum_{j=1}^{N_2} item_j^2}. \tag{2}$$

Figure 2 shows a sample set of rules (top) and its corresponding zone of influence (bottom). This means that, whenever the feature value is below 0.25, an item will be moved following the Max heuristic. For higher feature values, it will be moved according to the Min heuristic. For example, consider the last instance. The first two items will be selected using the Max heuristic, since feature values are 0 and 0.22. When selecting the third item, the feature value goes up to 0.43. Thus, from this point onward all items are selected using the Min heuristic, until the feature becomes, at least, equal to 0.5. Through this approach, problem instances one and three can be perfectly solved (i.e., $Q = 0$), while the second one can be solved with $Q = 1$, thus making it a better solver.

B. Problems Derived from Using Features Directly

Using the aforementioned model with raw features may exhibit two drawbacks [12]: likeliness and stagnation. The former appears when two problem instances with similar features are best solved by different actions (e.g., at the boundary of regions best solved by both actions). Should one rule (from the selector) be closest to both states, one of them would not be solved in the best possible way. Similarly, if problem states best solved by the same action are apart, clustering them frees up space to distribute them among other actions. Figure 3 shows an example of both scenarios. In the figure, each circle represents the best location for a rule, and their corresponding actions are given by A_i. The square marker represents the current state of a problem, indicated by F_T, and where the action to be taken, A_T, must be decided by using the closest point. However, on the left, rules one and three are so clustered that a small error when placing them (e.g., when evolving), could lead to a wrong decision. At the right, transformations help by clustering alike regions and expanding troublesome ones, allowing for a smoother change in the performance of a selector-in-training.

The second problem, to which we refer to as stagnation, is related to the nature of optimization procedures [12]. During the first iterations, improvements are significant and quite common. As the search progresses, they become less frequent and less significant. For our hyper-heuristic model, stagnation reflects on a population of selectors with small differences in both, features and performance. Nonetheless, by transforming features we can expand part of the feature space, allowing for a bigger variation which may lead to improvements.



$R|A = \begin{bmatrix} 0.20 | \text{Max} \\ 0.30 | \text{Min} \end{bmatrix}$
Max
Min
Stop
0
0.25
0.50
1

FIGURE 2 An example of a set of rules (top) and its corresponding zone of influence (bottom).

C. Previous Work Related to Feature Transformations

Feature transformation is an active research field in machine learning, where the idea is to use information from the original features to create new ones with improved predictive power. Several methods have been proposed. Some of the best known methods are normalization, standardization, and polynomial transformation [5]. An emerging area is the use of evolutionary computation for taking into account the behavior of the recognition system [6]. In spite of its success, this approach is computationally costly.

We have previously studied feature transformation for improving the performance of selection hyper-heuristics. In [12], we applied Eq. 3, where $K = 5$ is a parameter we determined empirically, and which behaves as shown in Fig. 4. To improve upon that idea, in this work we test two new transformations and a way for tailoring them to each feature (Sect. III).

$$\Phi(\mathbf{x}) = 1 - 2 \cdot \left(\frac{e^{-K \cdot \mathbf{x}} - e^{-K}}{1 + e^{-K \cdot \mathbf{x}}} \right). \qquad (3)$$

D. Kernels

Kernel functions implicitly embed mapping functions, and were popularized by Support Vector Machines [21] and the so-called *kernel trick*. This has enabled them to learn nonlinear functions, greatly improving their performance.

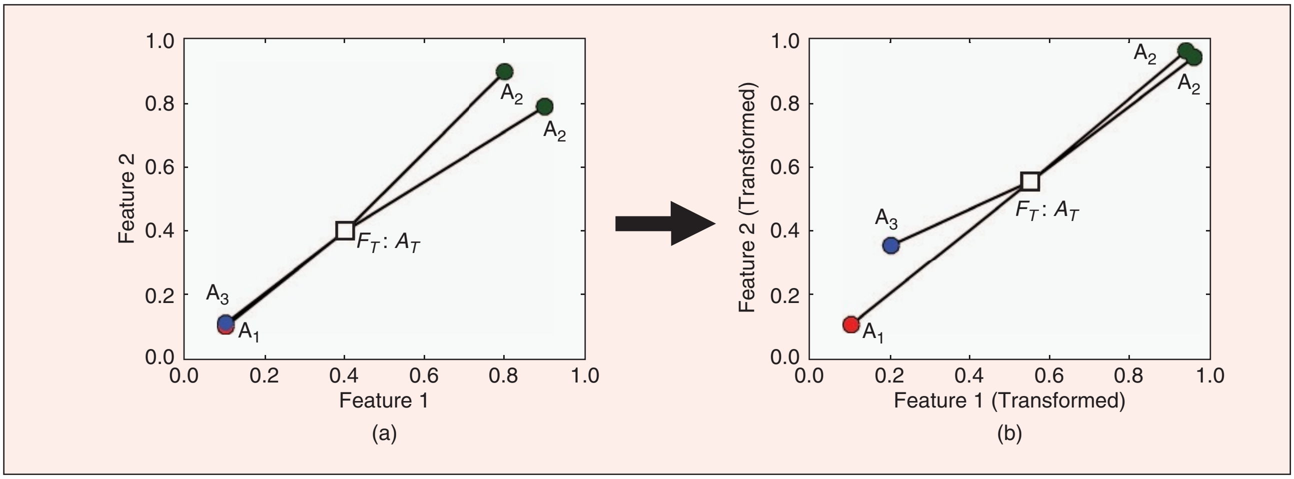


FIGURE 3 Distribution of rules before (a) and after (b) feature transformation. Square: Current state of the problem. Circles: Ideal location of rules. A_i: Actions of each ideal rule.

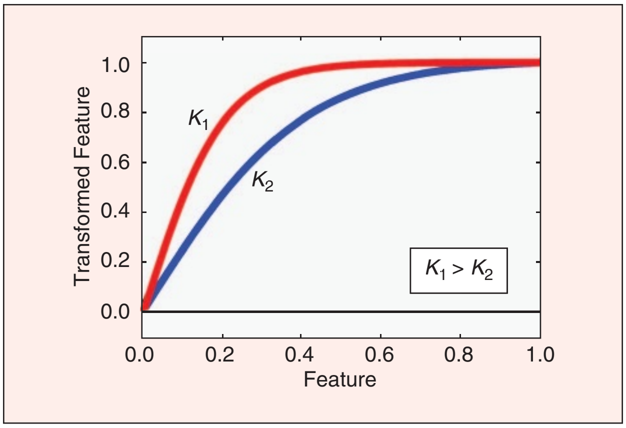


FIGURE 4 Overview of the exponential transformation previously used in [12].

Roughly speaking, a kernel is a mathematical function that computes the inner product between two vectors in a higher (and possibly infinite) dimensional space without explicitly performing the mapping. A kernel function, K, for two points, $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, is expressed in Eq. 4, where $\langle \cdot, \cdot \rangle$ is the inner product between two vectors.

$$K(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \langle \Phi(\mathbf{x}^{(1)}), \Phi(\mathbf{x}^{(2)}) \rangle. \tag{4}$$

Some common kernels are shown in Eq. 5, Eq. 6, and Eq. 7, respectively, where $\| \cdot \|$ is the Euclidean norm and d and γ are adjustable parameters.

$$\text{Linear Kernel: } K_L(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \langle \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \rangle \tag{5}$$

$$\text{Polynomial Kernel: } K_P(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = (\langle \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \rangle + 1)^d \tag{6}$$

$$\text{Radial basis function Kernel: } K_{RBF}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = e^{-\gamma \| \mathbf{x}^{(1)} - \mathbf{x}^{(2)} \|^2}. \tag{7}$$

E. Domain Under Study: Constraint Satisfaction Problems (CSPs)

There are many practical applications for a CSP [22], [23]. These problems can be defined by a set of variables, V, where each variable $v \in V$ contains two pieces of information: its domain (D_v) and its constraints (C). The former represents the finite set of available values for the variable, while the latter restricts combinations of the variable with others. Solving a CSP, thus, requires assigning a feasible value to every variable (a process known as "instantiation") in such a way that all constraints are satisfied [12].

A recurring approach for solving CSPs is to use a tree representation explored in a depth-first manner. Details are omitted due to space restrictions, but interested readers are referred to [12], [18] for more details on this. We simply mention that every node represents the assignment of one variable. Furthermore, constraints must be checked after arriving at each node to verify the feasibility of a solution.

Eight features were considered for this work as an attempt to incorporate information from the distribution of the constraints and conflicts in the instance. The set of features is consistent with the ideas considered in [18]:

- **Constraint density (p_1):** The constraint density is a measure of the proportion of constraints within the instance; the closer the value of p_1 to 1, the more constraints exist within the instance.
- **Constraint tightness (p_2):** The constraint tightness estimates how difficult the constraints are to be satisfied. Higher values indicate instances more likely to be unsatisfiable. The tightness of a constraint represents the proportion of conflicting tuples within such a constraint. Then, the constraint density of an instance is calculated as the average constraint tightness among all the constraints in the instance.
- **Clustering coefficient (c):** This feature considers the instance as a graph where variables are represented as nodes and constraints as edges. The local clustering coefficient of a variable measures how close their neighbors are to being fully connected. The clustering coefficient of an instance is the average of the local clustering coefficients among all the variables.
- **Upper and lower constraint density quartiles (UQp_1 and LQp_1):** These two features provide information of the distribution (based on the upper and lower quartiles) of the constraint density of the individual variables within the instance.
- **Upper and lower constraint tightness quartiles (UQp_2 and LQp_2):** The same idea than in the previous feature but focused on the constraint tightness of each particular constraint.
- **Kappa (κ):** This concept is suggested in the literature as a general estimation of how restricted a combinatorial problem is [24]. If κ is small, the problems usually have many solutions to their size. When κ is large, instead, the problems often have few solutions or have none at all.

Along with the features previously described, four commonly used heuristics are given as tools for hyper-heuristics:

- **DOM:** DOM instantiates first the variable that is more likely to fail. DOM estimates how likely a variable is to fail by counting the number of values in its domain, and chooses the variable with the fewest available values.
- **DEG:** This heuristic selects the variable involved in the maximum number of constraints with unassigned variables [25].
- **KAPPA:** It selects the next variable such that the new subproblem minimizes the κ factor for the whole instance [24].
- **WDEG:** WDEG attaches a weight to every constraint of the problem [26]. The weights are initialized to one and increased by one whenever its respective constraint fails during the search. Then, the weighted degree of a variable is calculated as the sum of the weights of the constraints in which the variable is currently involved. WDEG gives priority to the variable with the largest weighted degree.

In all cases, the performance of the methods for solving CSPs was measured by using at least one of the following metrics [12]:

❑ **Consistency Checks (CC):** Total revisions of constraints after an instance ends, such that the larger the number of constraints, the more expensive the search becomes. This value is used in the objective function during the training phase of the hyper-heuristics.

❑ **Adjusted Consistency Checks (ACC):** Similar to the previous one, but discarding instances where the solver times out.

❑ **Success Rate (SR):** Relation between the number of completed and tested instances. The higher the rate, the better the solver.

III. Our Proposed Approach

We explore two fronts for improving the predictive power of features in selection hyper-heuristics: explicit and implicit transformations. Our motivation for doing so is twofold. The first one is that original features may change a lot throughout the first iterations, but eventually arrive at a point of negligible change. By using feature transformations this behavior could be delayed. The second one is that part of the feature space may be wasted by considering feature values that never (or scarcely) appear in practice and that belong to the same solver. Through feature transformation these regions could be compressed, raising the importance of regions that belong to different solvers. We now provide the main elements of each transformation, and throughout this work we also explore the eventual benefit of combining them (see Sect. IV).

A. Proposed Explicit Feature Transformations

This section presents two explicit transformations for a single feature (i). Expressions are given in terms of a midpoint (M_i),and a half-width (W_i). For this work, we considered that every point in the training set may be meaningful within the test set and should be preserved. Thus, we defined $M_i = (\max(f_i) + \min(f_i))/2$ and $W_i = (\max(f_i) - \min(f_i))/2$. Here, f_i is a vector containing the values of feature i for the training instances. Figure 5 shows the location of all instances used in this work (except for the confirmatory testing; please refer to Sect. IV for more details). This plot corresponds to information yielded by a Principal Component Analysis (PCA) that was used to reduce the data from eight to two features. Moreover, data have been separated into training (stars) and testing (diamonds). Besides, the train/test ratio that will be used in the tests was also considered here. As Fig. 5 shows, unsolved and solved instances are spread out throughout the feature domain. Furthermore, the unsolved training instances (black stars) that seem away from the other ones actually share their location with unsolved testing instances (magenta diamonds). Therefore, if the hyper-heuristic uses this information during its training, it may perform better. Besides, as instances are solved, their features shift locations until reaching the spot indicated by red stars (training instances) and by green diamonds (testing instances). Using the transformation from [12] does not guarantee that every value will be included. On the other hand, using the current proposal, the hyper-heuristic can adapt to the data presented in the training instances.

Figure 6 shows both transformations. The idea is to map values within a given range to the full feasible interval, i.e., $[0, 1]$. In the Linear case (top), the way in which the feature is distributed remains unaffected by using Eq. 8. In the S-shaped case (bottom), extreme values are smoothed out and the middle region is highlighted via Eq. 9.

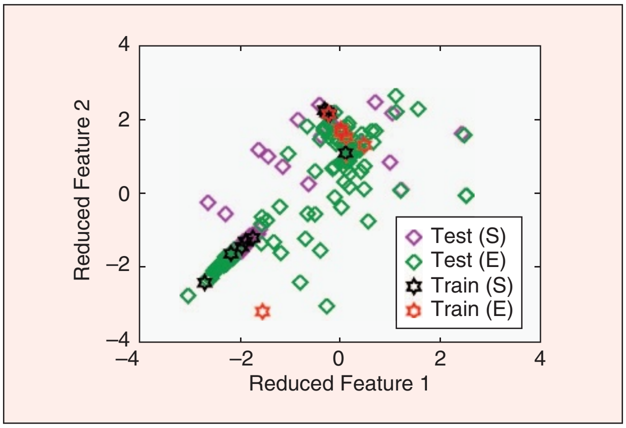


FIGURE 5 Plot of initial (S) and final (E) features used in this work. Data have been reduced from eight to two features by using PCA.

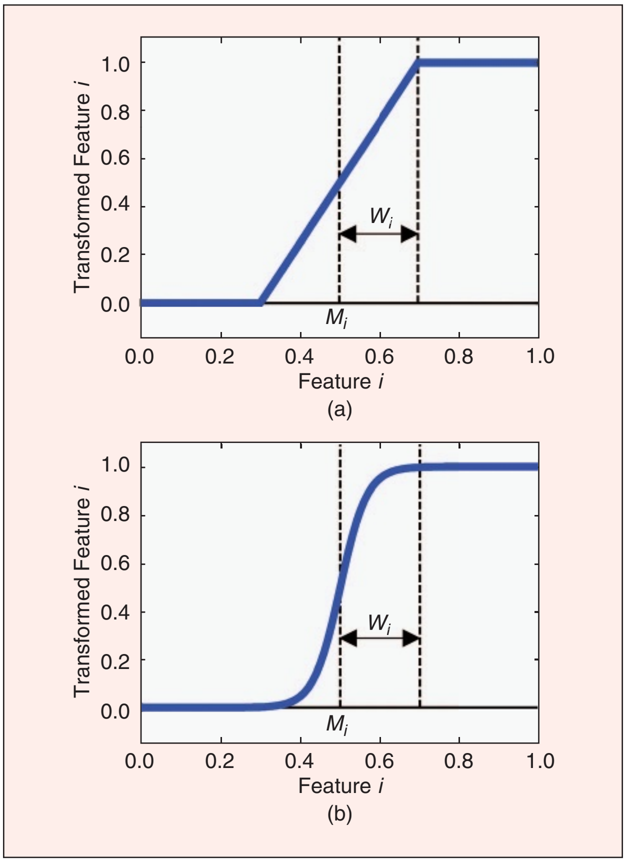


FIGURE 6 Overview of the Linear (top) and S-shaped (bottom) transformations used in this work. M_i is the center of the transformation and W_i represents its half-width.

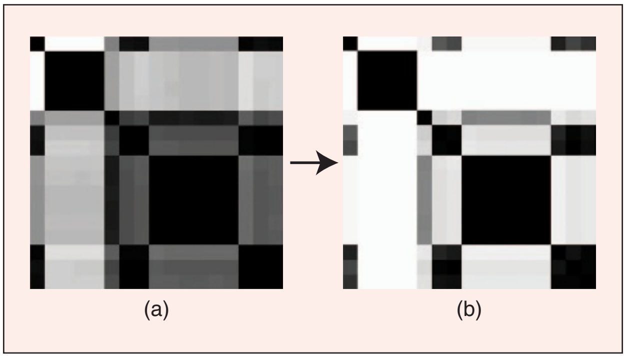


FIGURE 7 VAT image using Euclidean distance (a) and kernel-based distance (b).

$$\Phi_L(x_i, M_i, W_i) = \max\left(0, \min\left(1, \frac{x_i - M_i + W_i}{2W_i}\right)\right) \tag{8}$$

$$\Phi_S(x_i, M_i, W_i) = 1 - \left(\frac{1}{1 + e^{\frac{6M_i}{W_i}\left(\frac{x_i}{M_i} - 1\right)}}\right). \tag{9}$$

B. Proposed Implicit Feature Transformation

The above methods preserve low-dimensionality of the problem. This, however, can be a shortcoming when dealing with a complex distribution of instances, e.g., when instances best solved by different heuristics are very close. One way to ease this issue is by mapping to a higher dimensional space. To do so, a selection hyper-heuristic must compute the similarity between the set of rules and an instance, in this new feature space. Without loss of generality, here we assume the squared Euclidean distance. Let $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ be two points in the original m-dimensional feature space. Expanding the polynomial of the squared Euclidean distance reveals that, in the new feature space, this value can be computed by the inner products of the mapped instances. Thus, we use the kernel trick to perform this mapping since kernel functions allow performing an implicit mapping of features, yielding Eq. 10.

$$d^2(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = K(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) - 2K(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) + K(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}). \tag{10}$$

Using kernels is a more general approach since it subsumes the previous ones. Indeed, it can be shown that the linear kernel corresponds to the original feature space. In this paper, we focus on the Radial Basis Function (RBF) kernel, because it is one of the most popular and effective in the literature [27]. The RBF kernel, Eq. 7, has one adjustable parameter, which we set as $\gamma = 1/N_f$, where N_f is the number of features.

The RBF kernel can be regarded as a similarity measure between two instances. One way to visualize its effect is through Visual Assessment of similarity Tendency (VAT) images [28]. In this kind of images, each element is compared to all others using a similarity measure. Data is usually stored in matrix form (sorted by each cluster of data). Therefore, each value in the matrix represents the similarity between the element given by the row and the one given by the column. Afterwards, an image is created to reflect each similarity value in the matrix with a color: the darker the color, the more similar elements are. Thus, black patches indicate groups of data as similar as possible, while white regions indicate maximum dissimilarity between the pair of elements.

Figure 7 compares a VAT image with the Euclidean distance (left) against one with the RBF kernel (right). This evidences the benefit of using a kernel (since regions are more clearly separated). Therefore, this approach swaps the traditional "distance calculation" block from Fig. 1, by one using the kernel shown in Eq. 7. Since selection hyper-heuristics choose a heuristic based on the similarity to a set of rules, having too many can lower their performance by creating overlapped rules. Kernel methods can overcome this issue by their implicit mapping to a higher dimensional space (Fig. 8).

IV. Methodology

We followed a four-stage methodology (Fig. 9). Also, we considered the same 322 instances from [12] split in the same way: 5% for training and 95% for testing. Data are publicly available[1], and can be identified as: `geom, ehi-85` and `bqwh-15-106`.

A. Preliminary Testing

This first stage tries to determine an eventual gain from using transformations to enhance the predictive power of features. Therefore, we study the distribution of original and explicitly transformed features. We also study their effect on the zones of influence of each training instance. It is worth remarking that, at this stage, the eventual gain from the kernel-based approach cannot be estimated, since its calculations are done on the fly.

B. Initial Testing

A second stage selects the features used in [12] (i.e., p_1 and p_2). With this, we create 15 different selection hyper-heuristics for

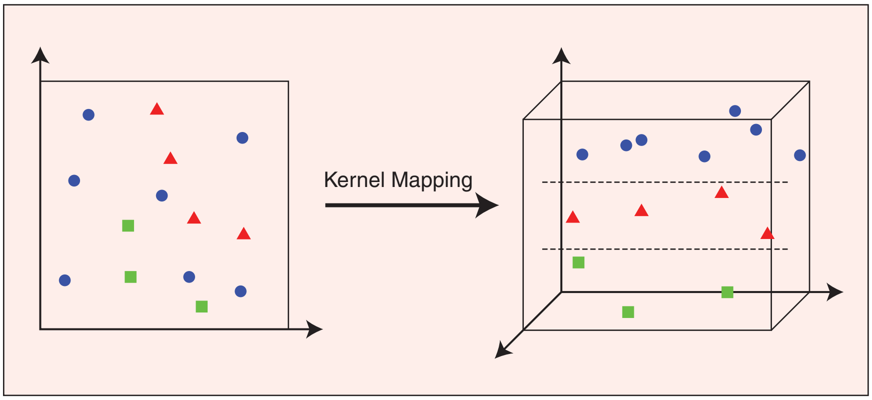


FIGURE 8 Separating the set of rules from a low dimensional space through a kernel mapping to a higher dimensional one.

[1] https://www.cril.univ-artois.fr/~lecoutre/benchmarks.html

each scenario (i.e., with no transformation and for the Linear, Exponential, S-shaped, and RBF kernel). We also include experiments for the combination of kernel with Linear and S-shaped transformations. This leads to 105 selection hyper-heuristics (15 × 7). First, we present their average performance. We compare it against standalone heuristics and baseline selection hyper-heuristics (i.e., with no transformation). Moreover, we compare our data against that of a synthetic oracle, which can perfectly predict the best solver for tackling each instance. Because such solver does not exist, we build it by using the best solution (among the base heuristics) for each instance. We consider it important to highlight that such a process is infeasible for a real application, but useful as a benchmark.

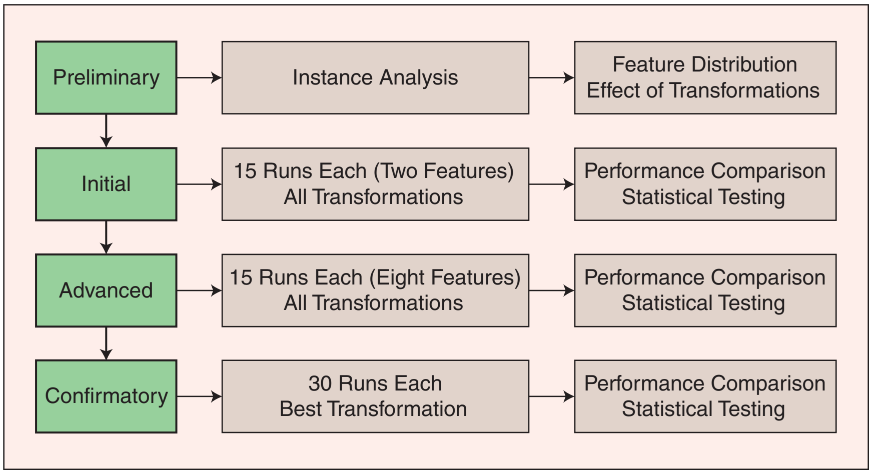


FIGURE 9 Overview of the four-stage methodology adopted in this work.

As a second approach, we analyze how stable the data are. We focus on the success rate and on the search cost of each selection hyper-heuristic (see Sect. II-E). We wrap this section up with a one-tailed Wilcoxon statistical test to determine significant increases in the performance of each approach.

C. Advanced Testing

This stage deepens the previous one by analyzing the effect of transformations over the whole set of eight features (see Sect. II-E). Again, we run 15 repetitions of each experiment. We also determine the performance gain of using each transformation. Moreover, we execute a one-tailed Wilcoxon statistical test to determine whether a significant performance increase can be achieved.

D. Confirmatory Testing

In this final stage, we explore the generality of our proposed approach. Therefore, we select a different domain and generate 30 base selection hyper-heuristics and 30 with the best transformation (more about this in Section V-D). In this work, we chose the knapsack problem, mainly due to its popularity and usefulness, and because knowledge about this combinatorial optimization problem is widespread. Our tests consider two sets of 600 instances each: one with 50 items, and one with 100 items. Each set was built up with the instances from [29], covering groups 11 to 16. In accordance with previous tests, we trained each selection hyper-heuristic using 5% of the instances (i.e., 30).

In this stage, selection hyper-heuristics can select among four popular heuristics. The first three select the item based on the maximum profit, the minimum weight, or the best profit/weight ratio, respectively. The fourth one selects items in their default order. Moreover, selection hyper-heuristics map an instance based on seven features calculated over the items remaining in the instance. Three of them use information from the profit, and correspond to the mean, median, and standard deviation. Another three correspond to the mean, median, and standard deviation of the weight. The final one is a measure of the correlation between profit and weight. It is important to highlight that each metric is normalized so that their values fall in the $[0, 1]$ range. Therefore, the first three features are divided by the maximum profit within the instance. The next three ones are, thus, divided by the maximum weight. The final feature is increased by one and divided in half. Please bear in mind that the maximum values are dynamic as they are calculated from the items remaining in the instance.

Lastly, but not less important, it is worth mentioning that total profit is used as the metric for two events. The first one is training the selection hyper-heuristics. The second one is assessing the eventual performance gain derived from the transformation. Also, it is important to highlight that the aforementioned profit corresponds to the sum of profits achieved on each instance. As such, throughout training profit is calculated over 30 instances, but it is calculated over 570 throughout testing. As before, we run a one-tailed Wilcoxon statistical test to determine whether a significant increase in profit can be achieved.

V. Results and Discussion

This section presents the main results of our work. To make things easier for the reader, the structure presented in the methodology (Sect. IV) is preserved, reserving one subsection for each main stage.

A. Preliminary Testing

Figure 10 shows the distribution of all features in the training set, and their transformations. In most cases, the S-shaped transformation expands representative data more than the Linear transformation does. Also, in all cases the median of the former was lower than that of the latter. Another thing worth mentioning at this point is that the model used in a previous work does not expand the feature range from zero to unity. However, it allows for values all the way to zero (see Figure 4). This may provide it with more flexibility for advanced stages of the search where features may migrate to lower regions.

Figure 11 shows the regions that each instance influences. Data are shown for the original features (left), and for the ones

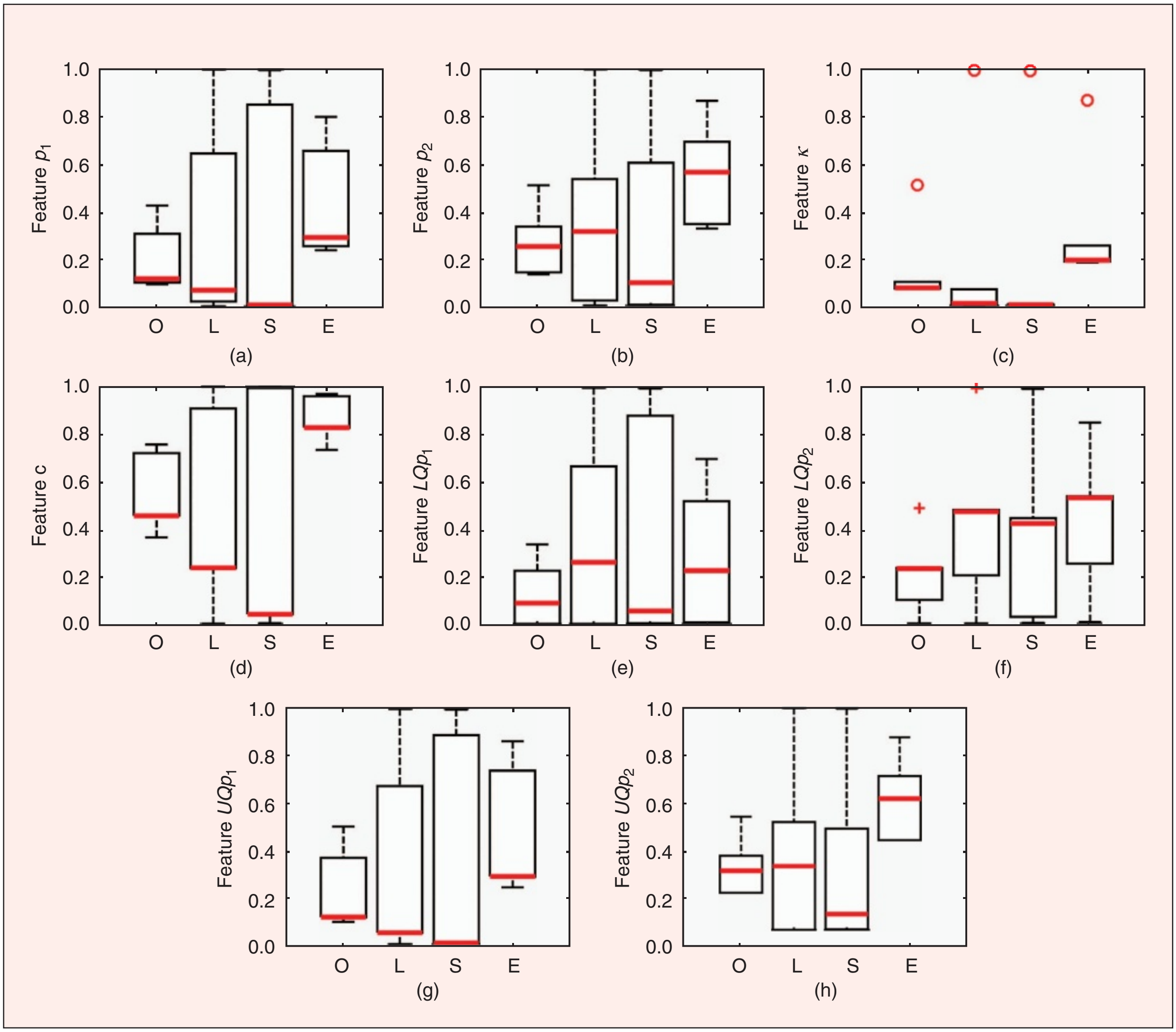


FIGURE 10 Distribution of all features: Original (O) values and those with Linear (L), S-shaped (S), and Exponential (E) transformations. Crosses: outliers between $1.5 * IQR$ and $3 * IQR$. Circles: outliers beyond $3 * IQR$. IQR: Interquartile range.

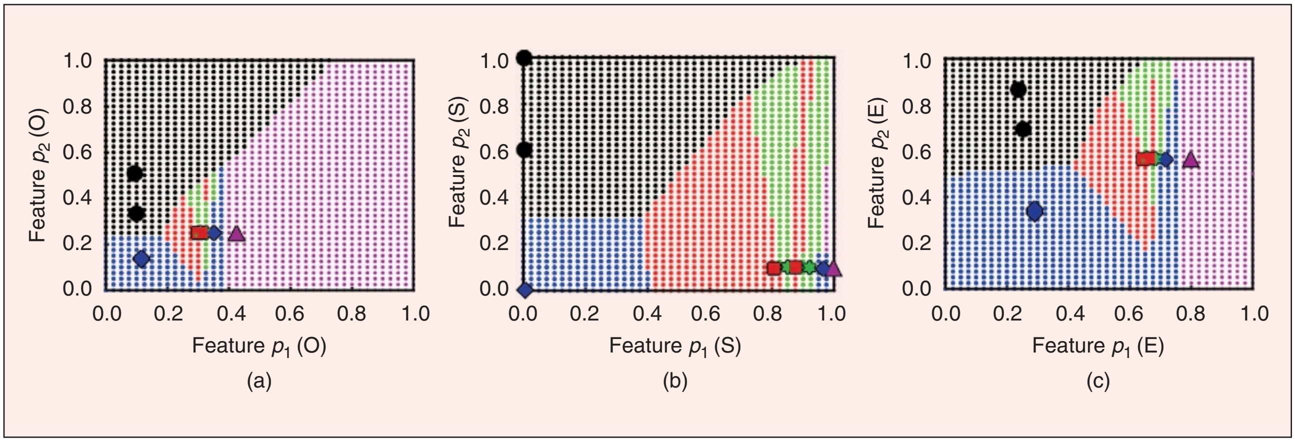


FIGURE 11 Example of how transformations affect the regions of influence created by each instance, considering two features. Data are shown for the Original (O) values and those with S-shaped (S) and Exponential (E) transformations. Elements in magenta represent ties.

transformed using the S-shaped (middle) and the previously reported (right) approaches. As shown, the region beyond $p_1 = 0.6$ and $p_2 = 0.6$ is unused, and thus wasted. On the contrary, there is a region at $0.3 \leq p_1 \leq 0.5$ and $p_2 = 0.2$ (approximately) where different kinds of instances are mixed up. By using both transformations, the wasted space is reduced and distributed throughout the remaining regions, broadening the zones in conflict and thus making them easier to separate.

B. Initial Testing

A comparison of each base heuristic against a synthetic oracle reveals that the latter performs a lot better (Fig. 12). Even so, this only implies that there is a latent benefit derived from an appropriate combination of each heuristic. Nonetheless, Oracle data were generated in a synthetic fashion by analyzing the performance of each standalone heuristic at each instance and selecting the best one. Thus, it represents a Utopian scenario where a perfect selection was carried out. In spite of this, and as it was expected, all selection hyper-heuristics (including those with no transformation) performed better than standalone heuristics. Moreover, all transformations exhibited an average performance quite close to that of the synthetic oracle (highlighted bar in green) in terms of both, number of adjusted consistency checks and success rate. Even so, the S-shaped transformation completed, on average, a bit more instances than the other approaches (highlighted bar in blue). The kernel-based approach was computationally cheapest (highlighted bar in red). It is also important to remark that, even the worst transformations were not so bad. In fact, the Linear transformation yielded a success rate 3% higher than the best performing heuristic (i.e., DOM) while requiring about 60% less ACC. Besides, the combination of kernel with S-shaped transformations required only 7% more ACC than the best heuristic (i.e., KAPPA) but increased the success rate in 22%. A comparison of the average behavior of selection hyper-heuristics with no transformation is also interesting. For this, we focus on search cost and on success rate. The best transformations shifted the success rate by 8% and by 10% (respectively), while requiring about 20% less ACC (in both cases).

Selection hyper-heuristics were distributed in an interesting fashion (see Fig. 13). For starters, all transformations increased the median success rate in over 10%. Two of them (S-shaped and kernel) also reduced the median cost of the search in about 20,000 consistency checks. Alas, the Linear and the previously proposed transformations (identified as 'Exponential') led to a more computationally expensive search path, increasing the median number of consistency checks. However, this number corresponds to the number of validations that must be performed in those instances where the solver could find a solution within the time limit. Therefore, this increase in cost could be derived from the additional instances that were solved. Figure 13 also includes data for the combinations of explicit and implicit transformations. We did so striving to analyze whether merging them led to a better performance. As shown, even if the performance improves, only the Linear transformation yields a behavior similar to that of the kernel. Though seemingly enhancing the success rate, it hinders the search cost. For the S-shaped transformation, however, including the kernel-based distance hampers performance, leading to less successful and more costly selection hyper-heuristics.

A Wilcoxon statistical test yielded the p-values shown in Table 1. The S-shaped and both mixed (i.e., K+L and K+S) transformations had a higher success rate than the original approach (p-values below 0.05). Even so, the p-value of the pure kernel (0.0548) was not too high, which makes it an alternative worth keeping in mind. Analyzing the search cost

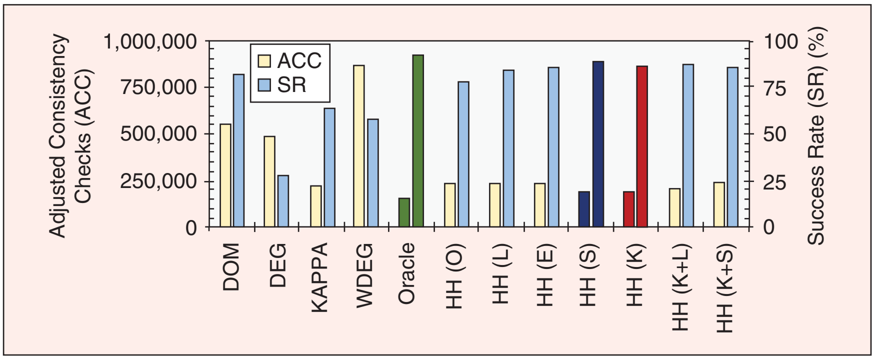


FIGURE 12 Average number of adjusted consistency checks (left bars) and success rate (right bars) for all base heuristics, for a synthetic oracle, and for all hyper-heuristics (average of 15 runs each) with two features: Original (O), Linear (L), Exponential (E), S-shaped (S), Kernel (K), Kernel+Linear (K+L), and Kernel+S-shaped (K+S). Highlighted columns: Synthetic Oracle (green), best adjusted consistency checks (red) and success rate (blue). Data distribution is shown in Fig. 13.

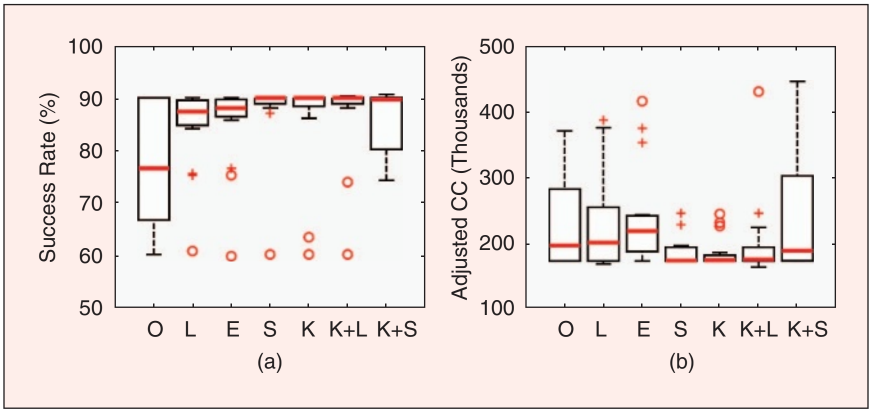


FIGURE 13 Success rate (a) and search cost (b) for 15 runs of hyper-heuristics operating with two features. Original values (O) and transformations: Linear (L), S-shaped (S), Exponential (E), Kernel (K), Kernel+Linear (K+L), and Kernel+S-shaped (K+S). Crosses: outliers between $1.5 * IQR$ and $3 * IQR$. Circles: outliers beyond $3 * IQR$. *IQR*: Interquartile range.

TABLE 1 **P-values of the Wilcoxon statistical test (two features), between the original approach and each modification. L: Linear. E: Exponential. S: S-shaped. K: Kernel. K+L: Kernel+Linear. K+S: Kernel+S-shaped. SR: Success rate. ACC: Adjusted consistency checks.**

METRIC	L	E	S	K	K+L	K+S
SR	0.2107	0.2502	0.0325	0.0548	0.0343	0.0492
ACC	0.4098	0.7466	0.0855	0.2214	0.1313	0.5659

TABLE 2 **P-values of the Wilcoxon statistical test (eight features), between the original approach and each modification. L: Linear. E: Exponential. S: S-shaped. K: Kernel. K+L: Kernel+Linear. K+S: Kernel+S-shaped. SR: Success rate. ACC: Adjusted consistency checks.**

METRIC	L	E	S	K	K+L	K+S
SR	0.1577	0.0276	0.3145	0.0003	0.5335	0.2858
ACC	0.2668	0.5000	0.1497	0.1403	0.1595	0.5659

reveals that no approach boasted a huge improvement over the original one. Nonetheless, the S-shaped transformation performed best, with a p-value of 0.0855.

C. Advanced Testing

Migrating to more features worked in favor of selection hyper-heuristics (Fig. 14). The median SR of all solvers increased to about 90% (even that of the selection hyper-heuristic with no transformation). Nonetheless, all transformations were still helpful since they increased the stability of this metric (i.e., SR), yielding only a few outliers with poor performance. The median number of required consistency checks was also reduced in about 50,000. Moreover, it was reduced to about one half for the worst performing hyper-heuristic. Even so, and similarly than with two features, the kernel-based approach was the one which reduced the most the variation in the cost of the search. Furthermore, again, combining both kinds of approach yielded mixed results. In fact, it was helpful for the Linear transformation. It is, however, not so much for the S-shaped one, since they became less successful and more costly than without the kernel.

This time around, the Wilcoxon statistical test revealed the p-values shown in Table 2. As can be seen, the pure kernel approach performed remarkably well (p-value of 0.0003). This makes it virtually safe to assume that its success rate is significantly higher than that of the original approach. Another result worth mentioning is that of the exponential transformation (p-value of 0.0276). Despite this, no approach provided a p-value below 0.10 when testing the search cost. Nevertheless, the pure kernel approach exhibited a standard deviation reduced by almost 40%.

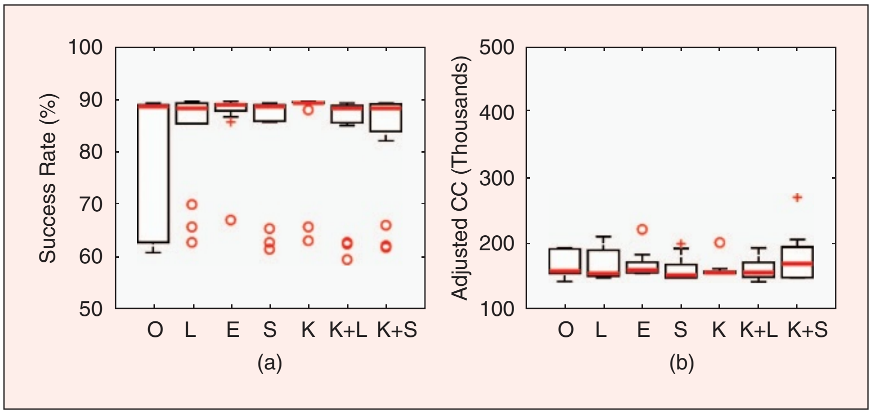


FIGURE 14 SR (a) and ACC (b) for 15 runs of selection hyper-heuristics operating with eight features. Original values (O) and transformations: Linear (L), S-shaped (S), Exponential (E), Kernel (K), Kernel+Linear (K+L), and Kernel+S-shaped (K+S). Crosses: outliers between 1.5 * *IQR* and 3 * *IQR*. Circles: outliers beyond 3 * *IQR*. *IQR*: Interquartile range.

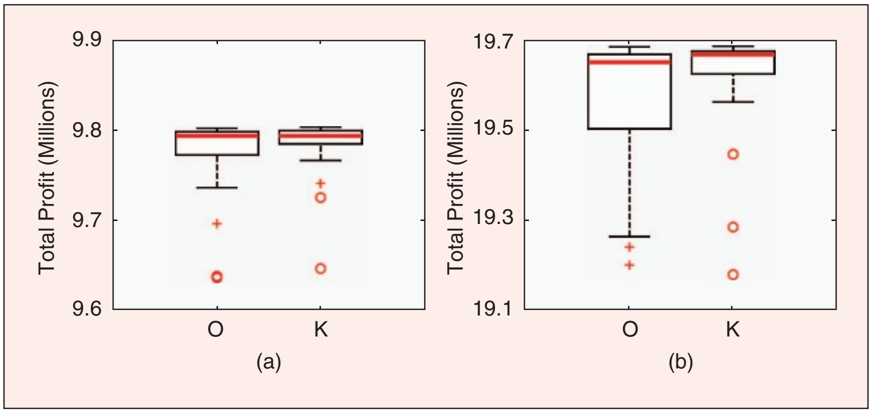


FIGURE 15 Total profit distribution for the original (O) and kernel-based (K) selection hyper-heuristics (30 runs each). a: instances with 50 items. b: instances with 100 items.

D. Confirmatory Testing

The pure kernel transformation was the only one that proved to be statistically better than the original approach during the previous phases. Hence, we selected it as our best approach and used it for this stage. Figure 15 shows the distribution of the performance achieved by 30 selection hyper-heuristics with no transformation (O), and by 30 with the kernel-based distance (K). The standard deviation of the profit was reduced by almost 30% (going from 44,275 to 31,900), for instances with 50 items (left). For instances with 100 items (right), it was reduced by almost 20% (going from 143,175 to 118,864). Moreover, statistical evidence (p-value of 0.02073) supports the claim that kernel-based distance produces, on average, more competent heuristics than Euclidean distance. Our data suggests that, as the problem increases in difficulty (represented by more items), the approach that relies on the kernel-based distance remains more stable.

VI. Conclusions and Future Work

Throughout this work we defined two approaches for carrying out an explicit transformation of features, and one for doing so implicitly. Our aim was to improve the performance of selection hyper-heuristics. We tested these ideas, and their combinations, on the widely used domain of CSPs. We found that, when considering two features, all transformations help the selection hyper-heuristic to generate solvers that perform closer (on average) to a synthetic oracle. Moreover, the S-shaped and the kernel-based transformations increased the median SR in about 15% while decreasing the median search cost in about 20,000 ACC. This means that both approaches led to a higher number of instances being solved while reducing the cost of solving them. Even so, combining both ideas did not prove fruitful. Linear transformation was the only one enhanced by adding the kernel, improving its success rate but hindering its search cost.

Increasing the number of features up to eight did not work as well for the S-shaped transformation as it did for the kernel. This time around the median success rate of both approaches lurked around 90%, but only the latter exhibited almost no variation (except for a few outliers). Similarly, the search cost was better through both approaches. In fact, the median search cost of the S-shaped transformation was the lowest one, being 5,000 adjusted consistency checks lower than that without the transformation. Only the one using kernel had a small variation, exhibiting a standard deviation almost 40% lower than that of the original approach.

Because of the aforementioned, we determined that the pure Kernel approach was the best way of incorporating transformations into selection hyper-heuristics. A confirmatory test run on the Knapsack domain considered instances with 50 and 100 items. Data revealed that the standard deviation of the profit achieved by hyper-heuristics could be reduced by almost 30% and 20% (respectively), making transformations a worthwhile effort. Moreover, a statistical test confirmed a significant increase in the performance of selection hyper-heuristics for the set with 100 items. Thus, we recommend following this idea and applying it to different domains and under different conditions, as to better assess how its benefits propagate.

In this work, we defined γ as the inverse of the number of features. However, some exploratory tests (omitted due to space restrictions) revealed that this may not always be the best approach for selecting it. Therefore, a future research avenue in this path could relate to improving the kernel-based transformation. This could be done by designing a procedure that tailors γ to each problem domain. In other words, one that tailors it to different configurations, e.g., by using information not only from the number of features but also from their nature. Another path worth following is to carry out a more extensive testing, e.g., by increasing the number and variety of instances and features. Such testing should include a careful analysis of all the different combinations of features, focusing on the effect of feature transformations on the performance of hyper-heuristics.

Acknowledgment

This research was partially supported by CONACyT Basic Science Projects under grants 241461, 221551 and 287479, and ITESM Research Group with Strategic Focus in Intelligent Systems.

References

[1] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *J. Oper. Res. Soc.*, vol. 64, no. 12, pp. 1695–1724, Dec. 2013.

[2] J. C. Ortiz-Bayliss, H. Terashima-Marín, and S. E. Conant-Pablos, "Learning vector quantization for variable ordering in constraint satisfaction problems," *Pattern Recognit. Lett.*, vol. 34, no. 4, pp. 423–432, Mar. 2013.

[3] T. N. Ferreira, J. A. P. Lima, A. Strickler, J. N. Kuk, S. R. Vergilio, and A. Pozo, "Hyper-heuristic based product selection for software product line testing," *IEEE Comput. Intell. Mag.*, vol. 12, no. 2, pp. 34–45, Apr. 2017.

[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1157–1182, Mar. 2003.

[5] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, vol. 72. New York: Springer, 2014.

[6] B. Xue and M. Zhang, "Evolutionary computation for feature manipulation: Key challenges and future directions," in *Proc. IEEE Congr. Evolutionary Computation*, Vancouver, Canada, July 2016, pp. 3061–3067.

[7] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[8] D. Pyle, *Data Preparation for Data Mining*, 1st ed. San Francisco, CA: Morgan Kaufmann, 1999.

[9] K. Smith-Miles and L. Lopes, "Measuring instance difficulty for combinatorial optimization problems," *Comput. Oper. Res.*, vol. 39, no. 5, pp. 875–889, May 2012.

[10] M. Montazeri, "HHFS: Hyper-heuristic feature selection," *Intell. Data Anal.*, vol. 20, no. 4, pp. 953–974, June 2016.

[11] E. Hart, K. Sims, K. Kamimura, and B. Gardiner, "A hybrid method for feature construction and selection to improve wind-damage prediction in the forestry sector," in *Proc. Genetic Evolutionary Computation Conf.*, Berlin, Germany, July 2017, pp. 1121–1128.

[12] I. Amaya, J. C. Ortiz-Bayliss, A. E. Gutiérrez-Rodríguez, H. Terashima-Marín, and C. A. Coello Coello, "Improving hyper-heuristic performance through feature transformation," in *Proc. IEEE Congr. Evolutionary Computation*, San Sebastian, Spain, June 2017, pp. 2614–2621.

[13] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.

[14] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[15] J. R. Rice, "The algorithm selection problem," in *Advances in Computers*, vol. 15. 1976, pp. 65–118.

[16] E. O'Mahony, E. Hebrard, A. Holland, C. Nugent, and B. O'Sullivan, "Using case-based reasoning in an algorithm portfolio for constraint solving," in *Proc. Irish Conf., Artificial Intelligence and Cognitive Science,* Cork City, Ireland, Aug. 2008, pp. 210–216.

[17] Y. Malitsky and M. Sellmann, "Instance-specific algorithm configuration as a method for non-model-based portfolio generation," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. CPAIOR 2012. Lecture Notes in Computer Science*, N. Beldiceanu, N. Jussien, and É. Pinson, Eds. Berlin, Germany: Springer, 2012, pp. 244–259.

[18] J. C. Ortiz-Bayliss, H. Terashima-Marín, and S. E. Conant-Pablos, "Combine and conquer: An evolutionary hyper-heuristic approach for solving constraint satisfaction problems," *Artif. Intell. Rev.*, vol. 46, no. 3, pp. 327–349, Oct. 2016.

[19] J. G. Marín-Blázquez and S. Schulenburg, "Multi-step environment learning classifier systems applied to hyper-heuristics," in *Proc. 8th Annu. Conf. Genetic and Evolutionary Computation*, Seattle, WA, July 2006, pp. 1521–1528.

[20] J. C. Ortiz-Bayliss, H. Terashima-Marín, and S. E. Conant-Pablos, "Using learning classifier systems to design selective hyper-heuristics for constraint satisfaction problems," in *Proc. IEEE Congr. Evolutionary Computation*, Cancun, Mexico, June 2013, pp. 2618–2625.

[21] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[22] J. Berlier and J. McCollum, "A constraint satisfaction algorithm for microcontroller selection and pin assignment," in *Proc. IEEE SoutheastCon*, Concord, MA, Mar. 2010, pp. 348–351.

[23] S. V. Bochkarev, M. V. Ovsyannikov, A. B. Petrochenkov, and S. A. Bukhanov, "Structural synthesis of complex electrotechnical equipment on the basis of the constraint satisfaction method," *Russian Elect. Eng.*, vol. 86, no. 6, pp. 362–366, June 2015.

[24] I. P. Gent, P. Prosser, and T. Walsh, "The constrainedness of search," in *Proc. Association Advancement Artificial Intelligence Conf. '96*, Portland, OR, Aug. 1999, pp. 246–252.

[25] S. A. Bittle and M. S. Fox, "Learning and using hyper-heuristics for variable and value ordering in constraint satisfaction problems," in *Proc. ACM 11th Annu. Conf. Companion Genetic Evolutionary Computation: Late Breaking Papers,* Quebec, Canada, July 2009, pp. 2209–2212.

[26] F. Boussemart, F. Hemery, C. Lecoutre, and L. Sais, "Boosting systematic search by weighting constraints," in *Proc. European Conf. Artificial Intelligence,* Valencia, Spain, Aug. 2004, pp. 146–150.

[27] A. J. Bagnall and G. C. Cawley, "On the use of default parameter settings in the empirical evaluation of classification algorithms," *CoRR*, vol. abs/1703.06777, Mar. 2017.

[28] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in *Proc. Int. Joint Conf. Neural Networks*, Honolulu, HI, May 2002, vol. 3. pp. 2225–2230.

[29] D. Pisinger, "Where are the hard knapsack problems?" *Comput. Oper. Res.*, vol. 32, no. 9, pp. 2271–2284, Sept. 2005.

Is Evolutionary Computation Evolving Fast Enough?

IMAGE LICENSED BY INGRAM PUBLISHING

Graham Kendall
*School of Computer Science,
University of Nottingham,
UK and University of Nottingham
Malaysia Campus, MALAYSIA*

Abstract—Evolutionary Computation (EC) has been an active research area for over 60 years, yet its commercial/home uptake has not been as prolific as we might have expected. By way of comparison, technologies such as 3D printing, which was introduced about 35 years ago, has seen much wider uptake, to the extent that it is now available to home users and is routinely used in manufacturing. Other technologies, such as immersive reality and artificial intelligence have also seen commercial uptake and acceptance by the general public. In this paper we provide a brief history of EC, recognizing the significant contributions that have been made by its pioneers. We focus on two methodologies (Genetic Programming and Hyper-heuristics), which have been proposed as being suitable for automated software development, and question why they are not used more widely by those outside of the academic community. We suggest that different research strands need to be brought together into one framework before wider uptake is possible. We hope that this position paper will serve as a catalyst for automated software development that is used on a daily basis by both companies and home users.

Digital Object Identifier 10.1109/MCI.2018.2807019
Date of publication: 10 April 2018

I. Introduction

Evolutionary Computation (EC) has been part of the research agenda for at least 60 years. In a typical EC algorithm, a population of potential solutions is created and they compete for survival. The weakest (less fit) members of the population are killed off, and the remaining members are retained and copies made, which are mutated. This new population is then evaluated with the expectation that the population's average fitness improves over time, along with the best performing individual solution.

It is debatable whether EC has had the impact in the commercial sector that other technologies have had, which have seen much more visible adoption. 3D printing is changing the way that manufacturing is done and is also moving into the home, to the extent that almost anybody can carry out 3D printing. Immersive reality is on the verge of changing society, in ways that are not totally clear yet. What is apparent is that applications such as Pokemon Go have sparked interest into the challenges and opportunities that immersive reality brings [1]–[5]. Ubiquitous computing is becoming more prevalent, enabling users to access computing resources in ways that were unimaginable even just a few years ago. Artificial Intelligence (AI) is becoming part

Corresponding Author: Graham Kendall (Email: Graham.Kendall@nottingham.ac.uk)

1556-603X/18©2018IEEE

of our daily lives, whether that is competing against the best human players in board games [6] or helping make self driving cars a reality [7]–[9]. EC has not had the same penetration as other technologies. A specific example we draw upon in this paper is large scale software development which is, arguably, where EC is most needed.

Writing a scientific paper that utilizes an evolutionary approach based on a real world problem is not the same as using an evolutionary approach to address a real world problem.

In this paper, we will look back at what EC promised and will suggest some challenges that, if addressed, might further advance EC and enable its wider adoption.

Writing a scientific paper that utilizes an evolutionary approach based on a real world problem is not the same as using an evolutionary approach to address a real world problem. This may seem a pedantic statement but a paper considering a problem that is drawn from the real world is not the same as addressing the actual problem faced by industry. Looking at a sample of EC papers, which are labeled as "real world applications" (see Related Work, Section II), often shows that the problem being tackled is a problem that would be recognized as a real world problem. However, the algorithm is often tested on benchmark datasets and/or uses a simplified model of the problem. It is our view that if a problem is presented as a real world problem, there should be an underlying model that addresses a real problem faced by the user community, rather than a simplified model that is an abstraction of the problem that users actually face.

We do recognize the importance that benchmark datasets play in the investigation, and development of, algorithmic approaches. Indeed, many important breakthroughs have been reported by investigating these real world abstractions. We also note that studying these simplified problems enables easier analysis of the results. We are also aware that this type of research (using a simplified problem) is why games are often used, as they have fixed rules, the rules are unambiguous and there is a winner and a loser. It is no coincidence that Chess has been called the drosophila of artificial intelligence [10]. We are also conscious that abstracting a problem so that the focus is on the methodology is good scientific research and that fully modeling a specific problem may not be of benefit to the wider academic community.

So, we are not critical of using abstractions of the real world but it is not conducive to promoting EC to the commercial sector, who require solutions to problems which go beyond these benchmarks, and which address the needs of their specific business.

The fact that the EC research community does not tend to tackle real, real world problems is partly (largely?) due to the industrial/university communities not working together. This is not a criticism of the companies, or the universities. Universities and companies often have different objectives (e.g. carrying out research vs. making a profit) and they work on different time scales (e.g. long term research projects vs developing new products to maintain a competitive edge). Another contributory factor may be that other methodologies may be better accepted by the commercial sector as they are easier to understand, implement and support. In this respect, the use of methodologies such as EC can be seen to be similar to a reluctance to utilize Artificial Neural Networks. The decisions they reach are not easy to understand so that the commercial sector is often unwilling to adopt them, preferring methodologies where the decisions can be more easily explained.

This paper is structured as follows. In the next section we consider related work, focusing on those papers that have reported using EC on real world problems. In Section III we ask if the potential of EC has been achieved? We note the significant achievements of the EC pioneers and ask why their seminal work has not translated into more uptake outside of the scientific community. Section IV looks at Genetic Programming (GP), bearing in mind it was probably this EC methodology that had (has?) the most promise to be used in the commercial sector. In Section V we consider a more recent methodology (Hyper-heuristics) which also has the potential to be used by industry. In Section VI we look specifically at Large Scale Software Development, highlighting some of its high profile failures and asking if/how EC can help in this area. In Section VII we present some suggested research directions, before concluding in Section VIII.

II. Related Work

The topic of deploying evolutionary algorithms in the real world has been studied before [11]. Within the context of this paper, [11] provides a number of inhibitors to using EC based algorithms in the real world. These include:

- ❏ The features of real world problems
- ❏ The lack of faith in the underlying model that represents the problem meaning that companies have little confidence in the solutions that are produced
- ❏ The fact that EC algorithms are not integrated within an overarching framework to assist with areas such as parameter settings
- ❏ The lack of the required skills of the developers
- ❏ Resistance to change

It is difficult to track down examples in the scientific literature where EC has been deployed in a company and is used as a matter of course in their routine activities. Of course, there will be examples that the scientific community is not aware of due to in-house research activity, commercial sensitivities or the time pressures within a company to write and present the contribution to the scientific community, but we do not believe that the use of EC has had large adoption within commercial companies. There are many examples (e.g. [12]–[14]) where real data

There are number of inhibitors to using EC based algorithms in the real world, including the features of real world problems.

has been used and the results are encouraging, but the algorithm has not been used by a company to support its on-going business activities. It is often the case that a company provides data, which is utilized to study the problem, it is then reported in the scientific literature but the algorithmic methodology is not used by the company.

A Genetic Algorithm (GA) [15] was reported in [16], presenting a two-phase algorithm for the "bid-line generation problem" (the problem of scheduling airline crew) for Delta Air Lines. As with many staff scheduling problems, there are many industry and legal factors to take into account [17], [18] so any systems that are developed for a given company are often bespoke. The first phase of their algorithm generates as many high-quality lines as possible. The second phase, where the GA is run, completes the assignments. The schedules that are produced were shown to be of comparable quality to those that were generated using a semi-automatic process that the airline had previously used. It is interesting to note that the four authors listed their affiliations as Delta Technology or Delta Airlines, indicating that the paper was written without a university collaborator. This, we feel, is important as it demonstrates that the industrial sector is deploying EC algorithms. However, many companies will not report these successes in the scientific literature for a number of reasons including lack of time, no pressure to publish and commercial confidentiality. These factors are likely to misrepresent the real scale of industrial take up of EC methodologies in industry, and the lack of reporting in the scientific literature could slow down progress.

Sundararajan et al. [19] considered the cross selling of loans in the banking sector, specifically the GEMB bank in Poland. They used a GA, within an overall framework which draws on different methodologies, which focussed on a predictive model for response, risk and profit. The GA that was developed was a standard GA with a few enhancements that included elitism and splitting the data into training and validation sets and using solutions from one set to inject into the other set if it finds that it performs well. Similar to [16], this paper also had no authors with a university affiliation.

A GA was also utilized in [20]. The two authors were from Intel, with no university affiliation listed. They presented a model for the Product Line Design and Scheduling Problem. The outer layer of their model was a GA. This handled the resource constraints, scheduling, and financial optimization. An inner layer utilized mathematical programming to optimize product composition. Their new approach replaced a spreadsheet solution which could take days, or even weeks, to carry out what-if analysis.

The gerrymandering problem (the process of manipulating electoral boundaries to gain a political advantage) was addressed in [21]. This was a joint paper between university colleagues and a representative from a government department (Philadelphia Water Department's Office of Watersheds). This paper was written as the result of a competition call. The authors won one of the competition categories which gave them the opportunity to present to the city council. The authors classify their algorithm as a form of Evolutionary Programming, rather than as a GA as they did not use a recombination operator.

A recent paper [22], a collaboration between two universities and Ernst & Young, considered the transportation and scheduling issues for the 2014 Special Olympics USA Games. The problem considered 3,300 athletes with intellectual disabilities, 1,000 coaches and over 70,000 spectators. The athletes competed in 16 sports, across 10 locations, spread over a 30-mile radius. The authors developed a GA to address the problem as exact methodologies were too computationally expensive. The resulting schedules were used during the games.

Another routing problem was addressed in [23], in a paper that did not include a company representative as an author. The paper presents a case study based on a humanitarian scenario, a local branch of the Meals on Wheels Association of America, which provides food to individuals who are in need. The approach adopted interfaces a spreadsheet with a GA and is being used by the Metro Meals on Wheels Treasure Valley. It is noted that the tool could be used anywhere that has access to Google Maps or MapQuest.

Ogris et al. [24] studied a primary school timetabling problem [25] from Slovenia. The paper was co-authored by university researchers and industrial collaborators. Their evolutionary algorithm (there was no crossover operator) comprised three objective functions, which were changed probabilistically. The system was used in three Slovenian primary schools but could easily be adapted to other schools and universities.

Simulated annealing [26] and Tabu search [27] are not classified as evolutionary methodologies, rather they are meta-heuristics [28]. However, they have been used in industrial applications so we thought it was worth briefly mentioning them here. We also note that the leading EC journal (IEEE Transactions on Evolutionary Computation) has previously reported work that includes these methodologies [29], [30], albeit hybridized with an evolutionary algorithm. Simulated annealing and tabu search has been reported as being deployed in industry, including Oil Field Drilling [31], Sports [32]–[35], Vehicle Routing [36], Underground Mine Layouts [37] and Personnel Scheduling [38].

The papers that we discuss above might suggest that there has been a lot of commercial applications of EC but considering that the field has been active for over 60 years, the number of reported applications of EC methodologies is somewhat small. No doubt, there are other papers that we have not included and there will be successes that are not reported in the scientific literature but we still argue that adoption of EC by the commercial sector is not as prevalent as some other technologies.

This might be about to change with recent interest in Deep Learning and the success of projects such as AlphaGo [6], which was able to win against the best Go player in the world, a feat that most people predicted would take another ten years. However, this is just one methodological example and, whilst Deep Learning Neural Networks have a bright future, it still does not answer the question as to why more EC methodologies have not had wider uptake.

III. Has the Potential been Realized?

EC has been an active research area since the 1950's [39], [40]. Many eminent scientists have been recognized as being pioneers in this field, demonstrating the strength in depth of this area. Table 1 shows the IEEE Computational Intelligence Society Pioneers, along with a small sample of their contributions. It is beyond question that these pioneers, along with the wider community, have made significant advances in EC.

When these pioneers were carrying out their early work, it was in their minds that it would be adopted by the wider community. For example, Box [41] says "Its basic philosophy is that it is nearly always inefficient to run an industrial process to produce product alone. A process should be run so as to generate product plus information on how to improve the product." In 1996, Schwefel said "…the past decade has witnessed an exponential increase in diverse applications, from design synthesis, planning and control processes, to various other adaptation and optimization tasks."

It is, perhaps, surprising that we have not seen more examples reported in the scientific literature of EC being deployed in commercial systems. Although the related work section provides some examples, and no doubt some are missing, but given that the field has been active for at least 60 years we might expect to see more examples being reported?

In comparison, 3D printing [91], [92] has seen a significantly faster uptake. The first patent was issued to Charles Hull in 1986, which can be traced back to his original invention from 1983. Since then the technology has seen rapid uptake, to the point where it is now possible to buy a 3D printer for home use. It is likely that we are only just seeing the start of the additive manufacturing technology and it is likely that many replacement parts, rather than being bought at a shop, or on-line, can be downloaded and printed at home. By comparison, the software development industry is not able to offer the home user a way to develop, or evolve, software unless they are already skilled programmers or willing to invest a significant amount of time learning a programming language.

Technologies such as GP and Hyper-heuristics (both discussed below), despite delivering excellent research advances, have not really made the transition from the research environment to a position where the benefits can be experienced by an average home user. In the next two sections, we focus on these two methodologies, though similar analysis could be made of the many other EC variants that have been researched over the years.

TABLE 1 **IEEE Computational Intelligence Society Evolutionary Computation Pioneers.**

YEAR	PIONEER	REFERENCES
2016	MARCO DORIGO	[42]–[44]
2015	THOMAS BÄCK	[45]–[47]
2014	GEORGE BURGIN	[48], [49]
2013	XIN YAO	[50]–[53]
2012	RUSSELL C. EBERHART, JAMES KENNEDY, AND J. DAVID SCHAFFER	[54]–[60]
2011	LARRY J. ESHELMAN	[55]–[57]
2010	DAVID E. GOLDBERG AND JOHN GREFENSTETTE	[61]–[63]
2008	DAVID B. FOGEL	[47], [64]–[68]
2005	KENNETH DE JONG	[69]–[72]
2004	RICHARD FRIEDBERG	[73], [74]
2003	JOHN H. HOLLAND	[61], [62], [75], [76]
2002	INGO RECHENBERG AND HANS-PAUL SCHWEFEL	[45], [77]–[80]
2001	MICHAEL CONRAD	[81]–[83]
2000	GEORGE BOX	[41], [84]
1999	ALEX S. FRASER	[40], [85]–[87]
1998	LAWRENCE J. FOGEL	[39], [48], [49], [88]–[90]

IV. Genetic Programming

Many of the papers that were discussed in Section II utilized GAs, yet GP is, arguably, the EC methodology that is most associated with automated software development.

Introduced by Koza [93]–[95], GP seeks to evolve computer programs and/or evolve functions. Does it matter which it does; evolve programs or functions?

In [96] the authors say (Section 1.1) "In genetic programming we evolve a population of computer programs." In one of the seminal GP papers [93], it states "Automatic programming requires developing a computer program that can produce a desired output for a given set of inputs", which is more akin to suggesting that GP evolves functions, rather than a program. We can debate whether a function (a relationship between a set of inputs and a permissible set of outputs) and a program (a sequence of coded instructions to automate a task on a computer) are the same thing but to the general public if GP is sold as evolving computer programs they will assume that this means that a complete program will be evolved, and not just a function (a mathematical function or a function for a given programming language), which is usually the case. We hasten to add that no criticism is implied, or meant, of the GP pioneers, or other researchers. The terminology has evolved over time and the expressions used in the scientific literature are the ones that are most applicable, or preferred, by the authors of a given paper. We note, as in many areas of EC—and even beyond, such as the heuristic community—that there are no widely accepted terms and definitions in much of the terminology that is used.

However, to the general public saying "evolve computer programs" may indicate that GP is much more general than the state of the art would suggest. There have been advances in moving towards more general environments. The 2016 Human Competitive Awards, the so called "Humies"[1], winner [97] says "Automated transplantation would open many exciting avenues for software development: suppose we could autotransplant code from one system into another, entirely unrelated, system. This paper introduces a theory, an algorithm, and a tool that achieves this." This is certainly a significant contribution to automated program development but there is still a lot of work to do, as acknowledged by the authors, "While we do not claim automated transplantation is now a solved problem, our results are encouraging."

Since 2004, the GP community has been able to compete in the Humies. This annual competition invites entries that report human-competitive results by any form of genetic or evolutionary computation. The entries must satisfy one of the following eight criteria (taken from[1]):

1) The result was patented as an invention in the past, is an improvement over a patented invention, or would qualify today as a patentable new invention.
2) The result is equal to or better than a result that was accepted as a new scientific result at the time when it was published in a peer-reviewed scientific journal.
3) The result is equal to or better than a result that was placed into a database or archive of results maintained by an internationally recognized panel of scientific experts.
4) The result is publishable in its own right as a new scientific result independent of the fact that the result was mechanically created.
5) The result is equal to or better than the most recent human-created solution to a long-standing problem for which there has been a succession of increasingly better human-created solutions.
6) The result is equal to or better than a result that was considered an achievement in its field at the time it was first discovered.
7) The result solves a problem of indisputable difficulty in its field.
8) The result holds its own or wins a regulated competition involving human contestants (in the form of either live human players or human-written computer programs).

The Humies have certainly demonstrated the versatility of GP (see Table 2), along with other EC approaches. However,

TABLE 2 Humies Gold Medal Winners (In some years the gold medal was shared, indicated by "=").

YEAR	ENTRY	REFERENCES
2017	"EXPLAINING QUANTUM CORRELATIONS THROUGH EVOLUTION OF CAUSAL MODELS"	[98]
2016	"AUTOMATED SOFTWARE TRANSPLANTATION"	[97]
2015	"EVOLUTIONARY APPROACH TO APPROXIMATE DIGITAL CIRCUITS DESIGN"	[99]
2014	"GENETIC ALGORITHMS FOR EVOLVING COMPUTER CHESS PROGRAMS"	[100]
2013=	"EVOLUTIONARY DESIGN OF FREECELL SOLVERS"	[101]
2013=	"SEARCH FOR A GRAND TOUR OF THE JUPITER GALILEAN MOONS"	[102]
2012	"GO WITHOUT KO ON HEXAGONAL GRIDS" AND "YVALATH: EVOLUTIONARY GAME DESIGN"	[103]
2011	"GA-FREECELL: EVOLVING SOLVERS FOR THE GAME OF FREECELL"	[104]
2010	"EVOLUTIONARY DESIGN OF THE ENERGY FUNCTION FOR PROTEIN STRUCTURE PREDICTION" AND "GP CHALLENGE: EVOLVING THE ENERGY FUNCTION FOR PROTEIN STRUCTURE PREDICTION" AND "AUTOMATED DESIGN OF ENERGY FUNCTIONS FOR PROTEIN STRUCTURE PREDICTION BY MEANS OF GENETIC PROGRAMMING AND IMPROVED STRUCTURE SIMILARITY ASSESSMENT"	[105]–[107]
2009	"AUTOMATICALLY FINDING PATCHES USING GENETIC PROGRAMMING" AND "A GENETIC PROGRAMMING APPROACH TO AUTOMATED SOFTWARE REPAIR"	[108], [109]
2008	"GENETIC PROGRAMMING FOR FINITE ALGEBRAS"	[110]
2007	"EVOLUTIONARY DESIGN OF SINGLE-MODE MICROSTRUCTURED POLYMER OPTICAL FIBRES USING AN ARTIFICIAL EMBRYOGENY REPRESENTATION"	[111]
2006	"CATALOGUE OF VARIABLE FREQUENCY AND SINGLE-RESISTANCE-CONTROLLED OSCILLATORS EMPLOYING A SINGLE DIFFERENTIAL DIFFERENCE COMPLEMENTARY CURRENT CONVEYOR" AND "NOVEL CANONIC CURRENT MODE DDCC BASED SRCO SYNTHESIZED USING A GENETIC ALGORITHM" AND "EVOLVING SINUSOIDAL OSCILLATORS USING GENETIC ALGORITHMS"	[112]–[114]
2005=	"TWO-DIMENSIONAL PHOTONIC CRYSTALS DESIGNED BY EVOLUTIONARY ALGORITHMS"	[115]
2005=	"LEARNING FROM LEARNING ALGORITHMS: APPLICATIONS TO ATTOSECOND DYNAMICS OF HIGH-HARMONIC GENERATION" AND "SHAPED-PULSE OPTIMIZATION OF COHERENT SOFT-X-RAYS"	[116], [117]
2004=	"AN EVOLVED ANTENNA FOR DEPLOYMENT ON NASA'S SPACE TECHNOLOGY 5 MISSION"	[118]
2004=	"AUTOMATIC QUANTUM COMPUTER PROGRAMMING: A GENETIC PROGRAMMING APPROACH"	[119]

[1] http://www.human-competitive.org/awards, last accessed 04 Feb 2018.

looking at the papers, which support the entries, shows that GP still requires tailoring for the problem at hand. It might also be argued that some of the problems are not challenging, with respect to the domains that they address and the fact that they do not suggest that they have a more generic applicability.

Genetic Programming and Hyper-heuristics have not really made the transition from the research environment to being available to a home user.

There are GP frameworks available, but they still require the knowledge and experience of the researcher to utilize that framework and then tailor it for the problem under consideration. Unquestionably, GP has succeeded, and continues to do so and the scientific literature has a significant body of peer reviewed work on this topic. However, it has yet to get to the position where it can be used by a non–expert user, sitting at home, who wants to evolve software for a problem they have.

V. Hyper-Heuristics

A hyper-heuristic has the aim of raising the level of generality of search/optimization algorithms, recognizing that no one search algorithm exists that is superior across all search/optimization problems [120]. Instead of searching the solution space directly, the most relevant heuristic to apply at any decision point is identified, which is applied to the solution space. It is hoped that a hyper-heuristic search algorithm can be applied to a wide range of problems, simply by changing the heuristics and utilizing the same heuristic search algorithm. Following these so called "Heuristic Selection Algorithms", later research investigated whether the heuristics themselves could be evolved [121], [122] thus saving the need to implement heuristics when new problems are tackled.

The first mention of the term "hyper-heuristic" in the scientific literature was in [123] (the term was also used in [124], but in a different context), although even earlier work could also be regarded as being a hyper-heuristic (e.g. [125], [126]), although the term was not used. A survey of hyper-heuristics is available in [127].

A 2000 research proposal (the author of this paper was one of the authors) said: "We will try to demonstrate how quick- and cheap-to-implement knowledge-poor heuristics can be used within a hyper-heuristic framework to provide a methodology suited to fast and cheap development of industrial and commercial systems. This will lead to a prototype hyper-heuristic 'toolbox' for the user community."

The authors of the proposal recognized that to provide a methodology suited to fast and cheap development of industrial and commercial systems was a challenging goal, and it was recognized that it would not be completed in the lifetime of the research award but, nonetheless, it was a long term vision for hyper-heuristics.

A generic hyper-heuristic framework is shown in Fig. 1. The initial research in hyper-heuristics focused on methodologies where several low-level heuristics were provided (no. 4 in the figure) and a high-level selector (no. 1) chooses which of the low level heuristics to apply at any given decision point. This was the so called "Heuristics to Choose Heuristics." Note should be taken of the domain barrier (no. 3). The high-level selector has no knowledge of the domain. Rather, it only knows how many heuristics there are and receives non-domain feedback, such as change in evaluation function, computation time etc. This enables the high level selector to operate on different domains, by replacing the low level heuristics by those that are able to address the new problem at hand.

Having to develop and replace a set of low level heuristics led to the obvious research question; can we evolve the low level heuristics so that we do not have to implement them when we want to change domains? A further question is, should a solution from one of the low level heuristics be accepted as the incumbent solution, and what form should that acceptance criteria take (e.g. always accept, improving only, sometimes accept worse solutions etc.) and can this acceptance criteria be evolved?

These latter questions are of more interest to the focus of this paper, as the approaches tend to be more EC based, and these research directions have been investigated in recent papers (e.g. [128]–[130]).

Hyper-heuristics have been an active research area for at least 20 years, and arguably back to the 1960's, yet there is still no off-the-shelf hyper-heuristic product that enables the commercial sector to benefit from this technology, let alone home users being able to access this methodology in the same way that they can now access 3D printing and immersive reality.

VI. Large Scale Software Development

As noted in Section IV, GP has had many successes and hyper-heuristic research (Section V) has made significant progress in the last 20 years. Both technologies still have some way to go

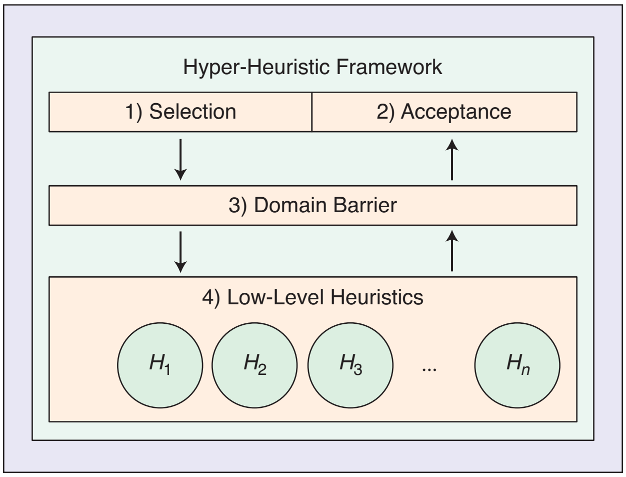


FIGURE 1 Hyper-Heuristic Framework.

TABLE 3 **Size of software developer community, from IDC Study.**

	ROLE	ESTIMATED # FOR 2014
ICT-SKILLED WORKERS	PROFESSIONAL SOFTWARE DEVELOPERS	11,005,000
	ICT OPERATIONS AND MANAGEMENT SKILLED WORKERS	18,008,900
	TOTAL	29,013,900
SOFTWARE DEVELOPERS	PROFESSIONAL SOFTWARE DEVELOPERS	11,005,000
	HOBBYIST SOFTWARE DEVELOPERS	7,534,500
	TOTAL	18,539,500

before being able to be offered to the business/home user in an easy to use form.

The scientific community recognizes that GP evolves functions, and saying that it evolves programs, could be viewed in a different way by the non-GP community, which means that their expectations are not met when they start using GP as a tool to integrate with their own systems.

Hyper-heuristic research has tended to focus on the main elements of the framework (see Fig. 1). There has been some work in trying to unify the various elements, but nothing is readily available at the moment that can be used off-the-shelf.

There are tools available, such as TSPLIB[2], MATLAB[3] and CPLEX[4] but these are either expensive, more suited to expert users and not necessarily EC related.

We know that large scale software development is difficult. Rosenberg [131] tells the story of Mitch Kapor who developed Lotus 1-2-3 and the popular personal information manager, Agenda. Kapor decided to develop a more up to date, extensible, fully functioning and featured personal information manager. What started as a grand vision became a tale of managing a large software development team with all the issues and problems that this brings. The resultant product, Chandler, is freely available but it never had the impact that was hoped for. The book [131] provides a stark reference to the difficulties of large scale software development, even by people who have developed highly successful products before.

Brooks [132], in his famous work—The Mythical Man-Month—noted that software development is difficult and when large software development projects do run into problems, adding additional manpower cannot save it. Indeed, it will make it even later.

There are many examples of software development projects failing. A small sample (there are numerous) are highlighted here:

1) "The U.S. Air Force has decided to scrap a major ERP (enterprise resource planning) software project after spending US$1 billion, concluding that finishing it would cost far too much more money for too little gain."[5]
2) "In 2003, Levi Strauss, was a global corporation, with operations in more than 110 countries but with an IT system that was an antiquated, 'Balkanised' mix of incompatible country-specific systems. So its bosses decided to migrate to a single SAP system and hired a team of fancy consultants (from Deloitte) to lead the effort. 'The risks seemed small,' wrote the researchers. 'The proposed budget was less than $5 m.' But very quickly things fell apart. One major customer, Walmart, required that the system interface with its supply chain management system, creating additional work. During the switchover to the new system, Levi Strauss was unable to fulfil orders and had to close its three US distribution centres for a week. In 2008, the company took a $192.5 m charge against earnings to compensate for the botched project and fired its chief information officer."[6]
3) "We examined 1,471 projects, comparing their budgets and estimated performance benefits with the actual costs and results. They ran the gamut from enterprise resource planning to management information and customer relationship management systems. Most, like the Levi Strauss project, incurred high expenses—the average cost was $167 million, the largest $33 billion—and many were expected to take several years. Our sample drew heavily on public agencies (92%) and U.S.-based projects (83%), but we found little difference between them and projects at the government agencies, private companies, and European organizations that made up the rest of our sample."[7]

There appears to be a need for more support for large scale software development projects. There are enough personnel working as software developers (see Table 3[8]) that any automation should be welcomed by the industry. Perhaps not by those whose jobs are at risk, but certainly by those who employ the developers. Of course, this is no different to many other industries, where jobs have been replaced by automation, but it does seem ironic that those responsible for automating so many jobs are now at risk themselves.

Even if we were able to get technologies such as GP and hyper-heuristics to the stage where they could be used by experienced software developers, it is not clear how these technologies could be packaged to make them readily available to business/home users, who are not experienced developers.

It is unrealistic, at least in the foreseeable future, to expect an evolutionary process to evolve a complete software product and perhaps this will never be an aim, or an expectation. Perhaps a more immediate aim would be to enable software developers to specify the requirements and interface as part of the software

[2] https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, last accessed 04 Feb 2018.

[3] https://www.mathworks.com/, last accessed 04 Feb 2018.

[4] https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer, last accessed 04 Feb 2018.

[5] https://www.cio.com/article/2390341/, last accessed 04 Feb 2018.

[6] https://www.theguardian.com/technology/2013/apr/21/fred-brooks-complex-software-projects, last accessed 04 Feb 2018.

[7] https://hbr.org/2011/09/why-your-it-project-may-be-riskier-than-you-think, last accessed 04 Feb 2018.

[8] https://www.infoq.com/news/2014/01/IDC-software-developers, last accessed 04 Feb 2018.

development life cycle and let an evolutionary process deliver required functionality, with some guarantees that it is fit for the purpose.

Indeed, this is similar to evolutionary art [133], which has a long and proven history [134]. In this paradigm the human is often part of the fitness evaluation and they judge the quality of the art that the evolutionary process produces. It could be envisaged that humans judge the quality of an evolved program by being part of the fitness evaluation. In this way, the human developer would not simply be a coder but would be tasked with guiding the evolutionary process via their feedback as to the effectiveness of each member of the population and would be helping to decide which programs are worthy of surviving to the next generation.

If developers come across some part of the system that is particularly difficult to develop they could call upon an EC based approach to evolve the required functionality.

This could be seen as being part of an agile approach to software development. That is, software developers start developing the software system by providing some functionality and gradually adding to it. If they come across some part of the system that is particularly difficult to develop they could call upon an EC based approach to evolve the required functionality, perhaps while they work on other parts of the system. Once the required functionality has been evolved, it is simply plugged into the system without the software developer having to do anything else. This functionality could even continue to evolve, should that be required, even when the system is deployed in a live environment.

It is likely that we would also have to draw on "Search-based Software Engineering" (i.e. the utilization of search methodologies such as GAs, simulated annealing and tabu search to address software engineering problems) [135]–[137]. If we are able to develop a user friendly framework that incorporates EC, search based software engineering; along with guarantees of what is delivered this would be a powerful product which would benefit the wider world, outside of the scientific community.

VII. Suggested Research Directions

Despite the large number of references in this paper, a more extensive survey of where EC has been used in the real world would certainly of benefit, if nothing else to serve as a baseline for future researchers. It would be useful to carry out a survey/analysis considering which methodologies from the scientific literature are utilized by the industrial community and to understand the reasons why some methodologies are adopted, whilst others are not. It would also be useful to survey the existing scientific literature to establish when authors say they are addressing a real world problem, is this really the case or are they modelling a simplified version of a problem, utilizing a benchmark dataset or addressing a problem that would not be recognized by the industrial community?

Most of the examples given utilized GAs. This is a little surprising as there are many other methodologies available [138], although GAs were one of the earliest and most popular EC methodologies. There might be some scope to look at how industry could benefit from other methodologies, as well as reporting non-GA examples that have been successfully deployed in industry. A book, or a series of articles, aimed at the commercial community might be useful so that there could be more take up.

The scientific community may benefit from a more complete survey where EC has been used in applications outside of the research arena. This might provide insights into the most useful methodologies, what domains are taking up the use of EC and the benefits that have arisen from using EC in a commercial environment.

Frameworks, that could be used out of the box, would be a valuable addition to the tools available to the commercial sector. It is recognized that some of these tools do exist but it is a steep learning curve, and sometimes expensive, for inexperienced users to start using them.

It would certainly be useful to investigate how various methodologies, such as EC, hyper-heuristics and search based software engineering could be integrated into a single framework.

If there was an integrated framework that enabled EC to be made easily available to the industrial/home user, it begs the question which EC methodology would be most suitable to use for a given problem provided to the framework? This is certainly worthy of further research. That is, provided with a problem should the framework use GA, GP; or one of the many other EC methodologies that are available, or even hybridizations of two, or more, of them?

VIII. Conclusion

The related work section of this paper has highlighted a number of projects where EC has been used, and is being used, in applications that have been deployed in the real world. It is noticeable that there are relatively few papers which report deployment of EC into a live industrial environment. It is also noticeable that many of these papers are from R&D departments within the companies involved.

We are certainly a long way from where an interested home user can access EC in the same way that access to 3D printing and immersive reality have become possible in the past few years.

EC has made significant research progress in the past 60 years but an integrated framework is lacking where all of this functionality can be easily accessed. The development of a framework would be welcome but there is research activity that needs to take place to support this framework so that the underlying complexity remains largely hidden from the end user.

Acknowledgment

This paper is based on a plenary talk that the author gave at the 2016 Congress on Evolutionary Computation, 24–29 July

2016, Vancouver. The author would like to acknowledge the support of IEEE and the Computational Intelligence Society, as without the opportunity to present the plenary talk it would not have been possible to write this paper.

References

[1] N. R. Murch, "Game on for Pokemon Go: Placement of Pokemon characters may breach confidentiality," *Br. Med. J.*, vol. 354, 2016.

[2] T. Althoff, R. W. White, and E. Horvitz, "Influence of Pokemon Go on physical activity: Study and implications," *J. Med. Internet Res.*, vol. 18, no. 12, pp. 82–95, 2016.

[3] M. Tateno, N. Skokauskas, T. A. Kato, A. R. Teo, and A. P. S. Guerrero, "New game software (Pokemon Go) may help youth with severe social withdrawal, hikikomori," *Phychiatry Res.*, vol. 246, pp. 848–849, 2016.

[4] M. Serino, K. Cordrey, L. McLaughlin, and R. L. Milanaik, "Pokemon Go and augmented virtual reality games: A cautionary commentary for parents and pediatricians," *Curr. Opin. Pediatrics*, vol. 28, no. 5, pp. 673–677, 2016.

[5] J. W. Ayers, E. C. Leas, M. Dredze, J.-P. Allem, J. G. Grabowski, and L. Hill, "Pokemon GO: A new distraction for drivers and pedestrians," *JAMA Internal Med.*, vol. 176, no. 12, pp. 1865–1866, 2016.

[6] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.

[7] S. Bringsjord and A. Sen, "On creative self-driving cars: Hire the computational logicians, fast," *Appl. Artif. Intell.*, vol. 30, no. 8, pp. 758–786, 2016.

[8] N. J. Goodall, "Can you program ethics into a self-driving car?" *IEEE Spectr.*, vol. 53, no. 6, pp. 28–58, 2016.

[9] S. E. Shladover, "The truth about "self-driving" cars," *Sci. Amer.*, vol. 314, no. 6, pp. 53–57, June 2016.

[10] N. Ensmenger, "Is chess the drosophila of artificial intelligence? A social history of an algorithm," *Soc. Stud. Sci.*, vol. 42, no. 1, pp. 5–30, 2012.

[11] V. Oduguwa, A. Tiwari, and R. Roy, "Evolutionary computing in manufacturing industry: An overview of recent applications," *Appl. Soft Comput.*, vol. 5, no. 3, pp. 281–299, 2005.

[12] N. M. Mohmad Kahar and G. Kendall, "A great deluge algorithm for a real-world examination timetabling problem," *J. Oper. Res. Soc.*, vol. 66, no. 1, pp. 116–133, 2015.

[13] N. G. Beligiannis, C. Moschopoulos, and D. S. Likothanassis, "A genetic algorithm approach to school timetabling," *J. Oper. Res. Soc.*, vol. 60, no. 1, pp. 23–42, 2009.

[14] G. T. Dias, P. J. de Sousa, and F. J. Cunha, "Genetic algorithms for the bus driver scheduling problem: A case study," *J. Oper. Res. Soc.*, vol. 53, no. 3, pp. 324–335, 2002.

[15] K. Sastry, D. E. Goldberg, and G. Kendall, "Genetic algorithms" in *Introductory Tutorials in Optimization and Decision Support Techniques*. New York: Springer, 2014, pp. 93–117.

[16] I. Christou, A. Zakarian, J.-M. Liu, and H. Carter, "A two-phase genetic algorithm for large-scale bidline-generation problems at Delta Air Lines," *Interfaces*, vol. 29, no. 5, pp. 51–65, 1999.

[17] J. van den Bergh, J. Beliën, P. de Bruecker, E. Demeulemeester, and L. de Boeck, "Personnel scheduling: A literature review," *Eur. J. Oper. Res.*, vol. 226, no. 3, pp. 367–385, 2013.

[18] A. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering: A review of applications, methods and models," *Eur. J. Oper. Res.*, vol. 153, no. 1, pp. 2–27, 2004.

[19] R. Sundararajan, T. Bhaskar, A. Sarkar, S. Dasaratha, D. Bal, J. K. Marasanapalle, B. Zmudzka, and K. Bak, "Marketing optimization in retail banking," *Interfaces*, vol. 41, no. 5, pp. 485–505, 2011.

[20] E. Rash and K. Kempf, "Product line design and scheduling at Intel," *Interfaces*, vol. 42, no. 5, pp. 425–436, 2012.

[21] R. Gopalan, S. O. Kimbrough, F. H. Murphy, and N. Quintus, "The Philadelphia districting contest: Designing territories for city council based upon the 2010 census," *Interfaces*, vol. 43, no. 5, pp. 477–489, 2013.

[22] A. Johnson, Y. Zhao, and X. Xu, "Transportation planning and scheduling for the 2014 special olympics USA games," *Interfaces*, vol. 46, no. 3, pp. 218–230, 2016.

[23] A. S. Manikas, J. R. Kroes, and T. F. Gattiker, "Metro meals on wheels treasure valley employs a low-cost routing tool to improve deliveries," *Interfaces*, vol. 46, no. 2, pp. 154–167, 2016.

[24] V. Ogris, T. Kristan, A. Škraba, M. Urh, and D. Kofjač, "iUrnik: Timetabling for primary educational institutions in Slovenia," *Interfaces*, vol. 46, no. 3, pp. 231–244, 2016.

[25] N. Pillay, "A survey of school timetabling research," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 261–293, 2014.

[26] E. Aarts, J. Korst, and W. Michiels, "Simulated annealing" in *Introductory Tutorials in Optimization and Decision Support Techniques*. New York: Springer, 2014, pp. 265–285.

[27] M. Gendreau and J.-Y. Potvin, "Tabu search" in *Introductory Tutorials in Optimization and Decision Support Techniques*, Boston, MA: Springer, 2014, pp. 243–263.

[28] E. Burke and G. Kendall, Eds., *Introductory Tutorials in Optimization and Decision Support Techniques*. New York: Springer, 2014.

[29] R. Bai, E. K. Burke, G. Kendall, J. Li, and B. McCollum, "A hybrid evolutionary approach to the nurse rostering problem," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 580–590, 2010.

[30] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, 2008.

[31] K. Eagle, "Using simulated annealing to schedule oil field drilling rigs," *Interfaces*, vol. 26, no. 6, pp. 35–43, 1996.

[32] M. A. Trick, H. Yildiz, and T. Yunes, "Scheduling major league baseball umpires and the traveling umpire problem," *Interfaces*, vol. 42, no. 3, pp. 232–244, 2012.

[33] A. Farmer, J. S. Smith, and L. T. Miller, "Scheduling umpire crews for professional tennis tournaments," *Interfaces*, vol. 37, no. 2, pp. 187–196, 2007.

[34] J. R. Willis and J. B. Terrill, "Scheduling the Australian state cricket season using simulated annealing," *J. Oper. Res. Soc.*, vol. 45, no. 3, pp. 276–280, 1994.

[35] M. Wright, "Timetabling county cricket fixtures using a form of Tabu search," *J. Oper. Res. Soc.*, vol. 45, no. 7, pp. 758–770, 1994.

[36] M. J. Fry and J. W. Ohlmann, "Route design for delivery of voting machines in Hamilton County, Ohio," *Interfaces*, vol. 39, no. 5, pp. 443–459, 2009.

[37] M. Brazil, P. Grossman, J. H. Rubinstein, and D. Thomas, "Improving underground mine access layouts using software tools," *Interfaces*, vol. 44, no. 2, pp. 195–203, 2014.

[38] K. W. Campbell, R. B. Durfee, and G. S. Hines, "Fedex generates bid lines using simulated annealing," *Interfaces*, vol. 27, no. 2, pp. 1–16, 1997.

[39] L. J. Fogel, "The human computer in flight control," *IRE Trans. Electron. Comput.*, vol. EC-6, no. 3, pp. 197–202, 1957.

[40] A. S. Fraser, "Simulation of genetic systems by automatic digital computers. I. Introduction," *Australian J. Biol. Sci.*, vol. 10, pp. 484–491, 1957.

[41] G. E. P. Box, "Evolutionary operation: A method for increasing industrial productivity," *J. Roy. Statist. Soc. Ser. C*, vol. 6, no. 2, pp. 81–101, 1957.

[42] M. Dorigo, M. Vittorio, and C. Alberto, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, no. 1, pp. 29–41, 1996.

[43] M. Dorigo, G. Di Caro, and L. Gambardella, "Ant algorithms for discrete optimization," *Artif. Life*, vol. 5, no. 2, pp. 137–172, 1999.

[44] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature*, vol. 406, pp. 39–42, 2000.

[45] T. Bäck and H. Schwefel, "Evolutionary computation: An overview," in *Proc. Int. Conf. Evolutionary Computation*, 1996, pp. 20–29.

[46] T. Bäck, "An overview of parameter control methods by self-adaptation in evolutionary algorithms," *Fundam. Inform.*, vol. 35, pp. 51–66, 1998.

[47] T. Bäck, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. London, U.K.: Oxford Univ. Press, 1998.

[48] L. Fogel and G. Burgin, "Competitive goal-seeking through evolutionary programming," Final Report, Contract AF 19(628), Air Force Cambridge Research Laboratories, Tech. Rep., 1969.

[49] G. Burgin and L. Fogel, "Air-to-air combat tactics synthesis and analysis program based on an adaptive maneuvering logic," *J. Cybern.*, vol. 2, no. 4, pp. 60–68, 1972.

[50] X. Yao, "A review of evolutionary artificial neural networks," *Int. J. Intell. Syst.*, vol. 8, no. 4, pp. 539–567, 1993.

[51] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 694–713, 1997.

[52] X. Yao, "Evolving artificial neural networks," *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[53] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, 1999.

[54] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms*, 1985, pp. 93–100.

[55] J. Schaffer, R. Caruana, L. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 51–60.

[56] L. Eshelman, R. A. Caruana, and J. Schaffer, "Biases in the crossover landscape," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 10–19.

[57] J. D. Schaffer and L. J. Eshelman, "On crossover as an evolutionarily viable strategy," in *Proc. Int. Conf. Genetic Algorithms*, 1991, pp. 61–68.

[58] J. D. Schaffer, D. Whitley, and L. J. Eshelman, "Combinations of genetic algorithms and neural networks: A survey of the state of the art," in *Proc. Int. Workshop Combinations Genetic Algorithms Neural Networks*, 1992, pp. 1–37.

[59] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995.

[60] R. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.

[61] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, no. 2–3, pp. 95–99, 1988.

[62] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artif. Intell.*, vol. 40, no. 1–3, pp. 235–282, 1989.

[63] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[64] D. B. Fogel, "An evolutionary approach to the traveling salesman problem," *Biol. Cybern.*, vol. 60, no. 2, pp. 139–144, 1988.

[65] D. B. Fogel, L. J. Fogel, and V. W. Porto, "Evolving neural networks," *Biol. Cybern.*, vol. 63, no. 6, pp. 487–493, 1990.

[66] D. B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 3–14, 1994.

[67] D. B. Fogel, Ed., *Evolutionary Computation: A Fossil Record*. New York: Wiley, 1998.

[68] K. Chellapilla and D. B. Fogel, "Evolution, neural networks, games, and intelligence," *Proc. IEEE*, vol. 87, no. 9, pp. 1471–1496, 1999.

[69] K. A. de Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, MI, 1975.

[70] K. A. de Jong, "Genetic algorithms: A 10 year perspective," in *Proc. Int. Conf. Genetic Algorithms and Applications*, 1985.

[71] K. A. de Jong, "Learning with genetic algorithms," *Mach. Learn.*, vol. 3, no. 3, pp. 121–138, 1988.

[72] K. A. de Jong, *Evolutionary Computation: A Unified Approach*. Cambridge, MA: MIT Press, 2016.

[73] R. M. Friedberg, "A learning machine: Part I," *IBM J. Res. Develop.*, vol. 2, no. 1, pp. 2–13, 1958.

[74] R. M. Friedberg, B. Dunham, and J. H. North, "A learning machine: Part II," *IBM J. Res. Develop.*, vol. 3, no. 3, pp. 282–287, 1959.

[75] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1975.

[76] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM J. Comput.*, vol. 2, no. 2, pp. 88–105, 1973.

[77] I. Rechenberg, "Cybernetic solution path of an experimental problem," Royal Air Force Establishment, Farnborough, Hampshire, England 1122, 1965.

[78] H. P. Schwefel, *Understanding evolution as a Collective Strategy for Groping in the Dark*. Berlin, Germany: Springer, 1991, pp. 388–397.

[79] H. P. Schwefel, "Evolutionary computation: History, status, and perspectives," in *Proc. Int. Conf. Artificial Neural Networks*, 1996, pp. 15–15.

[80] H. Beyer and H. Schwefel, "Evolution strategies: A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.

[81] M. Conrad and H. H. Pattee, "Evolution experiments with an artificial ecosystem," *J. Theor. Biol.*, vol. 28, pp. 393–409, 1970.

[82] D. Fogel, R. W. Anderson, R. G. Reynolds, and W. Rizki, "Memorial tribute to Dr. Michael Conrad," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 1–2, 2001.

[83] R. R. Kampfner and M. Conrad, "Computational modeling of evolutionary learning processes in the brain," *Bull. Math. Biol.*, vol. 45, no. 6, pp. 931–968, 1983.

[84] G. E. P. Box and N. Draper, *Evolutionary Operation A Method for Increasing Industrial Productivity*. New York: Wiley, 1969.

[85] A. S. Fraser, "Monte Carlo analyses of genetic models," *Nature*, vol. 181, pp. 208–209, 1958.

[86] A. S. Fraser, "Simulation of genetic systems," *J. Theor. Biol.*, vol. 2, pp. 329–246, 1962.

[87] D. Fogel, "In memoriam Alex S. Fraser," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 429–430, 2002.

[88] L. J. Fogel, "A new concept: The kinalog system," *J. Hum. Factors Soc.*, vol. 1, no. 2, pp. 30–37, 1959.

[89] L. J. Fogel, "Autonomous automata," *Ind. Res. Mag.*, vol. 4, no. 2, pp. 14–19, 1962.

[90] V. Piuri, "In memoriam: Dr. Lawrence J. Fogel," *IEEE Comput. Intell. Mag.*, vol. 3, no. 1, pp. 68–69, 2008.

[91] J. Kietzmann, L. Pitt, and P. Berthon, "Disruptions, decisions, and destinations: Enter the age of 3-D printing and additive manufacturing," *Bus. Horiz.*, vol. 58, pp. 209–215, 2015.

[92] B. Berman, "3-D printing: The new industrial revolution," *Bus. Horiz.*, vol. 55, pp. 155–162, 2012.

[93] J. R. Koza, "Hierarchical genetic algorithms operating on populations of computer programs," in *Proc. 11th Int. Joint Conf. Artificial Intelligence*, 1989, vol. 1, pp. 768–774.

[94] J. R. Koza, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[95] J. R. Koza and J. P. Rice, "Automatic programming of robots using genetic programming," in *Proc. 10th Nat. Conf. Association Advancement Artificial Intelligence*, 1992, pp. 768–774.

[96] R. Poli, W. B. Langdon, and N. F. McPhee. (2008). A field guide to genetic programming. [Online]. Available: http://lulu.com; http://www.gp-field-guide.org.uk

[97] E. T. Barr, M. Harman, Y. Jia, A. Marginean, and J. Petke, "Automated software transplantation," in *Proc. ACM Int. Symp. Software Testing and Analysis,* 2015, pp. 257–269.

[98] R. Harper, R. J. Chapman, C. Ferrie, C. Granade, R. Kueng, D. Naoumenko, S. T. Flammia, and A. Peruzzo, "Explaining quantum correlations through evolution of causal models," *Phys. Rev. A*, vol. 95, p. 042120, 2017.

[99] Z. Vasicek and L. Sekanina, "Evolutionary approach to approximate digital circuits design," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 432–444, 2015.

[100] O. E. David, H. J. van den Herik, M. Koppel, and N. S. Netanyahu, "Genetic algorithms for evolving computer chess programs," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 779–789, 2014.

[101] A. Elyasaf, A. Hauptman, and M. Sipper, "Evolutionary design of FreeCell solvers," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 4, pp. 270–281, 2012.

[102] D. Izzo, L. F. Simões, M. Märtens, G. C. de Croon, A. Heritier, and C. H. Yam, "Search for a grand tour of the jupiter galilean moons," in *Proc. ACM 15th Annu. Conf. Genetic Evolutionary Computation*, 2013, pp. 1301–1308.

[103] C. Browne, "Yavalath" in *Evolutionary Game Design*. London, U.K.: Springer, 2011, pp. 75–85.

[104] A. Elyasaf, A. Hauptman, and M. Sipper, "GA-FreeCell: Evolving solvers for the game of FreeCell," in *Proc. ACM 13th Annu. Conf. Genetic Evolutionary Computation*, 2011, pp. 1931–1938.

[105] P. Widera, J. M. Garibaldi, and N. Krasnogor, "Evolutionary design of the energy function for protein structure prediction," in *Proc. IEEE Congr. Evolutionary Computation*, 2009, pp. 1305–1312.

[106] P. Widera, J. M. Garibaldi, and N. Krasnogor, "GP challenge: Evolving energy function for protein structure prediction," *Genetic Program. Evolvable Mach.*, vol. 11, no. 1, pp. 61–88, 2010.

[107] P. Widera, "Automated design of energy functions for protein structure prediction by means of genetic programming and improved structure similarity assessment," Ph.D. dissertation, Sch. Comput. Sci., Univ. Nottingham, Nottingham, England, 2010.

[108] W. Weimer, T. Nguyen, C. Le Goues, and S. Forrest, "Automatically finding patches using genetic programming," in *Proc. 31st Int. Conf. Software Engineering*, 2009, pp. 364–374.

[109] S. Forrest, T. Nguyen, W. Weimer, and C. L Goues, "A genetic programming approach to automated software repair," in *Proc. ACM 11th Annu. Conf. Genetic Evolutionary Computation*, 2009, pp. 947–954.

[110] L. Spector, D. M. Clark, I. Lindsay, B. Barr, and J. Klein, "Genetic programming for finite algebras," in *Proc. ACM 10th Genetic Evolutionary Computation Conf.*, 2008, pp. 1291–1298.

[111] S. Manos, M. C. J. Large, and L. Poladian, "Evolutionary design of single-mode microstructured polymer optical fibres using an artificial embryogeny representation," in *Proc. ACM 9th Genetic Evolutionary Computation Conf.*, 2007, pp. 2549–2556.

[112] S. Kilinç, V. Jain, V. Aggarwal, and U. Cam, "Catalogue of variable frequency and single-resistance-controlled oscillators employing a single differential difference complementary current conveyor," *Frequenz*, vol. 60, no. 7–8, pp. 142–146, 2006.

[113] V. Aggarwal, "Novel canonic current mode DDCC based SRCO synthesized using a genetic algorithm," *Analog Integr. Circuits Signal Process.*, vol. 40, no. 1, pp. 83–85, 2004.

[114] V. Aggarwal, "Evolving sinusoidal oscillators using genetic algorithms," in *Proc. NASA/DoD Conf. Evolvable Hardware*, 2003, pp. 67–76.

[115] S. Preble, M. Lipson, and H. Lipson, "Two-dimensional photonic crystals designed by evolutionary algorithms," *Appl. Phys. Lett.*, vol. 86, no. 6, p. 061111, 2005.

[116] R. A. Bartels, M. M. Murnane, H. C. Kapteyn, I. Christov, and H. Rabitz, "Learning from learning algorithms: Application to attosecond dynamics of high-harmonic generation," *Phys. Rev. A*, vol. 70, p. 043404, 2004.

[117] R. Bartels, S. Backus, E. Zeek, L. Misoguti, G. Vdovin, I. P Christov, M M. Murnane, and H. C. Kapteyn, "Shaped-pulse optimization of coherent emission of high-harmonic soft X-rays," *Nature*, vol. 406, pp. 164–166,

[118] J. D. Lohn, G. S. Hornby, and D. S. Linden, *An Evolved Antenna for Deployment on NASA's Space Technology 5 Mission*. Boston, MA: Springer, 2005, pp. 301–315.

[119] L. Spector, *Automatic Quantum Computer Programming: A Genetic Programming Approach*. New York: Springer, 2007.

[120] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[121] E. K. Burke, M. Hyde, G. Kendall, and J. Woodward, "Automating the packing heuristic design process with genetic programming," *Evol. Comput.*, vol. 20, no. 1, pp. 63–89, 2012.

[122] E. K. Burke, M. Hyde, and G. Kendall, "Grammatical evolution of local search heuristics," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 406–417, 2012.

[123] P. Cowling, G. Kendall, and E. Soubeiga, *A Hyperheuristic Approach to Scheduling a Sales Summit*. Berlin, Germany: Springer, 2001, pp. 176–190.

[124] J. Denzinger, M. Fuchs, and M. Fuchs, "High performance ATP systems by combining several AI methods," Univ. of Kaiserslautern, Tech. Rep. SEKI-Report SR-96-09, 1996.

[125] H. Fisher and G. Thompson, "Probabilistic learning combinations of local job-shop scheduling rules," in *Industrial Scheduling*, J. Muth and G. Thompson, Eds. Englewood Cliffs, NJ: Prentice Hall, 1963, pp. 225–251.

[126] W. B. Crowston, F. Glover, G. Thompson, and J. D. Trawick, "Probabilistic and parametric learning combinations of local job shop scheduling rules," ONR Research Memorandum, GSIA, Carnegie Mellon Univ. Pittsburgh 1(117), Tech. Rep., 1963.

[127] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *J. Oper. Res. Soc.*, vol. 64, no. 12, pp. 1695–1724, 2013.

[128] S. Asta, E. Özcan, and A. J. Parkes, "Champ: Creating heuristics via many parameters for online bin packing," *Expert Syst. Applicat.*, vol. 63, pp. 208–221, 2016.

[129] N. R. Sabar, M. A. G. Kendall, and R. Qu, "A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 217–228, 2015.

[130] N. R. Sabar, M. A. G. Kendall, and R. Qu, "Automatic design of hyper-heuristic framework with gene expression programming for combinatorial optimization problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 309–325, 2015.

[131] S. Rosenberg, *Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software*. Three Rivers Press, 2008.

[132] F. P. Brooks, Jr., *The Mythical Man-Month*. Reading, MA: Addison-Wesley, 1975.

[133] C. G. Johnson, "Fitness in evolutionary art and music: A taxonomy and future prospects," *Int. J. Arts Technol.*, vol. 9, no. 1, pp. 4–25, 2016.

[134] S. Todd and W. Latham, Eds., *Evolutionary Art and Computers*. New York: Academic, 1992.

[135] M. Harman and B. F. Jones, "Search-based software engineering," *Inf. Softw. Technol.*, vol. 43, no. 14, pp. 833–839, 2001.

[136] K. Z. Zamli, B. Y. Alkazemi, and G. Kendall, "A tabu search hyper-heuristic strategy for t-way test suite generation," *Appl. Soft Comput.*, vol. 44, pp. 57–74, 2016.

[137] F. Sarro, F. Ferrucci, M. Harman, A. Manna, and J. Ren, "Adaptive multi-objective evolutionary algorithms for overtime planning in software projects," submitted for publication.

[138] K. Sörensen, "Metaheuristics: The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, 2015.

Jessica Moysen, Mario García-Lozano, and Silvia Ruíz
Department of Signal and Theory Communications, Universitat Politècnica de Catalunya-UPC, Barcelona, SPAIN

Lorenza Giupponi
Communications and Network Division, Centre Tecnològic de Telecomunicacions de Catalunya-CTTC, Barcelona, SPAIN

Conflict Resolution in Mobile Networks: A Self-Coordination Framework Based on Non-Dominated Solutions and Machine Learning for Data Analytics

FOOTAGE FIRM, INC.

Abstract

Self-organizing network (SON) is a well-known term used to describe an autonomous cellular network. SON functionalities aim at improving network operational tasks through the capability to configure, optimize and heal itself. However, as the deployment of independent SON functions increases, the number of dependencies between them also grows. This work proposes a tool for efficient conflict resolution based on network performance predictions. Unlike other state-of-the-art solutions, the proposed self-coordination framework guarantees the right selection of network operation even if conflicting SON functions are running in parallel. This self-coordination is based on the history of network measurements, which helps to optimize conflicting objectives with low computational complexity. To do this, machine learning (ML) is used to build a predictive model, and then we solve the SON conflict by optimizing more than one objective function simultaneously. Without loss of generality, we present an analysis of how the proposed scheme provides a solution to deal with the potential conflicts between two of the most important SON functions in the context of mobility, namely mobility load balancing (MLB) and mobility robustness optimization (MRO), which require the updating of the same set of handover parameters. The proposed scheme allows fast performance evaluations when the optimization is running. This is done by shifting the complexity to the creation of a prediction model that uses historical data and that allows to anticipate the network performance. The simulation results demonstrate the ability of the proposed scheme to find a compromise among conflicting actions, and show it is possible to improve the overall system throughput.

I. Introduction

A cellular network with intelligence and autonomous capabilities is called a self-organizing network (SON). The SON was introduced by the 3rd Generation Partnership Project (3GPP) as a key component of Long Term Evolution (LTE) networks, starting from the first release of this technology (Release 8) and expanding in the subsequent ones. The 3GPP has defined the main areas of the SON in [1], which are classified into self-configuration, self-healing and self-optimization. In addition, the 3GPP defined the minimization of drive tests (MDT) functionality in Release 9. This feature enables operators to collect user equipment (UE) measurements, with the purpose of optimizing network management [2]. Finally, in order to handle the potential conflicts that may exist due to the parallel execution of multiple SON

Digital Object Identifier 10.1109/MCI.2018.2807038
Date of publication: 10 April 2018

Corresponding Author: Jessica Moysen
(Email: jessica.moysen@tsc.upc.edu)

1556-603X/18©2018IEEE

functions, the self-coordination concept was introduced in Release 10 [3].

In this paper, we discuss the self-coordination problem. In particular, we focus on the output parameter SON conflict, i.e., when two or more SON functions aim at adjusting the same output parameter with opposite values. So, a SON coordinator controlling the actions of the SON functions during operation is considered a necessity [4]. In order to evaluate the performance of the proposed scheme, we focus on the handover management issue. We consider a well-known SON conflict between mobility load balancing (MLB) and mobility robustness optimization (MRO). MLB and MRO are two of the most important self-optimization functions that deal with mobility management. Both mechanisms modify the behavior of handover, which is a procedure that allows connections to be transferred between base stations in a seamless manner. However, each one pursues a different objective:

❑ MLB aims at balancing traffic load among cells so that cells with excess traffic (congested cells) can transfer some of their users to less loaded neighboring cells and vice versa. This can be done by changing the parameters that govern UE cell selection, such as handover thresholds, hysteresis margins and times to trigger a handover event. The primary goal is to achieve a higher system capacity, and this is done by distributing UE traffic across the available radio resources in the system.

❑ On the other hand, MRO is designed to improve mobility robustness:
- Minimization of call drops due to radio link failures: Depending on how the handover parameters are adjusted, too-early handovers may happen for some users. This means that the communication fails due to high propagation losses with the new cell. Similarly, too-late handovers may imply that communication with the serving base station is lost before a new connection is established.
- Minimization of unnecessary handovers: Too-short time-of-stays in the new cell or ping-pong (quick handover back to the previous cell) should not happen. For example, a vehicular user moving in a city where macrocells coexist with a layer of small cells should utilize a larger timer to trigger handovers toward picocells. This should avoid handovers from macro- to picocells, where vehicular users with high speed would have very short time-of-stays. Note that this would cause data throughput degradation due to the high volume of signaling transmission required to repeatedly update the serving cell.

Based on the information above, it is clear that MLB and MRO are two independent functions with independent objectives. However, if both are applied in parallel and without coordination, the actions requested by each SON function may be different. They may cause opposite changes in handover parameters, and so, they may enter into a conflict. Under these circumstances, the system would enter into a cycle of constant re-configuration, thereby causing performance degradation due to excessive signaling that requires radio resources.

In this context, machine learning (ML) is proposed as a candidate tool that allows the network to learn from experience and solve conflicts in an effective manner. The main feature is that the network is able to run different SON functions in parallel and improve the system performance. In particular, thanks to the use of a variety of techniques from data mining, statistics and ML, it is possible to analyze historical data to make predictions about unknown future events. This is known as predictive analytics and for the current work, it turns the network management from reactive to predictive. In this context, big data analytics are currently receiving big attention due to their capability to provide insightful information from the analysis of high volumes of data that are readily available for operators.

Based on that, the motivation behind this paper is to provide a tool that allows mobile network operators to become proactive by anticipating behaviors and making decisions accordingly. We focus on building a tool for efficient self-coordination that is based on the network performance prediction that is made by doing a proper data analysis of UE measurements. The tool works in two steps as graphically depicted in Figure 1:

1) In the first step, the proposed scheme learns from past experience to obtain a network performance predictor for each SON function individually. In particular, ML is used for data analytics. Available measurements are globally utilized in a learning process that yields an estimator by means of regression models.
2) Then, performance predictions are the inputs of a multi-objective optimization process, which searches for a set of non-dominated solutions or Pareto front. That is to say, the solutions

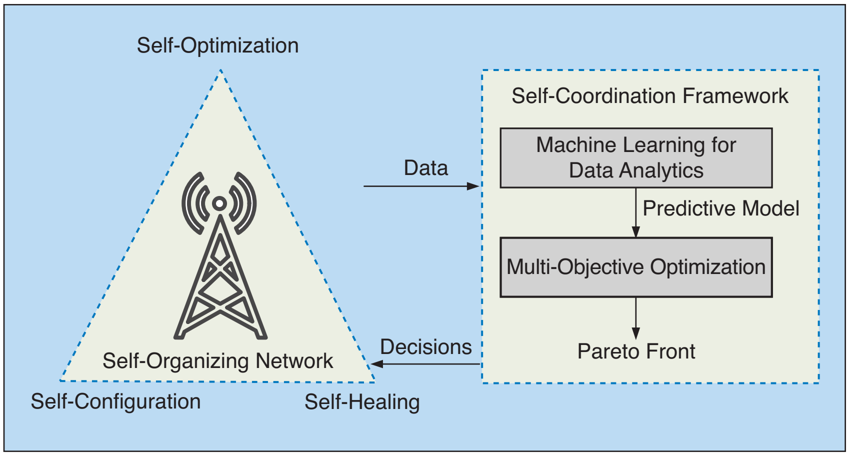


FIGURE 1 Self-coordination framework.

Self-organizing networks execute several parallel optimizations that aim at improving different aspects of mobile networks. Coordination among them is needed so that contradictory actions do not cancel each other.

cannot improve the performance of one SON function without degrading the other one. In this paper, we select the Non-dominated Sorting Genetic Algorithm II (NSGA-II), which is able to obtain a better spread of solutions with lower computational complexity than other algorithms [5].

In summary, the self-coordination framework aims at guaranteeing the operator's needs by finding the operating point that provides the best performance trade-off when an SON conflict happens. This is done through the multi-objective optimization process, which uses the insights from big data (i.e., it uses the patterns found in historical data) to predict the performance metric of each SON function, and then it searches for a representative subset of solutions that are non-dominated to each other.

This paper is organized as follows. Section II describes previous works and relates them to the current research. In Section III, we describe the self-coordination framework, its main design principles and the algorithms that we use to build it. In the same section, we discuss design details, the fine-tuning of the prediction model (Section III-A), and the multi-objective optimization process (Section III-B). In Section IV, we present the details of our particular case of study (MLB-MRO SON conflict) where the proposed scheme is applied. Section V describes the simulation platform and test scenarios. Subsequently, the simulation results and the corresponding discussion are presented. Finally, Section VI concludes this paper.

II. Related Work and Contributions

As we stated earlier, in order to guarantee a correct network operation, the SON functions have to be coordinated. Examples of solutions for the self-coordination problem can be found in [6], [7]. In [6], the authors focus on a preventive coordination mechanism that uses a policy-based decision process. The proposed scheme contains a policy engine to make automated operational decisions but under the control of the operator. That is to say, operators can decide which SON functions to execute by changing the rules that determine system behavior in a particular situation. However, in this solution, they address the coordination of a single SON function among the different cells. In [7], the authors propose a solution for the concurrent execution of multiple SON functions. They demonstrate that the proposed approach is able to coordinate several SON functions by taking advantage of game theory. To do that, each SON function is modeled as a Markov decision process (MDP) and solved by means of reinforcement learning (RL). However, a scalability vs. convergence trade-off arises and a detailed analysis is needed due to the required computational cost. Indeed, our approach tackles the conflict resolution problem for multiple SON functions addressing real-time computational requirements. In particular, it shifts processing complexity to the creation of a predictive model to be used when the network is in exploitation. This allows it to make fast and still accurate evaluations of the actions that are being evaluated by the SON coordinator and, hence, to make appropriate decisions in real time.

Since the implementation of MLB has a negative impact on the performance of MRO, the interaction between these two SON functions has received increased interest among researchers in this field. The work in [8] focuses on solving this conflict by adjusting the handover parameters taking into account the traffic load distribution. MLB modifies the handover parameters of the serving and the neighboring cells considering the MRO performance metrics and the load level of the neighboring cells. This way, it ensures a reduction in the number of too-early and too-late handovers and radio link failures while balancing the load among cells. This solution does not take advantage of learning from past experience, and decisions are always made based on instantaneous network information. Therefore, there is room to further minimize erroneous decisions. In this context, an example of an RL application for MLB-MRO SON conflict can be found in [9] and [10]. In both works, the authors take advantage of experience gained from past decisions in order to reduce uncertainty about the impact of the actions taken to resolve conflicts. The authors in [9] propose the use of Q-learning as an RL method. However, value-based methods such as Q-learning may compromise feasibility since they require a huge state (and action) space. Different authors provide a wide variety of strategies to reduce complexity such as function approximation and state space aggregation. The authors in [10] follow the latter while employing general policy iteration (GPI) [11]. The authors demonstrate that by using this method, the operator's needs can be met. However, the drawback of policy-based methods is that they suffer from high variance in the quality of results and they only consider a small subset of solutions so there is a certain probability that good solutions are not evaluated.

We propose to solve the self-coordination problem by means of multi-objective evolutionary algorithms in combination with ML. The use of multi-objective optimization alone would be unfeasible since the search for the solutions requires multiple performance evaluations to be done by means of system-level simulations. A real-time execution of this process would be absolutely unfeasible and this is one of the biggest problems in existing multi-objective approaches. Thanks to the use of ML, we can capitalize on the big data available in mobile networks and create a prediction model that allows real-time multi-objective optimizations. The proposed scheme exploits both components working together, namely, big data analytics and multi-objective optimization.

In this regard, we found several works in the literature regarding the benefits of big data in 5G networks, such as [12], [13], where the authors identify different sources of data that can act as an input to the SON entity, e.g., to perform load balancing and prediction operations, among others.

In this work, we take advantage of the MDT functionality (enhanced in Release 11) by collecting measurements indicating throughput and connectivity issues to estimate network performance. The exploitation of these huge amounts of data is analyzed with regression analysis, where the primary goal is to predict UE performance. We consider the analysis of our previous works [14]–[16], where the exploitation of the huge amount of data is analyzed with regression models to make better decisions for management purposes. In particular, our work in [15] focuses on two families of regression models, linear and nonlinear. A comparison was performed among different models, selected on a basis of low complexity and high accuracy. Prediction results were analyzed for different kinds and amounts of UE measurements. Based on the outcomes of that research, in this new work, we focus on bagging in combination with the support vector machine (SVM), referred to hereafter as the Bagged-SVM method.

Since we aim to find a trade-off between different goals, once the predictive model has been built, we solve the SON conflict by means of multi-objective optimization, and in particular, multi-objective evolutionary algorithms that use non-dominated sorting [5]. These kinds of algorithms are based on meta-heuristics that simulate the process of natural evolution and operate efficiently when a large number of parameters need to be configured simultaneously. While the multi-objective problem formulation presented herein can be solved by means of RL [17], it is important to note that following this approach, the agent needs a good strategy to explore the environment within a reasonable amount of time. Therefore, in this paper we have supplied the agent with NSGA-II, which is able to maintain a spread of solutions with a lower computational complexity.

Given this, the contributions of this paper can be summarized as follows:

- We present a tool for an efficient SON conflict resolution based on network performance predictions. A prediction model is created by extracting relevant information from the mobile network. This allows the operator to anticipate system behavior to make effective decisions in real-time. Scalability is guaranteed since the processing complexity is shifted to the creation stage of the model, allowing fast performance evaluation when optimization is running.
- In order to evaluate the proposed scheme, we analyze, without loss of generality, the MLB-MRO SON conflict. The simulation results show that the proposed scheme can not only solve the SON conflict but also improve overall system throughput.

III. General Self-Coordination Framework

We consider a wireless network composed of a set $\mathcal{M} = \{1,\dots,M\}$ of $M = |\mathcal{M}|$ cells regularly deployed with inter-site distance D. On each cell $j = (1,\dots,N)$, we consider N SON functions running in parallel. We denote by $\mathbf{c}^{(j)} = (c_1^{(j)},\dots,c_N^{(j)})$ the configuration parameter vector of a cell j, with $c_i^{(j)}$ denoting the value for the parameter of the SON function i, e.g., the transmission power (TXP), the antenna tilt (TILT), the action to switch ON or OFF the cell, handover parameters such as the Cell Individual Offset (CIO), hysteresis margin (HYS) or time to trigger (TTT). The N SON functions are implemented on every cell, which must send a request to the SON coordinator and get a positive response in order to adjust some of its parameters. These requests and responses are represented in Figure 2

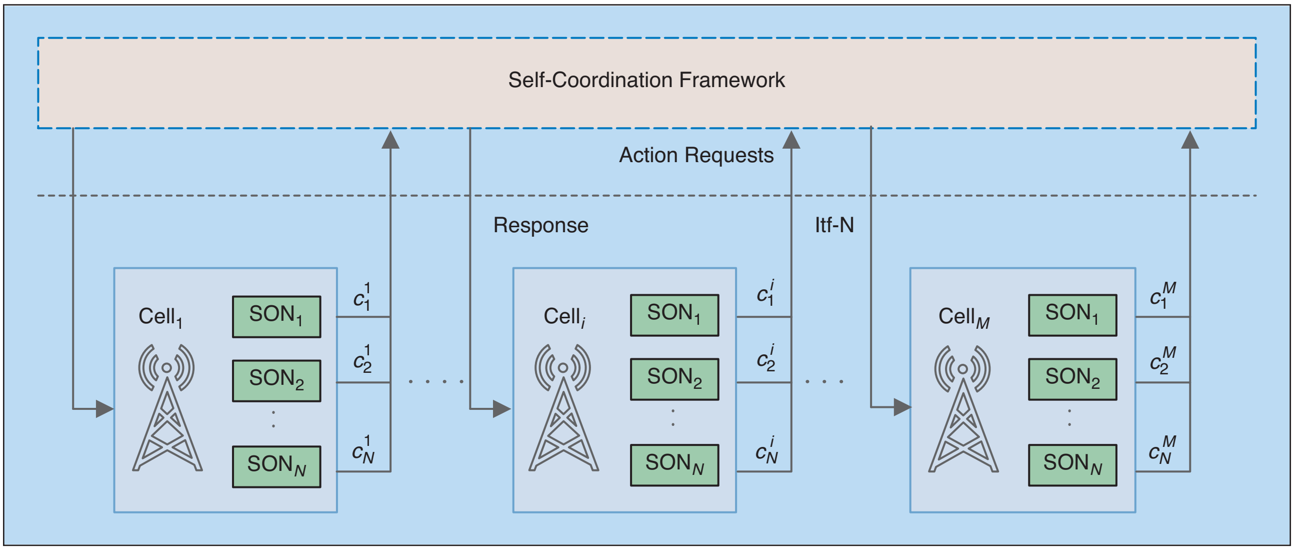


FIGURE 2 Each SON function requests the adjustment of different network parameters of cell *i*, and each one sends the action requests to the self-coordination framework.

Processing complexity is shifted to the creation of the prediction model. By using the patterns found in the historical data, we can make intelligent decisions faster. This allows to perform multi-objective optimization in real-time.

with the links between each cell and the SON coordinator and they must pass through the interface-N (Itf-N). We propose to solve the challenge of coordination among SON functions by analyzing their interactions based on the measurement history of the network. To do this, we design the self-coordination framework based on two main functions: 1) machine learning for data analytics, and 2) multi-objective optimization. Each is depicted in Figures 3 and 4, respectively, and described in the subsequent subsections.

A. Machine Learning for Data Analytics

The main objective of this function is to analyze large amounts of data sets to uncover hidden patterns, correlations and other useful information to make better decisions. This is done by considering the UE measurement reports, which contain power and quality measurements from the serving and neighbor base stations:

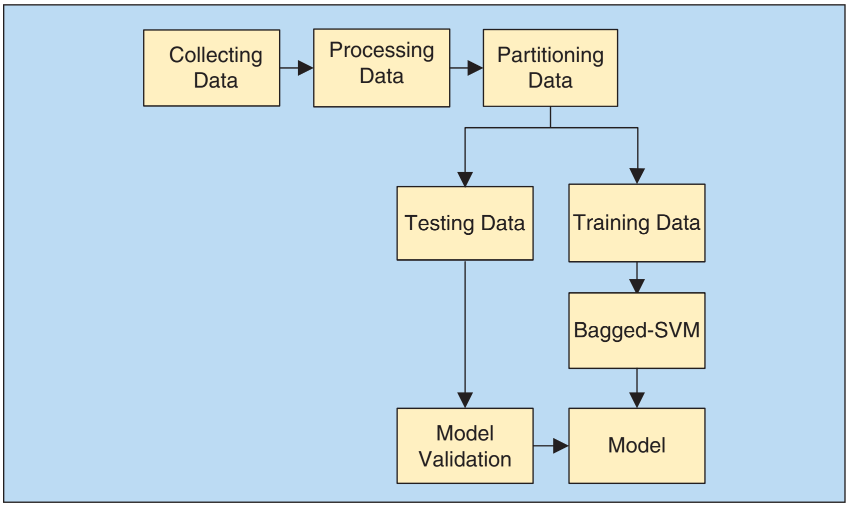


FIGURE 3 Machine learning for data analytics.

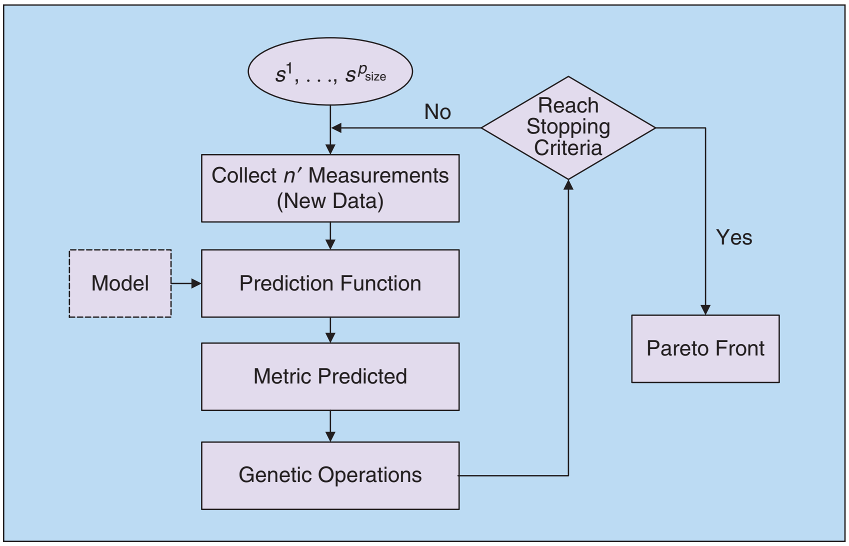


FIGURE 4 Multi-objective optimization.

❑ Reference Signal Received Power (RSRP): Average power received from LTE reference signals, used for channel estimation and handover/cell selection.

❑ Reference Signal Received Quality (RSRQ): Ratio between the total power received from reference signals and the total power received in the full bandwidth. The RSRQ measurement provides additional information about interference levels to make a reliable handover/cell reselection decision.

Once the data have been collected, we perform a data preparation process to extract and analyze the radio measurements. To do that, we propose to make use of ML techniques, which have been demonstrated to be very effective for making predictions based on observations. Among the most important ML tools for data analysis, we focus on regression analysis. It is important to note that evaluating the network performance for a given configuration requires a dynamic system-level simulation, which requires high computational time. If several configurations need to be assessed during the optimization, the computational cost is indeed prohibitive. So, regression analysis shifts the processing complexity to the model creation stage and allows fast performance evaluations when optimization is running.

Regression analysis is an ML technique, which allows us to predict the performance metric of each SON function. Regression takes an input vector ($\mathbf{x}$) and an output value (y) to develop a predictive model, returning the predicted output $\hat{y}$. We represent the input space by an n-dimensional input vector $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T \in \mathbf{R}^n$, where each dimension is an input variable. The size of the input space is $[\mathbf{x} \times n]$. The number of rows is the number of UEs at any place and anywhere, and the number of columns corresponds to the number of measurements n. In addition, a set involves m samples $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$. Each sample consists of an input vector $\mathbf{x}_i$, and a corresponding output y_i, which represents the UE performance indicator. Hence $x_i^{(j)}$ is the value of the input variable $x^{(j)}$ in sample i, and the error is usually computed via $|\hat{y}_i - y_i|$. To evaluate the accuracy of the model, after the data have been collected and normalized, we randomly select 3/4 of the data to be the training set (*x.train*, *y.train*), and place the rest into the testing set (*x.test*, *y.test*).

A well-known paradigm to improve the accuracy of regression models is ensemble learning, which combines multiple weak learners to solve the same problem. This approach usually produces more accurate solutions than a single model would. In this sense, one of the most useful techniques is bagging [18]. Bagging improves regression by subsampling the training samples and randomly generating training subsets. Then, it runs the learning algorithms several times, each one with a different subset, and a final regressor is obtained by averaging the different outputs. As we stated earlier, in this work, we focus on bagging in combination with SVM, referred to hereafter as the Bagged-SVM method.

SVM analysis is a popular ML tool for classification and regression, first identified by *Vladimir Vapnik* in [19]. The motivation to use SVM is that this method shows high accuracy in predictions and we can obtain good behaviors with nonlinear problems. For the current problem, given $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$, the goal is to find a function $f(\mathbf{x})$ that deviates from y_n by a value no greater than ϵ. The main idea is to characterize the optimal hyperplane, which will approximate all training samples with the required precision. For this purpose, the problem can be mapped onto a higher dimensional space. This allows the points to be relocated onto another space such that they become linearly separable and the fit is easier. In particular, data are transformed into a higher dimensional space by means of kernel functions. Since nonlinear kernels can be used, nonlinear regression is also possible. The estimation accuracy depends on the setting of the precision ϵ, the kernel parameters and the regularization parameter c [20]. Hence, we tune these parameters to generate predictions by creating a model on a subset of training samples. For further details on the theory behind SVM, the reader is referred to [21], [22].

The whole process is depicted in Figure 3 and summarized as follows:

1) Collecting data. For each UE, we collect RSRP and RSRQ from the serving and neighboring cells. To test the performance of each SON function, different metrics are obtained.
2) Processing data. Once data are collected, they are prepared by normalizing every variable.
3) Partitioning data. To validate the model to be created, observations are partitioned into two sets, one for calibration and the other for validation.
4) Building the ML models. We select the SVM regression model, where in order to enhance the performance of the learning algorithm, multiple data sets are used by means of the bagging technique. SVM is then applied to produce a regressor. For each test value, we predict and evaluate its performance against the actual value in terms of root mean square error (RMSE).

The model produced by the regressive algorithm is the input to the multi-objective optimization process described below.

B. Multi-Objective Optimization

As we stated earlier, we consider the situation in which optimizing a particular solution with respect to a single SON objective can result in unacceptable results with respect to the other SON objectives. That is, the parallel execution of N SON functions may generate a resource conflict if at least two of them request to adjust parameters in a way that cancels the actions the other one intends to take. Therefore, potential conflicts may occur, that will cause degradation of network performance. In this context, multi-objective optimization is a promising tool to find the point that provides the best trade-off performance of conflicting SON functionalities. In particular, we use multi-objective evolutionary algorithms to find a compromise between multiple SON functions. Evolutionary algorithms are population-based meta-heuristic algorithms that are capable of obtaining a set of solutions simultaneously, so they are very suitable to solve multi-objective optimization problems. The general approach determines an entire Pareto optimal solution set or Pareto front. This is a representative subset of solutions that are non-dominated with respect to each other. These kinds of solutions cannot improve any of the objectives without conflicting with at least one of the other objectives, i.e., optimizing a decision vector ($\mathbf{s}$) with respect to a single objective often degrades other objectives. As a consequence, a reasonable solution to a multi-objective problem is to find a set of solutions that satisfies the objectives at an acceptable level without being dominated by any other solution. That is, the final solution of the decision-maker is always a trade-off.

$$\begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} \rightarrow \begin{bmatrix} \mathrm{SON}_1 \\ \mathrm{SON}_2 \\ \vdots \\ \mathrm{SON}_N \end{bmatrix} \rightarrow \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_K \end{bmatrix}$$

Let us consider N SON functions that pursue the optimization of K objectives. A minimization problem with K objectives can be formulated as follows:

$$\begin{array}{ll} \text{minimize} & (F_1(\mathbf{s}), F_2(\mathbf{s}), \ldots, F_K(\mathbf{s})) \\ \text{subject to} & \mathbf{s} \in \mathbf{S} \end{array}$$

where $\mathbf{s} = s_1, \ldots, s_n$ is an n-dimensional decision variable vector in the solution space $\mathbf{S}$. Thus, the objective is to find an objective vector $\mathbf{s}^*$ (feasible solution) that minimizes a given set of K objective functions, i.e., we say that $\mathbf{s}^*$ is Pareto optimal if there is no alternative vector $\mathbf{s}$ where improvements can be made to at least one SON objective function without reducing another one.

Among the multi-objective formulations available in the literature, we use NSGA-II [5]. The reason for using this approach is that these kinds of methods take advantage of genetic algorithms

Ensemble learning produces more accurate predictions than a single regression model would. The bagged support vector machines method is used to create our predictor.

(GAs), which are heuristic algorithms that can adapt their objective functions to the multi-objective problem. GAs are able to improve partial solutions since they create new individuals by performing selection, crossover and mutation, but at the same time they keep the best individuals (i.e., the elitism). In this regard, NSGA-II allows less computational complexity than other evolutionary algorithms, and it also prevents the loss of good solutions once they are found since it preserves the elitism.

The concept of GA was introduced by Holland in [23]. GAs are inspired by the evolutionary theory explaining the natural selection. In GAs, a solution vector is called a chromosome, which is represented by a series of genes made of discrete units. Each unit controls one or more of the chromosome features. A chromosome corresponds to a unique solution $\mathbf{s}$ in the solution space $\mathbf{S}$. GAs manage a set of chromosomes or population, which is usually initialized by random valid solutions. As the search evolves, the population eventually converges to a single solution. To do that, the two most important operators are crossover and mutation. Crossover combines two parent chromosomes to form two new solutions called offspring. By iteratively applying the crossover operator, the best offsprings appear more frequently in the population, leading to convergence toward a good solution. On the other hand, mutation introduces genetic diversity in the population by means of random alterations in the offspring. The probability of mutation is very small and depends on the length of the chromosome. Therefore, the new chromosome produced by mutation will not be very different from the original one.

Given this, the procedure is adapted to our particular problem as follows: NSGA-II starts with a random initial population $\mathbf{s}^1, \ldots, \mathbf{s}^{p_{\text{size}}}$ of size p_{size}. Each feasible solution corresponds to a vector of size M, where each value denotes the network parameter to be optimized, which is associated with each cell. In order to evaluate the fitness of each chromosome, in each iteration t, we collect n' measurements at some arbitrary points in the scenario. These measurements are obtained as a consequence of having configured the parameters of the scenario according to each chromosome $\mathbf{s}$. These n' measurements (new data) and the built model already described in Section III-A, are the inputs to the prediction function, which gives us the UE performance. As a result, for each $\mathbf{s} \in \mathbf{S}$, we obtain a predicted performance metric. The algorithm applies crossover and mutation to create an offspring population. In generation t, an offspring population of size p_{size} is created from the parent population, and non-dominated fronts $F_1, F_2, \ldots, F_K$ are identified in the combined population. The next population is filled starting from solutions in F_1, then F_2, and so on. A high-level overview of the process is depicted in Figure 4.

In summary, we first exploit the huge amount of data already available in the network to predict future network performance. We build a prediction model based on historical UE measurements and we apply regression analysis techniques to predict network performance. Second, we use the built model as an input of the multi-objective evolutionary algorithm to solve potential conflicts by finding a set of solutions that satisfies the objectives without being dominated by any other solution. In this way, the gain in time is substantial since the built model allows fast performance evaluation when optimization is running.

IV. MLB and MRO Function Conflict

In this section, we analyze in detail the MLB-MRO SON conflict. First, we explain the handover triggering procedure. Second, we explain the reason for the conflict, and finally, the considered performance indicators.

A. Handover Triggering Procedure

Among the different handover triggering procedures in LTE, we focus on Event A3, which is the main criterion to manage intra-LTE mobility. Event A3 is defined as the situation in which the UE perceives that a neighbor cell's RSRP is better than the serving cell's RSRP by a certain margin [24]. In order to reduce the ping-pong effect, measurements used to assess the event are averaged, hysteresis margins are introduced and the conditions must be met during the so-called TTT. Hence, the event entering condition is defined as

$$\begin{aligned} RSRP_{nc} > RSRP_{sc} &+ \text{offset} \\ &+ \text{Hysteresis} - \text{CIO} \end{aligned}$$

where $RSRP_{nc}$ and $RSRP_{sc}$ are the averaged reference signal power strengths of the neighbor and the serving cell, respectively, while offset and hysteresis parameters cause the serving and neighbor cells to be more and less attractive, respectively [24]. The combination of both defines a net hysteresis margin that delays notification of the event to guarantee that the neighbor cell is now the dominant one. These values are used as a basis and can be modified to adapt the condition to a particular UE mobility status, for example, being reduced for a high-speed UE. Finally, the cell individual offset, or CIO, is a cell-specific parameter set by the serving cell for each of its neighboring cells. It is used for load management, since the higher its value, the more attractive the corresponding neighbor will be. The whole process can be appreciated in Figure 5a, which shows the trace of average RSRP measurements from the serving cell and a neighbor before and after a handover. On the other hand, Figure 5b illustrates the difference between the hysteresis margin and the TTT taken by each SON function.

B. SON Conflict

In order to explain the SON conflict, we implement MLB and MRO SON functions in a distributed manner. As previously introduced, both of them aim at adjusting the CIO, hysteresis and the TTT handover parameters for different purposes. The goal of MLB is to optimize the network quality of service (QoS) by evenly distributing the load among the different cells. On the other hand, the MRO aims at decreasing the number of radio link failures (RLFs) caused by too-early or too-late handovers.

Figure 6 shows that the two SON functions are implemented in a distributed manner and concurrently executed in a per-cell manner. In order to analyze the MLB-MRO SON conflict, we assume that the serving cell ($cell_A$) is overloaded and its neighbor ($cell_B$) has a lower load. Therefore, $cell_A$ chooses its neighbor $cell_B$ to balance the load, and the action request is to adjust the CIO, the hysteresis and TTT handover parameters. This way, the condition in Event A3 can be met by UEs close to the cell edge. Hence, several handovers will happen from $cell_A$ to $cell_B$, thus transferring part of the load to $cell_B$. This results in too-early handovers of UEs in $cell_A$ causing an increase in the RLFs rates and ping-pong effects. As a consequence, the MRO SON function detects these metrics and tries to reduce them by requesting new changes in CIO, the hysteresis and TTT parameters. Since $cell_A$ is still overloaded, the MLB changes the handover parameters again and the MRO will also change them in the opposite direction, thus yielding an endless loop.

The proposed scheme, BDA-NSGA-II, allows to obtain a set of network configurations that improves its performance when conflicting optimization functionalities are present.

C. Performance Indicators

Besides the UE measurement reports, which contain RSRP and RSRQ values coming from the serving and the neighboring cells, the data about the performance of each UE is also collected once it performs a handover. These metrics are chosen based on the objective of each SON function.

On the one hand, the main objective of MLB is to improve end-user experience and achieve higher system capacity by distributing user traffic across system radio resources. As a consequence, the load of a cell is measured in terms of the average physical resource blocks (PRBs) that can be allocated to the users and the average signal to interference plus noise ratio (SINR) of each cell. The number of bits at the physical layer, referred to as the transport block size (TBS), is chosen taking into account the data that need to be transmitted by a UE. The media access control (MAC) has to first decide on the modulation scheme that can be scheduled to the user and then check the physical resource grid for the availability of the resource blocks. Given this, the MAC can decide upon the modulation and coding scheme index and its TBS index taken from [25], and then decide the number of PRBs that can be allocated to the user (i.e., users are allocated a specific number of subcarriers for a predetermined amount of time).

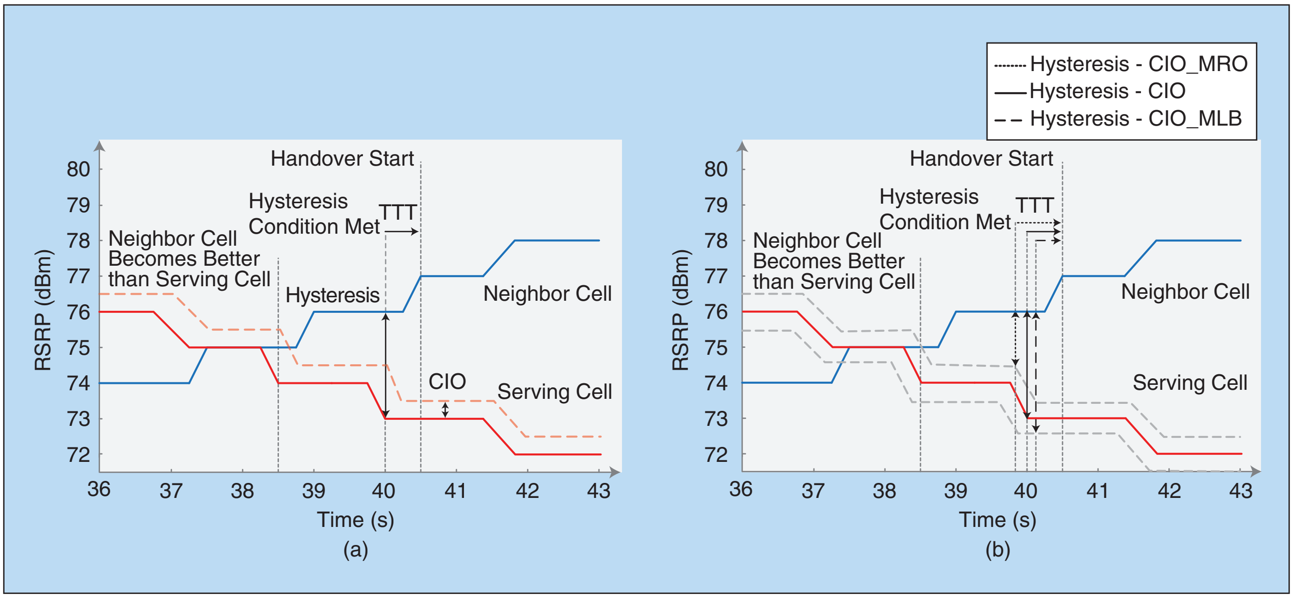


FIGURE 5 Effect of the handover parameters. The figure on the left side depicts the perceived RSRP of serving cell and a neighboring cell by a UE, which crosses the border of the cells. By default, the algorithm uses a hysteresis margin of 3.0 dB and a TTT of 256 ms. On the right, the MLB and MRO tune these values in order to achieve their own goals.

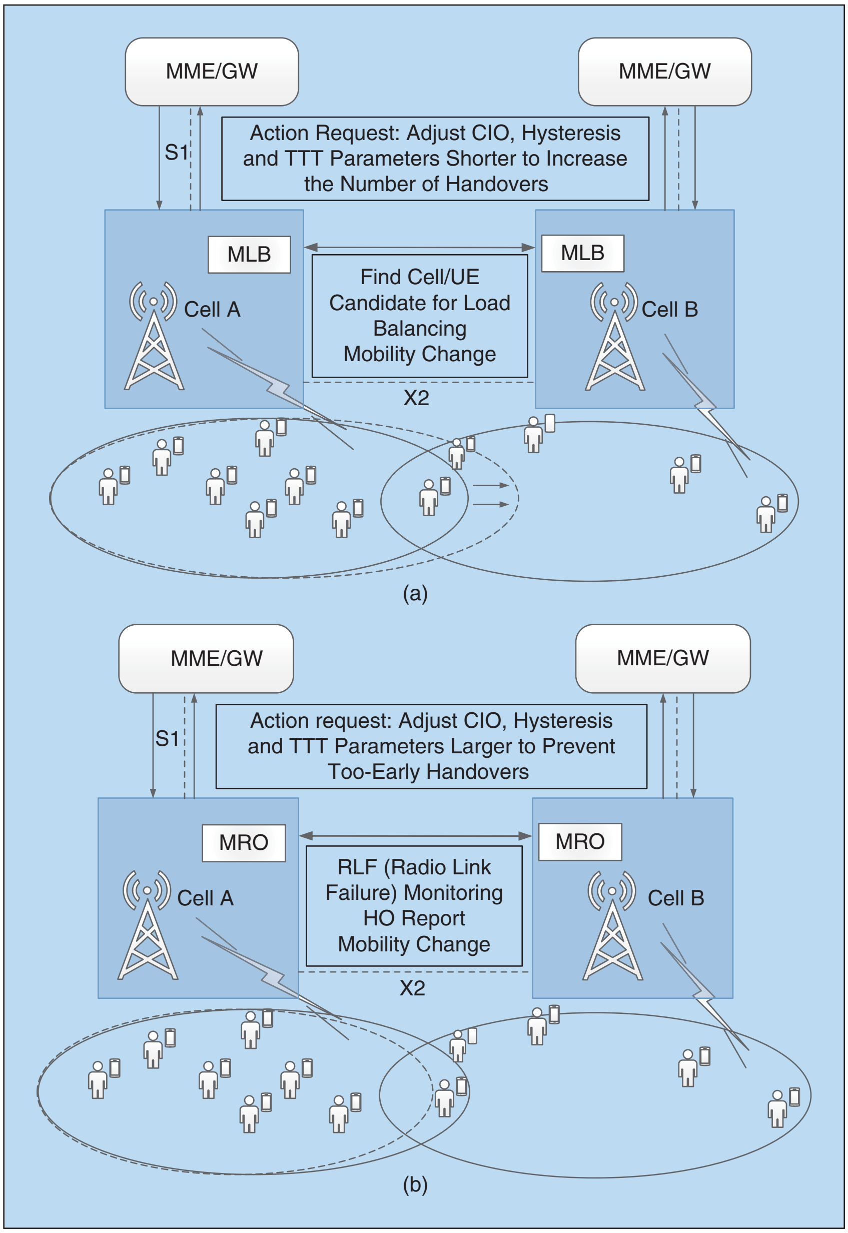

MME/GW
MME/GW
S1
Action Request: Adjust CIO, Hysteresis and TTT Parameters Shorter to Increase the Number of Handovers
MLB
MLB
Cell A
Cell B
Find Cell/UE Candidate for Load Balancing Mobility Change
X2
(a)
MME/GW
MME/GW
S1
Action request: Adjust CIO, Hysteresis and TTT Parameters Larger to Prevent Too-Early Handovers
MRO
MRO
Cell A
Cell B
RLF (Radio Link Failure) Monitoring HO Report Mobility Change
X2
(b)

FIGURE 6 In the figure on the left side, the MLB aims at increasing the number of handovers by adjusting the CIO_{AB} shorter to force UEs to handover to $cell_B$, whereas in the figure on the right side the MRO aims at decreasing the number of handovers by adjusting the CIO_{AB} larger to prevent too-early handovers.

On the other hand, the main objective of MRO is to guarantee proper mobility. So, in this case, RLF and the ping-pong effect are the performance metrics. RLFs may happen due to poor handover parameter settings, for example, a UE's being forced to hand off too early to a cell and, as a consequence, failing to establish the new connection before a timer expires. The ping-pong effect happens when a new handover is performed back to the source cell right after a successful handover and a new handover is performed before a determined minimum time of stay. In both cases, the MRO will be activated.

At the beginning of every Transmission Time Interval (TTI), the system will detect the load level of every cell. If the resource usage is higher than a threshold, the MLB will be activated in that particular cell. On the other hand, if the system detects that RLFs occur in the serving cell before, during or after a handover procedure, the MRO will be activated.

V. Performance Evaluation

As we stated earlier, in order to evaluate the performance of the proposed scheme, we consider the MLB-MRO SON conflict. We first present the simulation scenario and then the simulation results for this SON conflict.

A. Simulation Parameters

The proposed scheme is assessed by means of the 3GPP-compliant, full protocol stack, ns-3 LTE-Evolved Packet Core Network Simulator (LENA). The simulation setup is based on the assumptions for evaluating the SON conflict. We consider two sites of tri-sectored macro eNodeBs (i.e., three sectors per each site, resulting in six cells) regularly deployed with a 500 m inter-site distance and with UEs nonuniformly spread. A full-buffer traffic model, Transmission Control Protocol (TCP) and a Proportional Fair (PF) scheduler are considered. The PF works by scheduling a user when its instantaneous channel quality indicator is above its own average channel. The rest of the simulation parameters are described in Table 1.

The LTE module of ns-3 has been used to evaluate the capacity of the proposed scheme and to solve the SON conflict proposed in Section IV. By doing this, we model not only the LTE radio access network, but also the corresponding core network, known as the Evolved Packet Core (EPC). In this sense, the EPC mode is used to provide end-to-end connectivity to the users. The EPC model also enables a direct connection between two eNBs (X2 procedure) to handover a UE from a source eNB to a target eNB, and the user average throughput is obtained by enabling the Radio Link Control (RLC) simulation output. Notice that in the LTE-EPC ns-3 network simulator, generally, there is no RLF (handling of radio link failure is not yet modeled), i.e., once the UE goes into a connected mode, it does not change its state during the simulation. In order to bypass this issue, we tune the handover by

TABLE 1 Parameters.

LTE	VALUE
SCHEDULER	PROPORTIONAL FAIR
TRANSPORT PROTOCOL	TRANSMISSION CONTROL PROTOCOL (TCP)
DL/UL TRAFFIC MODEL	FULL-BUFFER
LAYER LINK PROTOCOL	RADIO LINK CONTROL (RLC)
MODE	UNACKNOWLEDGED MODE (UM)
CARRIER FREQUENCY	2 GHz
BANDWIDTH	5 MHz DL AND UL
NUMBER OF PRBs	25
TRANSMISSION TIME INTERVAL (TTI)	1 MS
SIMULATION TIME	200 S
MACRO CELL SCENARIO	
MACRO-CELL SITES	2
NUMBER OF CELLS	6
NUMBER OF UEs PER CELL	IN THE RANGE $[7-37]$
NUMBER OF UEs	108
MACRO-CELL Tx POWER	46 dBm
UE Tx POWER	10 dBm
DISTANCE BETWEEN ADJACENT MACRO-CELL SITES	500 M
OVERLOADED CELL	1

TABLE 2 Handover parameters.

PARAMETERS	VALUE
RSRP RANGE	0-97 dBM
CIO VALUES	0-30 dB
HYSTERESIS VALUES	0-15 dB
TTT	13 FIRST VALID VALUES (ALL IN MS) [24]
MOBILITY	RANDOM WALK MOBILITY

TABLE 3 Bagged-SVM parameters.

PARAMETERS	VALUE
SVM-TYPE	EPS-REGRESSION
SVM-KERNEL	RADIAL
COST c	100
GAMMA γ	2
EPSILON ϵ	0.1
NRMSE	0.06
SAMPLE METHOD	BAGGING
NUMBER OF ITERATIONS	1000

TABLE 4 GA parameters.

PARAMETERS	VALUE
NUMBER OF GENERATIONS	100
INITIAL POPULATION SIZE	50
SIZE OF $\mathbf{s}$	18
MUTATION PROBABILITY	0.25
CROSSOVER PROBABILITY	0.8
NUMBER OF ELITES	2

increasing the timers (e.g., the handover joining time). As a consequence, when a handover radio link failure occurs, handover timers will indicate such situation in a natural manner. In the case that a UE goes very far away from the eNB, schedulers just stop assigning resources to that UE, and once it comes back, allocations start again. As a result, we consider that a UE detects an RLF when its channel quality indicator is equal to zero, and the MAC traces of that UE will be zero, meaning, the UE is out of range during that period of time. The handover parameters are described in Table 2, and the ML and genetic algorithm parameters are described in Tables 3 and 4.

B. Simulation Results

We present the simulation results that allow a performance comparison between two different schemes:

1) The MLB and MRO are independently executed, and they tune the CIO, hysteresis and TTT handover parameters to achieve their own objectives.
2) Proposed scheme: The self-coordinator framework is activated to avoid the MLB-MRO SON conflict (BDA-NSGA-II).

In order to evaluate the performance of the MLB-MRO SON functions, each one is executed independently. We implement both of them by taking advantage of genetic algorithms. For this scheme, the metrics are the average SINR to evaluate the performance of the MLB SON function, and the RLF rate for the case of the MRO SON function. We define the RLF rate as the number of RLFs divided by the total number of handovers performed during a predefined time window. Figures 7a and 7b show the fitness value of the best individual found in each generation (Best) and the mean of the fitness values across the entire population (Mean). We observe that, the chromosome tends to get better as generations proceed. Figure 7a depicts the time evolution of the average SINR in the scenario during the evolution of the genetic algorithm. We observe that, as the generations proceed, the MLB SON function is able to find the configuration of the vector with handover parameters that maximize the value of SINR. Similar behaviors can be appreciated in Figure 7b, where the figure shows the evolution of the genetic algorithm in terms of the ratio of RLF.

Figures 8a and 8b show the performance of each cell in the network as a consequence of the actions taken by MLB and MRO SON functions. In Figure 8a, we observe that the actions taken by the MLB function yield higher values of SINR. This is because the number of overloaded cells has been reduced by choosing appropriate neighbors to balance the load. The action that is requested by the MLB function is tuning the values of the CIO, the hysteresis margin and the TTT to promote handovers. However, due to the new consecutive handovers, executed within a short period of time, the ratio of RLF increases (see Figure 8b). We observe opposite behaviors in case of the MRO SON function, which decreases the ratio of RLF in each cell, but also decreases the values of SINR in each cell.

In the following paragraphs, we analyze the performance results for the proposed scheme. As we stated in Section V, the metrics that have been considered are the average number of PRBs allocated to each user and the MLB and MRO objective functions, average SINR and RLF and ping-pong ratios, respectively.

As we already explained in Section III, once we extract the most relevant information from the mobile network, the next step is to obtain the network performance prediction model. To do that, we determine the SVM tuning parameters (see Table 4) prior to fitting the model, and then we fit the training data. The SVM regression model is run several times, each with a different subset of training samples. Finally, in order to evaluate the accuracy of the predictions, the performance of the learned function is measured on the test set, resulting in 94% prediction accuracy, which is expressed in terms of (1-NRMSE)×100, where NRMSE is the normalized RMSE. Then, the network performance prediction model is fed into the NSGA-II, which finds the best configuration of handover parameters for each cell. The first step when solving a multi-objective problem is to get a handle on the feasible region. Figures 9a and 9b capture the whole spectrum of the Pareto front considering the average number of PRBs and the average SINR respectively. Note that the Pareto front has four dimensions, one for each of the performance metrics to optimize: the radio link failure rate, the ping-pong rate, the average number of physical resource blocks allocated to UEs and the average SINR. Hence, in order to provide an insightful representation, we have provided two 3-D projections (Figures 9a and 9b). Each figure represents three of the objectives: the ratio of RLFs, the ping-pong rate and the average number of PRBs per UE in case of Figure 9a and the average SINR in Figure 9b. If there were a single point that minimizes these objectives, it would be the solution; but often, there are different optimal points for each objective. These are represented on the top of the figure and they determine the Pareto front. Any point on this front is a non-dominated or "Pareto optimal" solution. By moving along the curve, it is possible to prioritize the minimization of RLFs and ping-pong rate at the expense of the SINR, or vice versa, but we cannot improve all three of them at once, and the blue shaded region is the feasible region. However, any point in the feasible

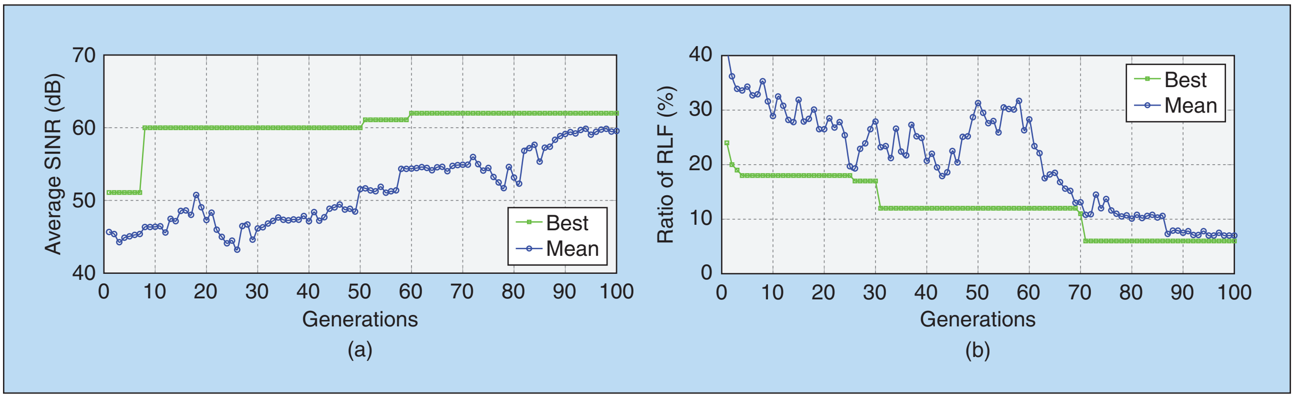


FIGURE 7 Evolution of the performance when MLB and MRO functions are independently executed.

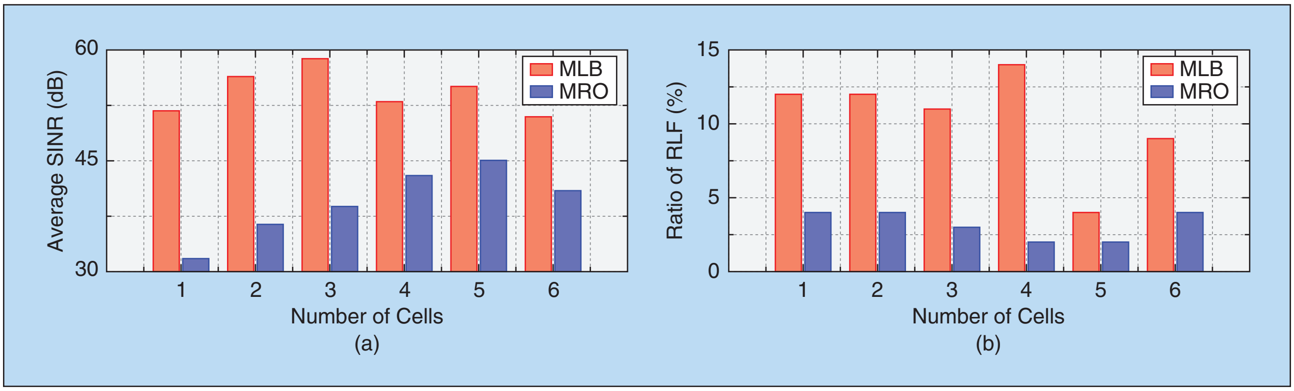


FIGURE 8 MLB-MRO performances.

region that is not on the Pareto front is not a non-dominated solution. Each objective can be improved with no penalty to the other. From these figures, we observe that the proposed scheme, referred to hereafter as BDA-NSGA-II, gives us the set of points that provides the best performance of conflicting SON functionalities, i.e., by predicting the performance metric based on the historical data collected from the network, we are able to find the operating point that provides the best performance trade-off. Moreover, Figures 10a-10c represent the parameter space. That is, each bin represents the ratio (%) of the values of the CIO, hysteresis and TTT on the Pareto front. In order to avoid misunderstanding due to the diversity in CIO and hysteresis values, we define three different indicators for Figures10a and 10b. Please note that the meaning of the x-axis labels are indicated in Table 5. In order to calculate the ratios, we consider the frequency distribution of the values of each handover parameter chosen by each SON function and the proposed scheme in the last population of solutions. Then, for each value we calculate the percentage as $(n_{\text{occurrences}}/p_{\text{size}}) \times 100$, where $n_{\text{occurrences}}$ is the number of times that each value has been selected in the solutions, and p_{size} is the size of the population itself.

TABLE 5 CIO and Hysteresis x-axis indicators.

	CIO	HYSTERESIS
LOW	[0, 10]	[0, 5]
MEDIUM	(10, 20]	(5, 10]
HIGH	(20, 30]	(10, 15]

We observe that for the case of CIO and hysteresis parameters, the medium

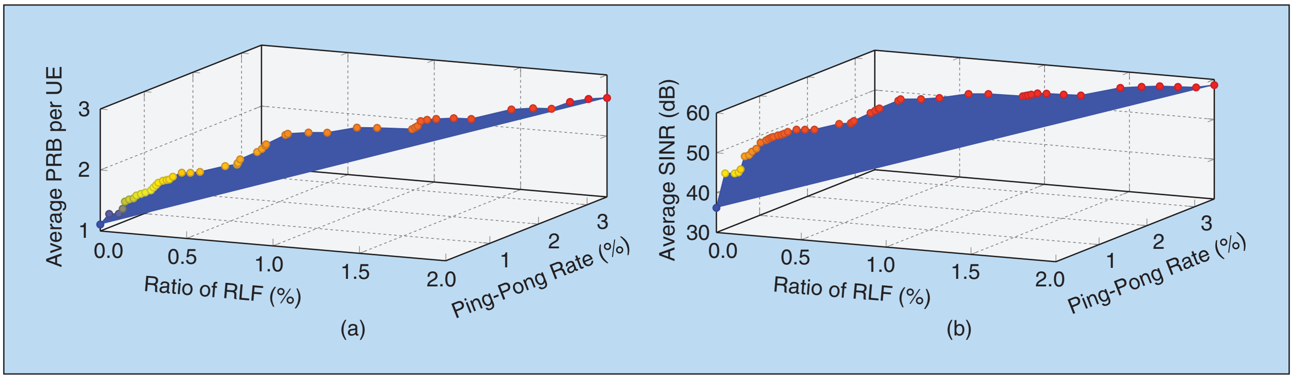


FIGURE 9 Objective space considering the average number of PRBs per UE, the ratio of RLF and ping-pong rate objectives (a), and the average SINR, the ratio of RLF, and ping-pong rate objectives (b).

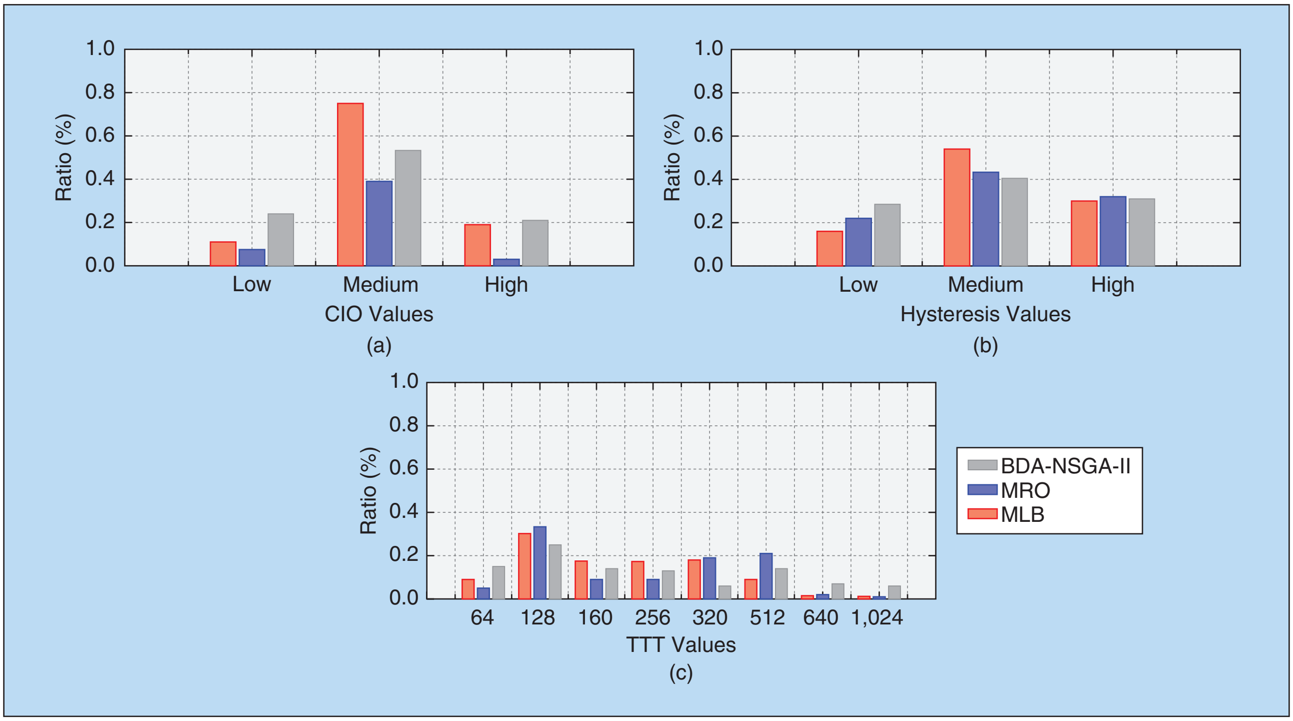


FIGURE 10 Ratio of the handover parameters across the Pareto front.

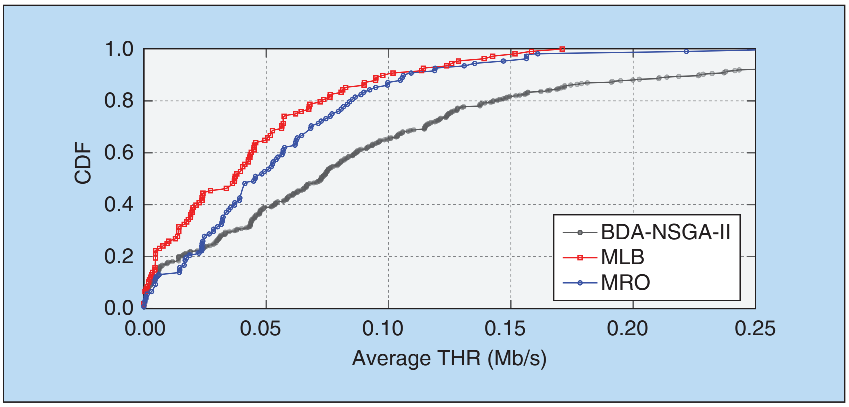


FIGURE 11 CDF of the average throughput values in the whole network.

values have the highest rate. On the other hand, for the TTT, which delays handover execution, just 8 values out of the 16 possibilities defined by the 3GPP are found, with 128 ms being the most likely one. It is important to note that all the solutions in the Pareto front are non-dominated and constitute a global solution to the SON conflict in the form of trade-off. That is to say, the tool provides a set of solutions, and the operator must choose one or another depending on the objective to be prioritized. It is a usual practice to define thresholds for each of the performance metrics and to choose a solution respecting all of them (or at least the closest one). In this particular example, we choose one of the solutions having an RLF ratio < 1%, ping-pong rate < 1%, average SINR > 15 dB and the average number of allocated PRBs > 2. Figure 11 shows the Cumulative Distribution Function (CDF) of the cell average throughput. From this figure, we observe that our proposed scheme is able to find the operating point that provides the best performance trade-off.

VI. Concluding Remarks

In this paper, we have addressed the issue of SON conflict. In particular, we focus on the SON conflict that results from the concurrent execution of multiple SON functions. The main contribution of this work is to present a framework that is able to take advantage of big data analytics, i.e., we exploit the huge amount of data already available in the network to predict future performance. We build a prediction model based on historical UE measurements, and we apply regression analysis techniques to predict network performance. The built model is then used as an input of multi-objective evolutionary algorithm to solve the potential conflicts by finding a set of solutions that satisfy the objectives at an acceptable level without being dominated by any other solution.

To evaluate the performance of the proposed scheme, we focus on the MLB-MRO SON conflict. The simulation results demonstrate the ability of the proposed scheme to solve conflicts based on a prediction of network performance, which is obtained from a proper analysis of UE measurements. As a result, the proposed scheme learns from past experience to predict network performance according to the target of each SON function and then solve the conflict based on non-dominated solutions.

Acknowledgment

This work was supported by the Spanish National Science Council and ERFD funds under TEC2014-60258-C2-2-R and TEC2017-89429-C2-2-R projects.

References

[1] "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self configuring and self optimizing network uses case and solutions (Release 9)," *3GPP*, Tech. Rep. TR 36.902, v9.2.0, 2009.

[2] J. Johansson, W. A. Hapsari, S. Kelley, and G. Bodog, "Minimization of drive tests in 3GPP release 11," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 36–43, Nov. 2012.

[3] "Study of implementation alternative for SON coordination," *3GPP* TSG SA WG5 (Telecom Management) Meeting 85, Tech. Rep. TSG S5-122330, 2012.

[4] Z. Altman, M. Amirijoo, F. Gunnarsson, H. Hoffmann, I. Z. Kovcs, D. Laselva, B. Sas, K. Spaey, A. Tall, H. van den Berg, and K. Zetterberg, "On design principles for self-organizing network functions," in *Proc. 11th Int. Symp. Wireless Communication Systems*, pp. 454–459, Aug. 2014.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[6] T. Bandh, H. Sanneck, and R. Romeikat, "An experimental system for son function coordination," in *Proc. IEEE 73rd Vehicular Technology Conf.*, pp. 1–2, May 2011.

[7] J. Moysen and L. Giupponi, "Self-coordination of parameter conflicts in D-SON architectures: A Markov decision process framework," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 18, Mar. 2015.

[8] J. Chen, H. Zhuang, B. Andrian, and Y. Li, "Difference-based joint parameter configuration for MRO and MLB," in *Proc. IEEE 75th Vehicular Technology Conf.*, pp. 1–5, May 2012.

[9] P. Muñoz, R. Barco, and I. de la Bandera, "Load balancing and handover joint optimization in LTE networks using fuzzy logic and reinforcement learning," *ELSIVER Comput. Netw.*, vol. 76, pp. 112–125, Jan. 2015.

[10] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON Coordination for parameter conflict resolution: A reinforcement learning framework," in *Proc. IEEE Wireless Communications Networking Conf. Workshops*, pp. 196–201, Apr. 2014.

[11] C. Szepesvári, *Algorithms for Reinforcement Learning, Ser. G: Reference Information and Interdisciplinary Subjects Series. Morgan Claypool*, 2010.

[12] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy, "Big data empowered self organized networks," in *Proc. 20th European Wireless Conf.*, May 2014, pp. 1–8.

[13] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE Netw.*, vol. 28, no. 6, pp. 27–33, Nov. 2014.

[14] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "A mobile network planning tool based on data analytics," *Mobile Inform. Syst.*, vol. 2017, Feb. 2017.

[15] J. Moysen, L. Giupponi, N. Baldo, and J. Mangues-Bafalluy, "Predicting QoS in LTE HetNets based on location-independent UE measurements," in *Proc. IEEE 20th Int. Workshop Computer-Aided Modeling Analysis and Design Communication Links and Networks*, Sept. 2015, pp. 124–128.

[16] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "On the potential of ensemble regression techniques for future mobile network planning," in *Proc. IEEE Symp. Computers and Communications*, June 2016, pp. 477–483.

[17] K. van Moffaert and A. Nowé, "Multi-objective reinforcement learning using sets of Pareto dominating policies," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3483–3512, Jan. 2014.

[18] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization," *Mach. Learn.*, vol. 40, pp. 139–157, Aug. 2000.

[19] V. N. Vapnik, *The Nature Statistical Learning Theory.* New York: Springer-Verlag, 1995.

[20] E. M. Jordaan and G. F. Smits, "Estimation of the regularization parameter for support vector regression," in *Proc. Int. Joint Conf. Neural Networks*, 2002, vol. 3, pp. 2192–2197.

[21] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge, U.K.: Cambridge Univ. Press, 2000.

[22] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[23] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: Univ. of Michigan Press, 1975.

[24] "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol Specification (Release 10)," *3GPP*, Tech. Rep. TS 36.331, v10.7.0, 2010.

[25] "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10)," *3GPP*, Tech. Rep. TS 36.213, v10.2.0, 2011.

CIG 2018

IEEE CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND GAMES

Maastricht University - Department of Data Science & Knowledge Engineering
Maastricht, The Netherlands
August 14-17, 2018

Call for Papers

Key Dates (2018)

January 15: Competition/ Tutorials/ Panels/ Special Sessions proposals
March 1: Deadline for abstracts
March 15: Deadline for paper submission
May 1: Author notification
May 15: Deadline for Competition/Vision/Demo papers
June 1: Deadline for camera-ready paper submission
June 15: Early bird registration deadline
August 14-17: Conference

Committees

General Chair: Mark Winands (Maastricht University)
Program Chairs: Yngvi Björnsson (Reykjavík University), Jialin Liu (Queen Mary University of London), Mike Preuss (ERCIS, WWU Münster)
Proceedings Chair: Cameron Browne (RIKEN Institute)
Tutorial and Keynote Chair: Stylianos (Stelios) Asteriadis (Maastricht University)
Special Session Chair: Santiago Ontañón (Drexel University)

Relevant Topics

- Machine-learning in games
- Adversarial search
- Player/opponent modeling
- AI in education
- Emotion recognition in game-play
- CI/AI-based game design
- Affective modeling
- Player experience
- Procedural content generation
- Game generation
- Intelligent interactive narrative
- Character development and narrative
- CI/AI for virtual cinematography
- Multi-agent and multi-strategy learning
- Applications of game theory
- General game playing
- Serious games

maastricht: CONVENTION BUREAU

IEEE Computational Intelligence Society

Contact:
cig2018@maastrichtuniversity.nl
Department of Data Science & Knowledge Engineering
Faculty of Humanities and Sciences
Maastricht University, The Netherlands

MaTHiSiS

Gemeente Maastricht

UM Department of Data Science & Knowledge Engineering

For more information, visit www.maastrichtuniversity.nl/cig2018/

Digital Object Identifier 10.1109/MCI.2018.2807081

Research Frontier

David Hübner
Brain State Decoding Lab, University of Freiburg, Freiburg, GERMANY

Thibault Verhoeven
Electronics and Information Systems, Ghent University, Ghent, BELGIUM

Klaus-Robert Müller
Machine Learning Group, Berlin Institute of Technology, Berlin, GERMANY
Department of Brain and Cognitive Engineering, Korea University, Seoul, KOREA
Max Planck Institute for Informatics, Saarbrücken, GERMANY

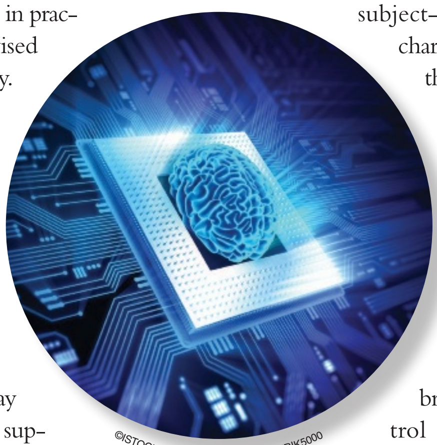
Pieter-Jan Kindermans
Machine Learning Group, Berlin Institute of Technology, Berlin, GERMANY

Michael Tangermann
Brain State Decoding Lab, University of Freiburg, Freiburg, GERMANY

Unsupervised Learning for Brain-Computer Interfaces Based on Event-Related Potentials: Review and Online Comparison

Abstract

One of the fundamental challenges in brain-computer interfaces (BCIs) is to tune a brain signal decoder to reliably detect a user's intention. While information about the decoder can partially be transferred between subjects or sessions, optimal decoding performance can only be reached with novel data from the current session. Thus, it is preferable to learn from unlabeled data gained from the actual usage of the BCI application instead of conducting a calibration recording prior to BCI usage. We review such unsupervised machine learning methods for BCIs based on event-related potentials of the electroencephalogram. We present results of an online study with twelve healthy participants controlling a visual speller. Online performance is reported for three completely unsupervised learning methods: (1) learning from label proportions, (2) an expectation-maximization approach and (3) MIX, which combines the strengths of the two other methods. After a short ramp-up, we observed that the MIX method not only defeats its two unsupervised competitors but even performs on par with a state-of-the-art regularized linear discriminant analysis trained on the same number of data points and with full label access. With this online study, we deliver the best possible proof in BCI that an unsupervised decoding method can in practice render a supervised method unnecessary. This is possible despite skipping the calibration, without losing much performance and with the prospect of continuous improvement over a session. Thus, our findings pave the way for a transition from supervised to unsupervised learning methods in BCIs based on event-related potentials.


©ISTOCKPHOTO.COM/ARTIST HENRIK5000

I. Introduction

Many applications in the field of human-device interaction need a calibration phase prior to the actual usage of the application. During calibration, the user is requested to perform a series of predefined tasks in order to collect example data, for which the user's intentions are known. Machine Learning (ML) methods then use this labeled data to learn the subject-specific brain signal characteristics and predict the user's intention on new unseen data. In the field of Brain-Computer Interfaces (BCI), these predictions can enable the user to control applications and physical devices by translating brain activity into control commands [1]–[5]. One research goal in BCIs is to replace lost or faulty neurobiological pathways, e.g., for patients with Amyotrophic Lateral Sclerosis (ALS) [6]–[8], by computational intelligence, for instance, to restore communication [2], [9] or control a wheelchair [10].

Digital Object Identifier 10.1109/MCI.2018.2807039
Date of publication: 10 April 2018

Corresponding authors: Pieter-Jan Kindermans (E-mail: p.kindermans@tu-berlin.de) and Michael Tangermann (E-mail: michael.tangermann@blbt.uni-freiburg.de)

Calibration is challenging in BCI, because the signal-to-noise ratio (SNR) is unfavorable and the subject-to-subject variability is large [5]. Depending on the type of paradigm chosen, the calibration time can differ between minutes [11] to multiple sessions [12]. Even though it was shown that the calibration time can be partly reduced by transferring brain signals from within the same subject [13] or other subjects [14]–[16], a rest of subject- and session-specific variation remains to be learned.

To tackle this learning challenge, different strategies have been proposed. They can be subdivided into two groups: The first group takes a pre-trained classifier and updates it with unlabeled new data from the current session [17]–[23]. We refer to this approach as *unsupervised adaptation*. Algorithms implementing unsupervised adaptation rely on the assumption that suitable training data is available or can be recorded in order to pre-train a classifier. However, for subjects with limited attention span or atypical brain patterns, e.g., stroke survivors, this might not be the case. To overcome this limitation, a second group of algorithms was recently proposed for BCIs. These algorithms can learn the individual brain characteristics from scratch without requiring any labeled data at all [24]–[30]. We refer to them as *unsupervised learning* methods. They are a generalization of the first group of algorithms as they can also be initialized with good parameters obtained via transfer learning. See Fig. 1 for an illustration of the difference between the two groups.

Both approaches are able to update their decoding model during the actual usage of the BCI application and hence, also to adapt to changing signals over time. To accomplish this, the ML method is required to learn from unlabeled data, i.e., when the user's intentions are unknown. The BCI community seems to be in need of unsupervised decoding methods: The review of Nicolas and colleagues [31] identified unsupervised adaptation as one of the six "key challenges for BCI deployment outside the lab". Millán and colleagues [32] stress the importance of adaptation for skill-learning in BCI (e.g., for rehabilitation after stroke), as it "increases the likelihood of providing stable feedback to the user, a necessary condition for people to learn to modulate their brain activity". Unsupervised learning also bears the potential of exploiting large unlabeled data sets to find common brain patterns—a key ingredient for developing true plug & play BCI systems.

A different line of work explores strategies to adapt the policy of the interaction between user and computer instead of adapting the brain signal decoder [33]–[38]. These *policy adaptation* approaches rely on the detection of error-related potentials, i.e. signals, that reflect the observation of an error, in order to infer the correct or intended actions of the user.

In our paper, we first carefully review the attempts of unsupervised adaptation and unsupervised learning for decoding event-related potentials (ERPs)—brain responses that are widely used in BCI paradigms. Then we present results from an online BCI study based on recordings of the electroencephalogram (EEG), comparing three different unsupervised learning methods that build its decoders from scratch. For the first time, we demonstrate that unsupervised learning methods in BCI can—in practice—utilize unlabeled data as efficiently as a state-of-the-art supervised method after an initial ramp-up. With that contribution, we strive to further increase the usability of BCI systems in practice.

A. Event-Related Potentials

ERPs are evoked transient brain responses to a sensory, cognitive or motor event. One way of eliciting them is to present external stimulation events to the user, e.g., by visually highlighting symbols on a computer screen [9], or by presenting sounds [39]. By mapping each symbol or sound to a control command, the user can select an action by focusing his attention on the corresponding stimulus event. This makes it possible to control a wide range of applications based on visual ERPs, e.g., for spelling [9], [40], web browsing [41], games [42], [43], browsing and sharing pictures [44], predicting emergency brakes in a driving scenario [45], controlling objects in a virtual environment [46], [47] and artistic expression through painting [6], [8].

Visual ERP-based BCIs have several desirable features [5], [11]: (a) They require virtually no subject training, (b) can be realized with standard hardware,

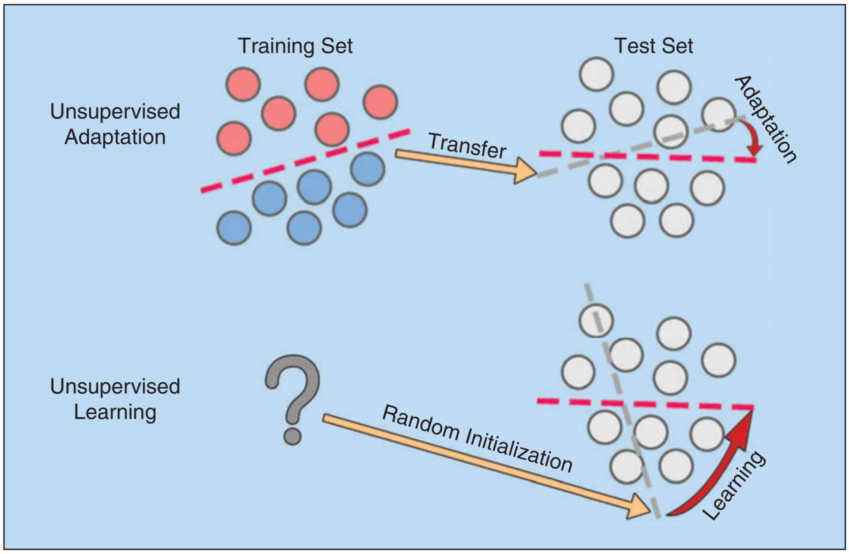


FIGURE 1 Unsupervised learning vs. unsupervised adaptation. Red and blue dots indicate historic labeled training data from two classes. Grey dots depict unlabeled data. Dashed lines indicate classification models. The general goal is to find a model which separates the two classes as good as possible. Label information is necessary only in the adaptation scenario. For transferring the classification model, only a slight adaptation may be necessary while the unsupervised learning algorithm has to learn the model from a random initialization.

(c) have a high user acceptance, (d) generally need less than 10 minutes to be calibrated [11] and (e) are effective for almost all healthy users [48] and for many patients with ALS [6]–[8]. Overall, BCIs based on visual ERPs are widely used, even though faster alternatives exist in terms of information transfer [49]. Examples are code-modulated visual evoked potentials (c-VEP) [50] also including broad-band stimulation paradigms [51] and paradigms based on steady state visual evoked potentials (SSVEP). For the latter, Chen et al. [52] have demonstrated how users can write up to one character every second. The c-VEP and SSVEP approaches, however, require a high temporal precision of the visualization hardware and a high level of gaze control. SSVEP stimulation can be perceived as a high workload and—due to its flickering characteristics—may even evoke seizures in epileptic users.

Importantly, visual ERP-based BCIs often have the advantage that the stimulus presentation mode leads to a special structure of the collected brain signal data, which can be exploited by unsupervised learning methods. For instance, in the case of the well-known P300 speller by Farwell and Donchin [9], the user can select to spell between 36 symbols which are arranged in a 6 × 6 grid by focusing his attention on the target letter. Rows and columns are highlighted in alternating order. A complete highlighting round of 12 events is called an *iteration*. Typically, multiple iterations are necessary to uniquely determine the attended character. This highlighting scheme is inducing constraints on the data, e.g., exactly one row and one column of the symbol grid will contain the selected letter while five rows and columns do not contain it. Also, knowing the selected symbol uniquely determines each event as being attended (target) or not-attended (non-target). These and more constraints allow for efficient learning from unlabeled data in ERP-based BCIs, something which is not yet sufficiently explored in the oscillatory domain. For an analysis of transfer learning and unsupervised adaptation in the oscillatory domain, which is not in the scope of this review, we refer to the work by Lotte and colleagues [53].

II. Review of Related Work

Different attempts have been undertaken to accomplish learning from unlabeled data in ERP-based BCIs. We begin by reviewing examples from the group of *unsupervised adaptation* techniques (sometimes also referred to as semi-supervised [53] methods), before discussing *unsupervised learning* approaches. We want to emphasize that all unsupervised methods can be used for an ordinary visual P300 speller unless specified otherwise. The ML model is hidden from the user such that the interaction between user and computer remains the same except for the quality of the control signals.

TABLE 1 Overview of unsupervised adaptation and unsupervised learning methods for event-related potentials.

MAIN CONCEPT	REFERENCE
UNSUPERVISED ADAPTATION	
1) NAÏVE LABELING: ADAPTATION BASED ON PREDICTED LABELS	LU, 2009, [17]; KINDERMANS, 2011, [54]
2) CO-TRAINING TWO CLASSIFIERS BASED ON PREDICTED LABELS	PANICKER, 2010, [18]
3) USAGE OF ERROR-RELATED POTENTIALS AS LABEL INFORMATION	ZEYL, 2016, [19]
4) POOLED MEAN & COVARIANCE ADAPTATION DISREGARDING LABELS	VIDAURRE, 2011, [20]; IN ERP: DÄHNE, 2011, [21]
5) ALTERNATIVELY ESTIMATING CSP AND RIEMANNIAN CLASSIFIER	BARACHANT, 2014, [55]; IN MEG: BOLAGH, 2016 [22]
UNSUPERVISED LEARNING	
1) EXPLOITING TASK CONSTRAINTS AND ERROR-RELATED POTENTIALS	GRIZOU, 2014 [28], [29]; ITURRATE, 2015 [30]
2) UTILIZE DATA CONSTRAINTS WITH EXPECTATION-MAXIMIZATION (EM)	KINDERMANS, 2012, 2014 [24], [25]
3) MODIFY PARADIGM TO LEARN FROM LABEL PROPORTIONS (LLP)	HÜBNER, 2017, [26]
4) MIX: COMBINE THE MEAN ESTIMATIONS FROM EM AND LLP	VERHOEVEN, 2017, [27]

A. Unsupervised Adaptation for ERP

Unsupervised adaptation always relies on a classifier that has been pre-trained on supervised data from the same or other subjects. For transferring it to a novel user or to the next session, the pre-trained classifier is then adapted using unlabeled data gained during the usage of the BCI application. An overview of currently published methods is given in the top part of Table 1.

1) Naïve Labeling

Lu et. al [17] proposed an approach in which a subject-independent classifier is first trained on historic data and then used to predict the labels for newly recorded ERP signals. Assuming that these predictions are correct, the model is then retrained with the new data to obtain an updated classifier. Obtained labels are called "naïve" as it is uncertain whether they are correct or not. To measure the degree of uncertainty, Lu et al. introduced a confidence score that is measuring how consistently the labels were predicted during the spelling of one letter. Only when a high consistency is observed, they trusted an estimated label. Otherwise, the unlabeled data was discarded. While their approach worked well in an offline study using a visual spelling paradigm with 10 healthy subjects, it can be expected to have severe problems when the initial accuracy is close to chance level, e.g., in patient data or in auditory ERP data with a low SNR. In this case, the instability of the labeling can cause runaway errors [56]. The self-labeling approach was also used by Kindermans et al. [54] for a class re-weighted version of the Ridge regression and was shown to outperform a non-adaptive classifier on the BCI Competition III data set [57].

2) Two-classifier Co-training Approach

Panicker et al. [18] extended the idea of naïve labeling using two classifiers—Fisher linear discriminant analysis and Bayesian linear discriminant analysis—which co-train each other. To do so, both classifiers are first initialized on a labeled training data set. Then both classifiers determine the labels for a chunk of unseen and unlabeled data points. These points with corresponding estimated labels are then added to the current training data set of the *other* classifier and both classifiers are retrained. This procedure is repeated until convergence or until the improvements (measured by a confidence score) are minimal. The authors evaluated this approach using an offline visual ERP speller study with data from five healthy subjects. For this relatively small number of subjects, it was found that the co-training approach outperforms the naïve labeling strategy of a single classifier in most situations, however, runaway errors may still occur.

3) Pooled Mean and Covariance Adaptation

Vidaurre et al. [20] suggested an unsupervised adaptation method of a linear discriminant analysis (LDA) classifier. LDA assumes a class-wise normal distribution with class means $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ and shared covariance matrix Σ_C. It finds a linear hyperplane defined by the orthogonal vector $\boldsymbol{w}$ by computing [58]:

$$\boldsymbol{w} = \Sigma_C^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \tag{1}$$

Empirical data shows that the assumptions made by the LDA are closely met by ERP data [58] and hence, LDA is a widely used and competitive classifier in BCIs [58], [59]. One can show that replacing the shared covariance matrix Σ_C by the global covariance Σ, which disregards label information, leads to the same direction of $\boldsymbol{w}$ given the correctly recovered class means. Technically, this can be understood as a least squares classifier with re-scaled outputs [60]. For that reason, Vidaurre et al. proposed an adaptation scheme which adapts either only the common class means or both, the class means and the global covariance matrix in an unsupervised fashion. This approach was shown to outperform a fixed supervised classifier on motor imagery data both in simulations and online. It can readily be applied to ERP data as demonstrated in [21].

4) Adaptation Based on Error-Related Potentials

When the user perceives a mistake, e.g., when an incorrectly spelled letter was shown to the user, a time-locked error-related potential (ErrP) can be observed. These ErrPs can be decoded with an accuracy of around 80% [61]–[63] and—depending on the application—may be useful to automatically correct detected errors [64]. Initially proposed for code-modulated visual evoked potentials, Spüler et al. [65] proposed to ignore the data, if an ErrP is detected after showing the predicted character since the true class label is unknown and the estimated class label is suspected to be wrong. Other groups used ErrPs to adapt the policy of a virtual or real robot in order to achieve a certain goal [33]–[38]. In Section II-B1, we review an approach that can jointly learn to decode ErrPs and to adapt its policy to control a device.

Recently, Zeyl et al. [19] compared an adaptation of the decoder based on (a) ErrPs, (b) a naïve-labeling approach based on target confidence and (c) a hybrid approach which combines (a) and (b) in a visual ERP speller. The problem with exploiting ErrPs in the context of the classical visual ERP speller is that feedback signals are only shown at the end of each trial, and hence, ErrPs are harvested rarely compared to the number of presented stimulus events. To alleviate this mismatch, Zeyl and colleagues proposed to show both the row and the column selection as two separate decisions to the user to collect ErrPs more frequently. Interestingly, an offline analysis and a simulated online experiment with 11 healthy subjects showed that the naïve-labeling approach performed best, with the hybrid approach close behind and the pure ErrP approach significantly worse. This indicates that additional information from the ErrPs could not contribute in improving the adaptation in this specific experimental scenario.

5) Alternatively Training a Spatial Filter and Riemannian Classifier

Barachant and colleagues proposed an information theoretical framework which allows measuring distances between trial covariance matrices based on concepts of the Riemannian geometry [66]. The use of this representation and Riemannian distance has the advantage of being invariant under affine transformations which would not be the case in the original Euclidean space. Supervised classifiers operating on Riemannian distances have been successfully applied to ERP signals [55]. Although mentioned as an option, unsupervised adaptation was not implemented in their work on EEG-based ERP data [55], but it was implemented successfully on magnetoencephalography (MEG) data by authors around Bolagh from the same group [22]. Again, the premise is that labeled historic data from earlier subjects is available which is used to obtain an initial estimate of the novel unlabeled data.

An iterative two-step procedure for estimating these labels is at the core of their approach. It makes use of a widely-used spatial filtering method, Common Spatial Patterns (CSP) [67]. As this algorithm requires labels, which are not available in an unsupervised adaptation approach, the current label estimates are used in every iteration of the procedure. The first step involves to replace the original trials by new "super trials". These are formed by CSP-filtered original trials, enlarged by the two CSP-filtered class means. Super trials are then used to calculate the so-called feature covariance matrices (one per trial). The second step takes place in Riemannian space, where distances between these novel feature covariance matrices and mean covariance matrices can be computed. A Riemannian classifier based on labels of the last iteration (or on labels of historic data in case of the first iteration) is used to update the label estimate of each trial. These two steps are repeated until convergence.

This approach won the open "DecMeg2014" Kaggle competition. It could easily be transferred to EEG data.

B. Unsupervised Learning of ERP

We now address the second group of classifiers. Unsupervised learning approaches can learn the model parameters without requiring any labeled data at all, not even historical data. This type of learning is substantially more difficult as no initialization or prior information of the parameters is available compared to the approaches described in the previous section.

Assuming a two-class problem with high SNR, one could imagine an obvious approach: applying a clustering algorithm would allow splitting the data into two groups, e.g., by assuming a Gaussian distribution of each class. One could then further identify the two clusters as target and non-target classes by utilizing structure imposed by the experimental design. In case of ERP paradigms, fewer data points can be expected in the cluster formed by target points compared to the non-target cluster.

However, given the low SNR in ERP-based EEG recordings, this obvious approach would require an enormous number of data points. Practically, it is not feasible. Instead, unsupervised learning methods need to exploit the data constraints provided by the ERP application as good as possible. Only this information allows them to solve the classification task despite the low SNR and missing labels.

We review four algorithms that implement unsupervised learning: (1) an approach combining task constraints with ErrPs, (2) the probabilistic expectation-maximization algorithm, (3) the deterministic learning from label proportions—which requires a modification of the paradigm—and (4) the combination of the latter two algorithms. An overview is shown in the bottom part of Table 1.

1) Exploiting Task Constraints and ErrPs

The calibration-free approach by Grizou [28], [29] and Iturrate [30] is able to simultaneously calibrate the system while the user controls the BCI by making intelligent use of the given task constraints and ErrPs. The authors demonstrated the feasibility of the approach on a virtual 5 × 5 grid where the user should move a cursor to a goal position [30]. Users achieve control by monitoring the moving cursor and passively assessing whether it moves in the right or wrong position. In the latter case, an error-related potential is automatically elicited by the user. Detecting those ErrPs would allow a BCI controller to determine the goal position. Now, the learning task is to simultaneously infer the unknown goal position as well as to train an ErrP decoder. This chicken-and-egg problem is solved by utilizing the observation that each of the 25 possible goal positions should lead to a different sequence of elicited ErrPs, thus providing only 25 possible ways to label the ErrP data. Their algorithm then assigns a higher likelihood to data sets that are most consistent, where consistency is measured as the separability between the two classes (correct or incorrect direction). The goal position desired by the user is the one associated with the most consistent data set, which can, in turn, be used to update the parameters of the ErrP classifier. An online study with eight healthy subjects showed that this method allowed users to correctly navigate the cursor to more goals compared to a scenario where a supervised adaptation was conducted prior to the experiment and with the same total experiment time. Although their navigation problem is formulated in a grid shape, this technique was not yet applied to any ERP-based spelling paradigm, see [37].

2) Expectation-Maximization

The approach by Kindermans and colleagues [24] also simplifies the overall learning task by trying to infer the latent variable (selected symbol) of a matrix speller rather than solving the more complicated problem to decide for each stimulus whether it was a target or non-target. This reduces the number of possible configurations from an exponentially growing number in the latter case, to a limited one–36 in the case of the original visual ERP speller. Importantly, the number of possible configurations only depends on the grid size, and does not change when more iterations are recorded to spell one character. With this constraint in mind, Kindermans et al. proposed to use a version of Bayesian least square regression [24], [60] which assumes that the feature vectors can be linearly projected onto two one-dimensional Gaussian distributions (one for targets and one for non-targets), which share the same within-class variance. In the original formulation, the goal of the decoder training is then to find that linear projection $\boldsymbol{w}$ and per-class variance β which maximizes the probability of observing the data given the model parameters.

The learning task is tackled by utilizing an expectation-maximization (EM) algorithm which alternatively estimates the probabilities of the latent variables—which letter was selected by the subject—during the expectation step [E-step] and optimizes the parameters given these probabilities in the maximization step [M-step]. The EM procedure is repeated until convergence. This method can be seen as a mathematically rigorous version of the naïve labeling approach from Section II-A1.

An online study with 10 young healthy users showed that the EM algorithm can successfully decode auditory ERP signals from scratch without any label information [25]. Given a sufficient amount of data, the EM approach can compete with a supervised classifier. In cases when non-stationarities occur in the data [25], the EM has the potential to outperform a non-adaptive supervised classifier. However, even though EM works well in practice, no guarantees on the decoding quality can be provided, and practically the EM algorithm often converges to unfavorable local maximums—especially when only a limited amount of unlabeled data is available.

3) Learning from Label Proportions

The third method, learning from label proportions (LLP), was initially proposed by Quadrianto et al. [68] and was first applied to BCI by Hübner et al. [26]. It is based on the idea that the unlabeled data points derived during the online use of a BCI can be subdivided into groups, where every group displays a different ratio of target to non-target data points. To enable this approach for a visual matrix speller, a modification of the spelling paradigm is necessary. Hübner et al. [26] proposed to modify the spelling interface with the following three adjustments: (a) The spelling matrix is extended by 10 additional #-symbols which serve as visual blanks and should never be attended by the user—as such they are non-targets by construction. (b) Instead of using a row-column highlighting scheme, the flexible highlighting scheme by Verhoeven et al. [69] is used. (c) Each trial (i.e., spelling one character) is composed of two interleaved highlighting sequences, where sequence 1 only highlights normal character (it does not highlight #-symbols) and sequence 2 highlights normal characters as well as #-symbols. These modifications lead to two subgroups of EEG data, namely the epochs measured during sequence 1 and epochs from sequence 2. The groups show different but known target- to non-target ratios, which are stored in a mixing matrix $\mathbf{\Pi}$ and are known from constructing the sequences. The mean ERP responses of sequence 1 and 2 ($\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$) are then given by a linear combination of the target and non-target class means $\boldsymbol{\mu}_T, \boldsymbol{\mu}_N$ as

$$\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} = \mathbf{\Pi} \begin{bmatrix} \boldsymbol{\mu}_T \\ \boldsymbol{\mu}_N \end{bmatrix}, \text{ where } \mathbf{\Pi} := \begin{bmatrix} \pi_T^1 & \pi_N^1 \\ \pi_T^2 & \pi_N^2 \end{bmatrix}. \tag{2}$$

By inverting $\mathbf{\Pi}$ and computing the sample means of the two sequences $\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2$, unsupervised estimates for the class means are obtained. Importantly, the LLP approach comes with the theoretical guarantee of converging to the right class means given independent and identically distributed data points [26]. As a final step in the approach of Hübner et al., a modification of the linear discriminant analysis—using the global covariance instead of the shared covariance matrix similar to the case in Section II-A2—is used to compute the desired projection vector $\boldsymbol{w}$.

An online study with 13 healthy subjects showed that LLP could reliably learn the classifier weights from scratch [26]. Simulated online experiments revealed, that this unsupervised learning approach initially outperforms the EM-approach, but falls behind when more and more data is available from extended online use.

4) Mixture Method

The mixture method (MIX) by Verhoeven et al. [27] describes an analytical combination of the EM and LLP method. It is built on the observation that the previously explained two methods, EM and LLP, have complementary strengths and weaknesses. It uses a reformulation of the EM algorithm which is explicitly estimating the class means instead of the projection only. In the MIX method, the estimation of the class-wise means is proposed as a linear combination of the mean estimations found with the EM ($\boldsymbol{\mu}_{EM}$) and those estimated by the LLP method ($\boldsymbol{\mu}_{LLP}$),

$$\hat{\boldsymbol{\mu}}_{MIX}(\gamma) = (1-\gamma)\hat{\boldsymbol{\mu}}_{EM} + \gamma\hat{\boldsymbol{\mu}}_{LLP} \tag{3}$$

where $\gamma \in [0,1]$ denotes the mixing coefficient. The coefficient γ is found by minimizing the expected mean squared error between the estimated value $\hat{\boldsymbol{\mu}}_{MIX}$ and the unknown true parameter value $\boldsymbol{\mu}$. Verhoeven and colleagues showed, that this approach leads to an analytical formulation for the optimal mixture coefficient $\gamma^\star$ [27]:

$$\gamma^\star = \frac{1}{2} + \frac{\sum_d Var[\hat{\boldsymbol{\mu}}_{EM,d}] - \sum_d Var[\hat{\boldsymbol{\mu}}_{LLP,d}]}{2\|\hat{\boldsymbol{\mu}}_{EM} - \hat{\boldsymbol{\mu}}_{LLP}\|^2}. \tag{4}$$

Here, d denotes the feature dimensionality, and $Var[\hat{\boldsymbol{\mu}}_{(\cdot),d}]$ denotes the variance (over different realizations of the data) of the estimator for the dth entry of the estimated mean $\hat{\boldsymbol{\mu}}_{(\cdot)}$. This variance is a measure of the uncertainty of the estimated value. The higher the uncertainty on the output of the LLP method, the higher the weight given to the output of the EM method and vice versa. To estimate the variance in LLP, one can derive a closed-form solution which only depends on the mixing matrix and data variance. For the EM, no closed-form solution exists. Additionally, only one realization of the data is observable in practical applications, and simulating other realizations is time-consuming and inaccurate. Hence, the authors used the approximation that the EM-estimator converges asymptotically to a Gaussian distribution where the variance can be computed based on the data [27].

Verhoeven et al. [27] compared LLP, EM and MIX in offline simulations on data of 13 subjects which is openly available at the Zenodo database[1]. It was found that the MIX method does not only combine the strengths of EM and LLP but that it actually transcends the two single performances for almost any amount of data on the group grand average. Interestingly, the simulations also showed that the MIX performance can compete with a supervised classifier which had the complete label information available and has been trained on the same amount of training data as MIX after a short ramp-up.

In this section, we reviewed the most promising unsupervised adaptation and unsupervised learning methods for ERP-based BCIs. It is of course also possible to combine the best of the two worlds. While an unsupervised adaptation method *crucially* relies on a pre-trained classifier, unsupervised learning methods can certainly also benefit from a good initialization of the model parameters based on historic labeled (or unlabeled) data. A study by Kindermans et al. [70] showed how transfer learning significantly improves the EM algorithm. Even if the model is initialized poorly, unsupervised learning methods have a high chance of learning a good classifier which is not the case for unsupervised adaptation methods.

III. Methods

The main goal of the current paper is to verify the promising results of the MIX study in an online study. Even though authors of offline simulations try hard to

[1] DOI: http://doi.org/10.5281/zenodo.192684.

avoid overfitting pitfalls, only the evaluation in an online experiment can serve as a gold standard to validate novel decoding methods. Thus, we describe in the following an online study which compares the unsupervised learning approaches, EM, LLP and MIX in a visual ERP speller.

A. Participants

In a single experimental online session per participant, a copy-spelling task was executed by 12 healthy volunteers (8 female, 4 male, mean age: 26 years, age range: 19–31years). The BCI used a visual ERP paradigm in order to elicit target- and non-target components in the measured EEG signal. Two of the subjects (S2, S8) had prior EEG experience. The ethics committee of the University Medical Center Freiburg approved the study. All participants gave written informed consent prior to participation. A session took about 3 hours (including the EEG set-up and washing the hair), and participants were compensated with 8 Euros per hour.

B. Experimental Paradigm and Classifier Updates

An overview of the experimental structure is given in Fig. 2**B**. As the setup is very similar to the one used in [26], we restrict our description to the essential information. Within a single session, a subject was asked to spell the beginning of a sentence in each of three blocks. The text consists of the 35 symbols "Franzy jagt im Taxi quer durch das". Each block, one of the three decoding algorithms (EM, LLP, MIX, see Section II-B) was used in order to guess the attended symbol. The order of the blocks was pseudo-randomized over subjects, such that each possible order of the three decoding algorithms was used twice. This randomization should reduce systematic biases by order effects or temporal effects, e.g., due to fatigue or task-learning.

A trial describes the process of spelling one character. Each of the 35 trials per block contained 68 highlighting events. The stimulus onset asynchrony (SOA) was 250 ms and the stimulus duration was 100 ms leading to an interstimulus interval (ISI) of 150 ms. The chosen SOA was rather long compared to other visual ERP spellers and could be reduced to speed up spelling, but we wanted to ensure comparability to our previous studies [26], [27]. The classifier was randomly initialized at the beginning of each block. Classifiers were updated at the end of every trial after the target character has been estimated. The LLP classifier was always retrained from scratch using all unlabeled data while the EM classifier was updated iteratively using the previous classifier as initialization and all unlabeled data for updating. Although the copy-spelling task in principle would have allowed access to the true labels, they were not provided to any of the three classifiers at any time during the online session. However, label information was utilized to assess performances in offline analysis afterward.

Highlighting sequences were generated using the flexible framework by Verhoeven et al. [69] which ensures that the given target to non-target ratios in S1 and S2 are obtained and furthermore, tries to minimize the number of double flashes and neighboring flashes. Please see also our previous work [26] for a more detailed description of how the flexible highlighting sequences are utilized to support LLP. The actual highlighting effect consists of a combination of brightness enhancement, rotation, enlargement and a trichromatic grid overlay, which has been reported favorable to brightness highlighting alone [44].

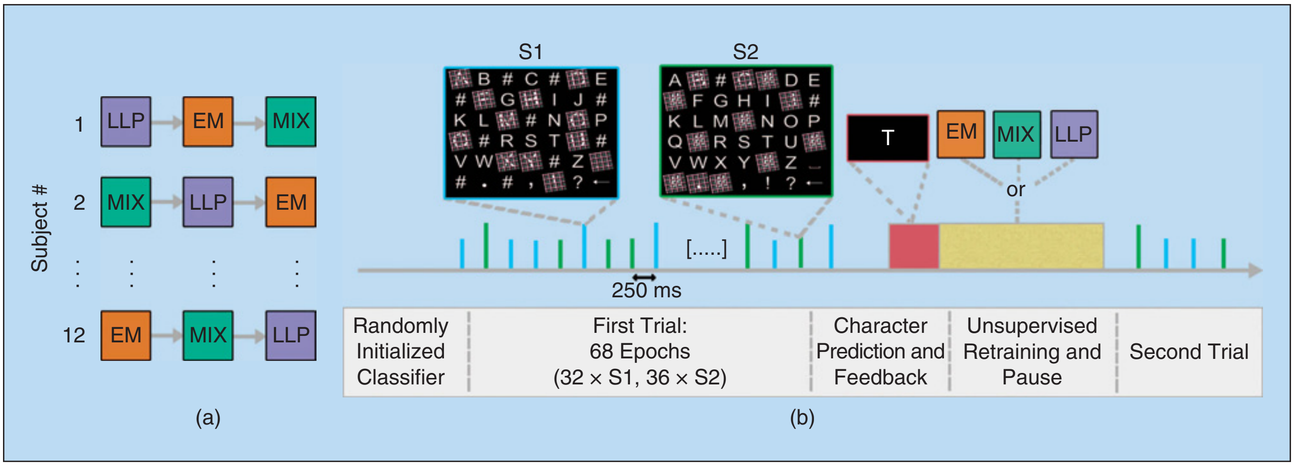

Subject #
1
LLP
EM
MIX
2
MIX
LLP
EM
12
EM
MIX
LLP
(a)
S1
S2
T
EM
MIX
LLP
or
[.....]
250 ms
Randomly Initialized Classifier
First Trial: 68 Epochs (32 × S1, 36 × S2)
Character Prediction and Feedback
Unsupervised Retraining and Pause
Second Trial
(b)

FIGURE 2 Structure of the online experiment. (a) Each subject performed three experimental blocks. Each block used a different unsupervised classifier (Expectation-Maximization (EM), Learning from Label Proportions (LLP) or a Mixture of the two (MIX)). (b) At the start of a block, the corresponding classifier was initialized randomly. The speller was modified to allow the application of LLP: One trial (i.e., spelling one character) consisted of pseudo-randomly interleaved highlighting events drawn from sequence 1 (S1, indicated by blue vertical bars) or sequence 2 (S2, green bars). In S1, a highlighting event operated on ordinary characters only, while events from S2 also highlighted "#" symbols. As the "#" symbols are not contained in the copy-spelling text, they always remain in a non-target role. This modification leads to different target to non-target ratios in S1 and S2 enabling LLP. Attended (target events) and not attended stimuli (non-target events) are indicated by shorter and longer bars, respectively. This label information, however, remained unknown to the classifiers. After each trial, the classifier predicts a character which is shown to the user. This is followed by an update of the classifier (see text).

The MIX method performs best for almost all subjects and is able to consistently reach a high decoding accuracy with an average of around 80% after data of around seven characters has been recorded.

C. Implementation

We adopted the implementations of the algorithms as proposed for EM [25], LLP [26] and the MIX method [27]. Please note the following. Since the EM algorithm relies on a good (random) initialization, Kindermans et al. [25] proposed to initialize five classifier pairs in parallel, thus, increasing the chance of having a good random initialization. They also proposed to optimize the per-class variance β directly. In contrast, the goal of the study by Verhoeven et al. [27] was to compare the mean estimations of the three methods. Hence, the authors of the latter study used only one classifier and applied the same linear classifier with covariance-shrinkage by Ledoit-Wolf [58], [71] to all three classifiers. In this paper, we used the original implementation of the EM algorithm with five randomly initialized pairs of classifiers because we observed a better performance and hence, have a fairer comparison.

For the online experiment, the algorithms were implemented in the BBCI toolbox [40] in Matlab. A simplified code version for offline analysis (without external toolboxes) is available for Matlab[2].

D. Experimental Setup, Data Acquisition and Processing

Subjects were sitting at 80 cm distance from a 24-inch flat screen. The EEG signals from 31 passive Ag/AgCl electrodes (EasyCap) were recorded, which were placed according to the extended 10–20 system, and whose impedances were kept below 20 kΩ. The signals were recorded and amplified by a multichannel EEG amplifier (BrainAmp DC, Brain Products) at a sampling rate of 1 kHz. An optical sensor on the screen indicated the exact starting time point of each highlighting event.

The collected data was band-pass filtered with a third order Chebyshev Type II filter between 0.5 and 8 Hz and downsampled to 100 Hz. Epochs were windowed to [−200, 700] ms relative to the stimulus onset and corrected for baseline shifts observed in the interval [−200, 0] ms. For each channel, the mean amplitudes of six intervals ([50, 120], [121, 200], [201, 280], [281, 380], [381, 530] and [531, 700] ms relative to the stimulus onset) were computed as features. As strict instructions of participants during the online session regarding the avoidance of obvious artifacts seemed effective, we refrained from rejecting any epochs in the preprocessing of the online session and offline analysis after the experiment. Apart from the described ones, no further preprocessing steps were applied.

E. Performance Scores

The accuracies of the three different unsupervised classifiers were assessed with two different metrics. First, the spelling accuracy was computed, which simply indicates whether a character was spelled correctly or incorrectly. Second, we computed the Area Under the Curve (AUC) of the receiver operating characteristic curve for discriminating between target and non-target epochs. The range of the AUC is between 0 and 1, where 0.5 indicates the theoretical chance level and 1 indicates perfect separability, i.e., each event can be classified correctly as target or non-target.

IV. Experimental Results

We start by presenting the results of the online study. We observed that the group-averaged visual ERP responses upon target and non-target stimuli (data not shown) are very similar to the ones reported in our previous work [26] with respect to the latencies of ERP peaks, their amplitudes and spatial locations of peaks on the scalp. This similarity is expected, as we have used the same highlighting scheme and a similar group of subjects.

Regarding classification, Fig. 3**A** shows the target vs. non-target classification accuracies for each subject and each of the three unsupervised learning method and Fig. 3**B** shows the grand average over the 12 subjects. While LLP reliably improves in the beginning but only shows slow learning over time, the EM algorithm performs more dichotomous. Depending on the random initialization, the classifier can either find the projection very early (S7) or only relatively late (S6, S9). The MIX method performs best for almost all subjects and is able to consistently reach a high decoding accuracy with an average of around 80% after data of around seven characters has been recorded. We would like to emphasize that seven characters correspond to only 168 s of unsupervised training time or 476 unlabeled epochs which suffice to reliably estimate attended characters (see Fig. 3**C**). The characteristic behaviors of the three classifiers also transfer to the spelling accuracy which is depicted in Fig. 3**C**.

Having found that the MIX method is outperforming the two competing unsupervised learning methods by a large margin in the online study, the question remains how well it competes with a supervised classifier. We compared the unsupervised MIX performance with supervised shrinkage-regularized LDA classifier [58] which is a highly competitive supervised classifier in the field of BCIs [59]. As no supervised classifier was used in the online experiments, we could realize such a comparison only in a post-hoc offline re-analysis of the data. Both classifiers were trained on the first $N-1$ characters and tested on the *Nth* character. Fig. 4 shows the results.

We tested the null hypothesis that both single epoch classification accuracies come from the same distribution. The non-parametric Wilcoxon rank sum

[2] Github repository: https://github.com/DavidHuebner/Unsupervised-BCI-Matlab.

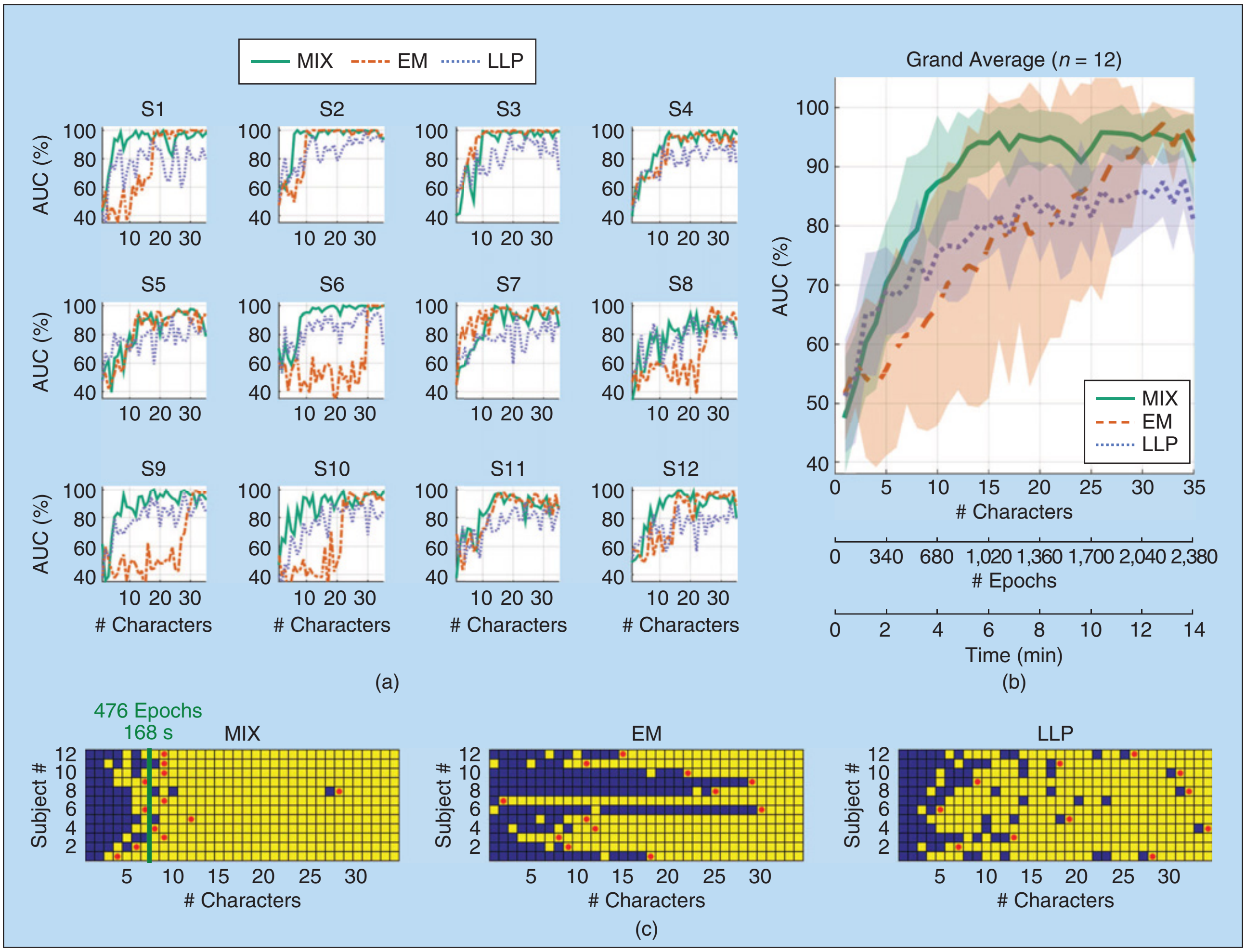


FIGURE 3 Online performance. (a) Binary target vs. non-target classification accuracies for each subject as seen online. The AUC was computed on the latest (unseen) character. (b) Grand average classification accuracy. Shaded area depicts the mean ± standard deviation across subjects. (c) Overview of correctly and incorrectly spelled characters for all 12 subjects. Blue squares denote incorrectly spelled characters while yellow squares indicate correctly spelled characters. For each subject and method, the red circles indicate the first time point of perfect control, where all post-hoc reanalyzed characters and all upcoming letters are correctly decoded. Post-hoc reanalyzed characters are obtained by reapplying the current (improved) classifier to the data from all trials up to the current one. **EM**: Expectation-Maximization, **LLP**: Learning from Label Proportions, **MIX**: Mixture of the two methods.

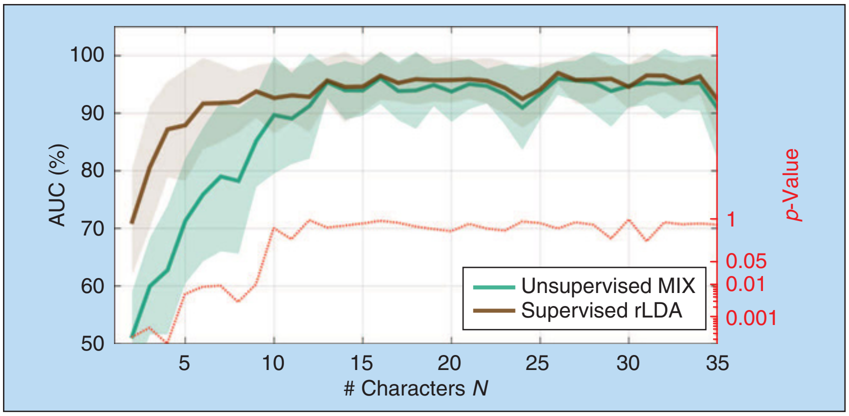


FIGURE 4 Comparison of the unsupervised MIX method with a supervised regularized LDA classifier. Both classifiers were trained on the first $N-1$ characters and tested on the Nth character. The thick lines depict the grand average over 12 subjects while the shaded area shows the standard deviation across subjects. The red dotted line shows the p-value of a Wilcoxon rank sum test comparing the supervised and unsupervised performance for character N.

test showed that significant differences exist only for the first 9 characters (for $p = 0.05$). No significant differences in the performance of the unsupervised and supervised method can be observed for 10 or more characters.

V. Discussion

This is the first online study which demonstrates that (after a short ramp-up period) an unsupervised ERP classifier for BCIs, which was trained from scratch, shows comparable performance as a state-of-the-art supervised classifier trained on the same amount of, but labeled data. This means that unlabeled data from the highly structured ERP domain is about as valuable as labeled data for the training of a classifier once a sufficient amount of data is available. We also found that the online performance observations of LLP, EM and MIX match well with the three main results reported in an earlier offline study by Verhoeven et al. [27] regarding (a) the single epoch classification, (b) spelling accuracies and (c) the observed relative performance differences between the three methods. We conclude that no severe overfitting has occurred in the offline study by Verhoeven et al. [27] and that these promising results are indeed transferable to the online case.

The unsupervised approaches have two key advantages over traditional supervised methods. First, the unproductive calibration time required for training a supervised classifier becomes superfluous with unsupervised learners. This time can be utilized on the desired application right from the start, even though the performance needs to ramp up over a few trials. Subjects perceived the ramp-up phase as challenging because of the confusing feedback. Many subjects were blaming themselves rather than the computer for the incorrect feedback. We want to stress that for some applications, e.g., text spelling, the text generated during the ramp-up period can be corrected post-hoc by simply reapplying the improved classifier at a later time point onto the initially collected data [25]. Another approach to mitigate the ramp-up effect of the MIX method is, of course, to also incorporate transfer learning. Its benefit for unsupervised learning approaches has been shown by Kindermans and colleagues [70].

The second big advantage of unsupervised classifiers is the ability to continuously learn from unlabeled data. It is well-known that EEG signals do not only change from calibration to online sessions [72], but are also non-stationary over longer sessions [72] due to changing human factors (fatigue, motivation and learning) [2], [73] or non-human factors (drying gel leading to changing impedances, changed environmental conditions). As supervised classifiers are fixed after training, non-stationarities can cause supervised methods to deteriorate over a session, while unsupervised methods have the chance to adapt to changing data distributions and maintain or even improve their classification accuracy over time [25].

The biggest limitation of the presented unsupervised learning approaches is that—so far—they are mostly restricted to ERP data and are not directly applicable to, e.g., motor imagery data. The reason is that they explicitly utilize the rich structure introduced by the ERP paradigm. For instance, the EM algorithm exploits that one latent variable—the selected symbol—uniquely determines all target and non-targets epochs of a trial. The LLP approach requires a slight modification of the stimulation paradigm in order to create different class proportions. Here, one could consider the option of switching back to a spelling matrix without visual blanks to avoid highlighting unnecessary symbols after the ramp-up phase [26]. While this might slightly change the ERP responses, the continued unsupervised adaptation should be able to adapt to these changes. Re-visiting existing ERP-based BCI applications, however, makes evident, that such class proportion differences might be available in some applications already without changing the interface, e.g., in applications which implement a two-step selection procedure where the number of symbols differs in the first and second selection step. In general, we think that future work should go towards jointly adapting the paradigm and classifier by considering the user, interface and decoder as a holistic system. A first conceptual attempt has been made by Mladenović et al. [74].

The MIX method is the result of combining two unsupervised learning ideas with complementary strengths and weaknesses [27]. By reviewing other attempts, we hope to further foster the combination of different ideas. In the past, Kindermans et al. [70] already proposed a joint Bayesian framework utilizing a language model, dynamic stopping and transfer learning. These add-on techniques can also be combined with the unsupervised MIX method if increased spelling speed is required. Certainly, this set could be further extended, for instance by exploiting error-related potentials [19]. We think that coping with the low SNR in BCI data requires the aggregation of information from different temporal and neuronal sources and a careful exploitation of the underlying data constraints.

VI. Conclusion

We reviewed different strategies to learn from unlabeled data in ERP-based BCIs. There is clear evidence that unsupervised adaptation outperforms non-adaptive supervised classifier. We also found conceptually different learning strategies based on predicted labels, additional user input such as error-related potentials or based on the exploitation of the underlying ERP data constraints. As demonstrated with the MIX method, combining these ideas can substantially improve unsupervised learning approaches. An online study with 12 healthy subjects showed that the MIX method is currently by far the most promising unsupervised learning approach which can even compete with a supervised state-of-the-art method that has the same amount of training data and full label information available after a short ramp-up. If a slight modification of the ERP paradigm is accepted by BCI users, then unsupervised learning methods can in practice completely replace supervised methods. This opens the opportunity for true plug & play systems and the ability to learn from large unlabeled data sets to find common patterns and improve transfer learning.

Acknowledgment

DH and MT thankfully acknowledge the support by BrainLinks-BrainTools Cluster of Excellence funded by the German Research Foundation (DFG), grant number EXC 1086. DH and MT further acknowledge the bwHPC initiative, grant INST 39/963-1 FUGG. PJK gratefully acknowledges funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement NO 657679. TV thankfully acknowledges financial support from the

Special Research Fund of Ghent University. KRM thanks DFG (DFG SPP 1527, MU 987/14-1) and the Federal Ministry for Education and Research (BMBF No. 2017-0-00451) as well as support by the Brain Korea 21 Plus Program by the Institute for Information & Communications Technology Promotion (IITP) grant (1IS14013A) funded by the Korean government.

References

[1] B. Blankertz, G. Curio, and K.-R. Müller, "Classifying single trial EEG: Towards brain computer interfacing," in *Proc. Advances Neural Information Processing Systems*, Vancouver, 2001, pp. 157–164.

[2] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, June 2002.

[3] G. Dornhege, J. d R. Millán, T. Hinterberger, D. McFarland, and K.-R. Müller, *Towards Brain–Computer Interfacing*. Cambridge, MA: MIT Press, 2007.

[4] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Müller, "Introduction to machine learning for brain imaging," *NeuroImage*, vol. 56, no. 2, pp. 387–399, May 2011.

[5] J. Wolpaw and E. W. Wolpaw, *Brain–Computer Interfaces: Principles and Practice*. New York: Oxford University Press, 2012.

[6] J. I. Münßinger, S. Halder, S. C. Kleih, A. Furdea, V. Raco, A. Hösle, and A. Kubler, "Brain painting: First evaluation of a new brain–computer interface application with ALS-patients and healthy volunteers," *Frontiers Neurosci.*, vol. 4, no. 182, pp. 1–11, Nov. 2010.

[7] E. W. Sellers and E. Donchin, "A P300–based brain–computer interface: Initial tests by ALS patients," *Clin. Neurophysiol.,* vol. 117, no. 3, pp. 538–548, Mar. 2006.

[8] C. Zickler, S. Halder, S. C. Kleih, C. Herbert, and A. Kübler, "Brain painting: Usability testing according to the user-centered design in end users with severe motor paralysis," *Artif. Intell. Med.*, vol. 59, no. 2, pp. 99–110, Oct. 2013.

[9] L. A. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr. Clin. Neurophysiol.*, vol. 70, no. 6, pp. 510–523, Dec. 1988.

[10] R. Leeb, D. Friedman, G. R. Müller-Putz, R. Scherer, M. Slater, and G. Pfurtscheller, "Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic," *Comput. Intell. Neurosci.*, vol. 2007, no. 79642, pp. 1–8, July 2007.

[11] R. Fazel-Rezai, B. Z. Allison, C. Guger, E. W. Sellers, S. C. Kleih, and A. Kübler, "P300 brain computer interface: Current challenges and emerging trends," *Frontiers Neuroeng.*, vol. 5, no. 14, pp. 1–14, July 2012.

[12] A. Kübler, N. Neumann, J. Kaiser, B. Kotchoubey, T. Hinterberger, and N. P. Birbaumer, "Brain–computer communication: Self-regulation of slow cortical potentials for verbal communication," *Arch. Phys. Med. Rehab.*, vol. 82, no. 11, pp. 1533–1539, Nov. 2001.

[13] M. Krauledat, M. Tangermann, B. Blankertz, and K.-R. Müller, "Towards zero training for brain–computer interfacing," *PLOS One*, vol. 3, no. 8, p. e2967, Aug. 2008.

[14] S. Fazli, S. Dähne, W. Samek, F. Bießmann, and K.-R. Müller, "Learning from more than one data source: Data fusion techniques for sensorimotor rhythm-based brain-computer interfaces," *Proc. IEEE*, vol. 103, no. 6, pp. 891–906, May 2015.

[15] S. Fazli, F. Popescu, M. Danóczy, B. Blankertz, K.-R. Müller, and C. Grozea, "Subject-independent mental state classification in single trials," *Neural Netw. Official J. Int. Neural Netw. Soc.*, vol. 22, no. 9, pp. 1305–1312, Nov. 2009.

[16] V. Jayaram, M. Alamgir, Y. Altun, B. Schölkopf, and M. Grosse-Wentrup, "Transfer learning in brain–computer interfaces," *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 20–31, Jan. 2016.

[17] S. Lu, C. Guan, and H. Zhang, "Unsupervised brain computer interface based on intersubject information and online adaptation," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 17, no. 2, pp. 135–145, Feb. 2009.

[18] R. C. Panicker, S. Puthusserypady, and Y. Sun, "Adaptation in P300 brain–computer interfaces: A two-classifier cotraining approach," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 12, pp. 2927–2935, July 2010.

[19] T. Zeyl, E. Yin, M. Keightley, and T. Chau, "Partially supervised P300 speller adaptation for eventual stimulus timing optimization: Target confidence is superior to error-related potential score as an uncertain label," *J. Neural Eng.*, vol. 13, no. 2, pp. 026008, Feb. 2016.

[20] C. Vidaurre, M. Kawanabe, P. von Bünau, B. Blankertz, and K.-R. Müller, "Toward unsupervised adaptation of LDA for brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 3, pp. 587–597, Nov. 2010.

[21] S. Dähne, J. Höhne, and M. Tangermann, "Adaptive classification improves control performance in ERP-based BCIs," in *Proc. 5th Int. Brain–Computer Interface Conf.*, Graz, 2011, pp. 92–95.

[22] S. N. G. Bolagh, M. B. Shamsollahi, C. Jutten, and M. Congedo, "Unsupervised cross-subject BCI learning and classification using Riemannian geometry," in *Proc. 24th European Symp. Artificial Neural Networks Computational Intelligence Machine Learning*, Bruges, 2016.

[23] E. Daucé, T. Proix, and L. Ralaivola, "Reward-based online learning in non-stationary environments: Adapting a P300-speller with a 'backspace' key," in *Proc. Int. Joint Conf. Neural Networks*, Killarney, 2015, pp. 1–8.

[24] P.-J. Kindermans, D. Verstraeten, and B. Schrauwen, "A Bayesian model for exploiting application constraints to enable unsupervised training of a P300-based BCI," *PLOS One*, vol. 7, no. 4, pp. e33758, Apr. 2012.

[25] P.-J. Kindermans, M. Schreuder, B. Schrauwen, K.-R. Müller, and M. Tangermann, "True zero-training brain–computer interfacing: An online study," *PLOS One*, vol. 9, no. 7, pp. e102504, July 2014.

[26] D. Hübner, T. Verhoeven, K. Schmid, K.-R. Müller, M. Tangermann, and P.-J. Kindermans, "Learning from label proportions in brain–computer interfaces: Online unsupervised learning with guarantees," *PLOS One*, vol. 12, no. 4, p. e0175856, Apr. 2017.

[27] T. Verhoeven, D. Hübner, M. Tangermann, K.-R. Müller, J. Dambre, and P.-J. Kindermans, "Improving zero-training brain–computer interfaces by mixing model estimators," *J. Neural Eng.*, vol. 14, no. 3, p. 036021, Apr. 2017.

[28] J. Grizou, I. Iturrate, L. Montesano, P.-Y. Oudeyer, and M. Lopes, "Interactive learning from unlabeled instructions," in *Proc. 13th Conf. Uncertainty Artificial Intelligence*, Québec, 2014, pp. 1–8.

[29] J. Grizou, I. Iturrate, L. Montesano, P.-Y. Oudeyer, and M. Lopes, "Calibration-free BCI based control," in *Proc. 28th Association Advancement Artificial Intelligence Conf.*, Quebec, 2014, pp. 1213–1220.

[30] I. Iturrate, J. Grizou, J. Omedes, P.-Y. Oudeyer, M. Lopes, and L. Montesano, "Exploiting task constraints for self-calibrated brain–machine interface control using error-related potentials," *PLOS One*, vol. 10, no. 7, pp. e0131491, July 2015.

[31] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, Jan. 2012.

[32] J. d R. Millán, R. Rupp, G. R. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kübler, R. Leeb, et al. "Combining brain–computer interfaces and assistive technologies: State-of-the-art and challenges," *Frontiers Neurosci.*, vol. 4, no. 161, pp. 1–15, Mar. 2010.

[33] R. Chavarriaga, P. W. Ferrez, and J. d R. Millán, "To err is human: Learning from error potentials in brain–computer interfaces," in *Proc. Int. Conf. Cognitive Neurodynamics,* Shanghai, 2008, pp. 777–782.

[34] I. Iturrate, L. Montesano, and J. Minguez, "Robot reinforcement learning using EEG-based reward signals," in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, 2010, pp. 4822–4829.

[35] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. d R. Millán, "Teaching brain–machine interfaces as an alternative paradigm to neuroprosthetics control," *Sci. Rep.*, vol. 5, no. 13893, pp. 1–10, Sept. 2015.

[36] T. O. Zander, L. R. Krol, N. P. Birbaumer, and K. Gramann, "Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 113, no. 52, pp. 14898–14903, 2016.

[37] R. Chavarriaga, I. Iturrate, and J. D. R. Millán, "Robust, accurate spelling based on error-related potentials," in *Proc. 6th Int. Brain–--Computer Interface Meeting*, Graz, 2016, p. 15.

[38] A. F. Salazar-Gomez, J. DelPreto, S. Gil, F. H. Guenther, and D. Rus, "Correcting robot mistakes in real time using EEG signals," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2017.

[39] M. Schreuder, B. Blankertz, and M. Tangermann, "A new auditory multi-class brain–computer interface paradigm: Spatial hearing as an informative cue," *PLOS One*, vol. 5, no. 4, pp. e9813, Apr. 2010.

[40] B. Blankertz, M. Tangermann, C. Vidaurre, S. Fazli, C. Sannelli, S. Haufe, C. Maeder, L. Ramsey, I. Sturm, G. Curio, and K.-R. Müller, "The Berlin brain–computer interface: Non-medical uses of BCI technology," *Frontiers Neurosci.*, vol. 4, p. 198, Dec. 2010.

[41] M. Bensch, A. A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, M. Bogdan, W. Rosenstiel, and N. Birbaumer, "Nessi: An EEG-controlled web browser for severlely paralyzed patients," *Comput. Intell. Neurosci.*, vol. 2007, p. e71863, Sept. 2007.

[42] A. Nijholt, D. P.-O. Bos, and B. Reuderink, "Turning shortcomings into challenges: Brain–computer interfaces for games," *Entertainment Comput.*, vol. 1, no. 2, pp. 85–94, Apr. 2009.

[43] M. Ahn, M. Lee, J. Choi, and S. C. Jun, "A review of brain–computer interface games and an opinion survey from researchers, developers and users," *Sensors*, vol. 14, no. 8, pp. 14601–14633, Aug. 2014.

[44] M. Tangermann, M. Schreuder, S. Dähne, J. Höhne, S. Regler, A. Ramsay, M. Quek, J. Williamson, and R. Murray-Smith, "Optimized stimulation events for a visual ERP BCI," *Int. J. Bioelectromagnetics*, vol. 13, no. 3, pp. 119–120, Nov. 2011.

[45] B. Blankertz, L. Acqualagna, S. Dähne, S. Haufe, M. Schultze-Kraft, I. Sturm, M. Ušćumlić, M. Wenzel, G. Curio, and K.-R. Müller, "The Berlin brain–computer interface: Progress beyond communication and control," *Frontiers Neurosci.*, vol. 10, no. 530, pp. 1–24, Nov. 2016.

[46] J. D. Bayliss, "Use of the evoked potential P3 component for control in a virtual apartment," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 11, no. 2, pp. 113–116, June 2003.

[47] C. Groenegress, C. Holzner, C. Guger, and M. Slater, "Effects of P300-based BCI use on reported presence in a virtual environment," *Presence Teleoperators Virtual Environ.*, vol. 19, no. 1, pp. 1–11, Feb. 2010.

[48] C. Guger, S. Daban, E. Sellers, C. Holzner, G. Krausz, R. Carabalona, F. Gramatica, and G. Edlinger, "How many people are able to control a P300-based brain–computer interface (BCI)?" *Neurosci. Lett.*, vol. 462, no. 1, pp. 94–98, Sept. 2009.

[49] S. Gao, Y. Wang, X. Gao, and B. Hong, "Visual and auditory brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1436–1447, May 2014.

[50] G. Bin, X. Gao, Y. Wang, Y. Li, B. Hong, and S. Gao, "A high-speed BCI based on code modulation VEP," *J. Neural Eng.*, vol. 8, no. 2, pp. 025015, Mar. 2011.

[51] J. Thielen, P. V. D. Broek, J. Farquhar, and P. Desain, "Broad-band visually evoked potentials: Re(con)volution in brain–computer interfacing," *PLOS One*, vol. 10, no. 7, p. e0133797, July 2015.

[52] X. Chen, Y. Wang, M. Nakanishi, X. Gao, T.-P. Jung, and S. Gao, "High-speed spelling with a noninvasive brain–computer interface," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 112, no. 44, pp. E6058–E6067, Sept. 2015.

[53] F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proc. IEEE*, vol. 103, no. 6, pp. 871–890, June 2015.

[54] P.-J. Kindermans, D. Verstraeten, P. Buteneers, and B. Schrauwen, "How do you like your P300 speller: Adaptive, accurate and simple?" in *Proc. 5th Int. Brain–Computer Interface Conf.*, Graz, 2011, pp. 96–99.

[55] A. Barachant and M. Congedo, "A plug&play P300 BCI using information geometry," arXiv Preprint, arXiv:1409.0107, Aug. 2014.

[56] L. I. Kuncheva, C. J. Whitaker, and A. Narasimhamurthy, "A case-study on naïve labelling for the nearest mean and the linear discriminant classifiers," *Pattern Recognit.*, vol. 41, no. 10, pp. 3010–3020, Oct. 2008.

[57] B. Blankertz, K.-R. Müller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. R. Millan, M. Schroder, and N. Birbaumer, "The BCI competition III: Validating alternative approaches to actual BCI problems," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 14, no. 2, pp. 153–159, June 2006.

[58] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components, a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, May 2011.

[59] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 4, no. 2, pp. R1–R13, Jan. 2007.

[60] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 128. New York: Springer, 2006.

[61] P. W. Ferrez and J. D. R. Millán, "Error-related EEG potentials generated during simulated brain–computer interaction," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 3, pp. 923–929, Feb. 2008.

[62] A. Llera, M. A. van Gerven, V. Gómez, O. Jensen, and H. J. Kappen, "On the use of interaction error potentials for adaptive brain computer interfaces," *Neural Netw.*, vol. 24, no. 10, pp. 1120–1127, Dec. 2011.

[63] R. Chavarriaga, A. Sobolewski, and J. D. R. Millán, "Errare machinale est: The use of error-related potentials in brain–machine interfaces," *Frontiers Neurosci.*, vol. 8, no. 208, pp. 1–13, July 2014.

[64] P. Margaux, M. Emmanuel, D. Sébastien, B. Olivier, and M. Jérémie, "Objective and subjective evaluation of online error correction during P300-based spelling," *Adv. Hum.–Comput. Interact.*, vol. 2012, no. 4, pp. 1–13, Jan. 2012.

[65] M. Spüler, W. Rosenstiel, and M. Bogdan, "Online adaptation of a c-VEP brain–computer interface (BCI) based on error-related potentials and unsupervised learning," *PLOS One*, vol. 7, no. 12, p. e51077, Dec. 2012.

[66] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multi-class brain computer interface classification by Riemannian geometry," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 920–928, Apr. 2012.

[67] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Mag.*, vol. 25, no. 1, pp. 41–56, Jan. 2008.

[68] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le, "Estimating labels from label proportions," *J. Mach. Learn. Res.*, vol. 10, pp. 2349–2374, Oct. 2009.

[69] T. Verhoeven, P. Buteneers, J. Wiersema, J. Dambre, and P.-J. Kindermans, "Towards a symbiotic brain–computer interface: Exploring the application–decoder interaction," *J. Neural Eng.*, vol. 12, no. 6, pp. 066027, Nov. 2015.

[70] P.-J. Kindermans, M. Tangermann, K.-R. Müller, and B. Schrauwen, "Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller," *J. Neural Eng.*, vol. 11, no. 3, pp. 035005, May 2014.

[71] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivariate Anal.*, vol. 88, no. 2, pp. 365–411, Feb. 2004.

[72] P. Shenoy, M. Krauledat, B. Blankertz, R. P. Rao, and K.-R. Müller, "Towards adaptive classification for BCI," *J. Neural Eng.*, vol. 3, no. 1, pp. R13–R23, Mar. 2006.

[73] C. Jeunet, B. N'Kaoua, and F. Lotte, "Advances in user-training for mental-imagery-based BCI control: Psychological and cognitive factors and their neural correlates," *Progress Brain Res.*, vol. 228, pp. 3–35, June 2016.

[74] J. Mladenović, J. Mattout, and F. Lotte, "A generic framework for adaptive EEG-based BCI training and operation," in *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. Boca Raton, FL: CRC Press, 2017, ch. 33, pp. 595–612.

Call for Papers for Journal Special Issues

Special Issue on "Computational Intelligence for Smart Energy Applications to Smart Cities"

Journal: ***IEEE Transactions on Emerging Topics in Computational Intelligence***
Guest Editors: Wei-Yu Chiu, Hongjian Sun, Chao Wang, and Athanasios V. Vasilakos
Submission Deadline: May 15, 2018
Further Information: Wei-Yu Chiu (wychiu@ee.nthu.edu.tw)
http://cis.ieee.org/images/files/Documents/Transactions/TETCI/SI10_CFP_CISEASC.pdf

Special Issue on "New Advances in Deep-Transfer Learning"

Journal: ***IEEE Transactions on Emerging Topics in Computational Intelligence***
Guest Editors: Zhaohong Deng, Jie Lu, Dongrui Wu, Kup-Sze Choi, Shiliang Sun, and Yusuke Nojima
Submission Deadline: June 30, 2018
Further Information: Zhaohong Deng (dengzhaohong@jiangnan.edu.cn)
http://cis.ieee.org/images/files/Documents/Transactions/TETCI/SI11_CFP_DTL.pdf

Special Issue on "Theoretical Foundations of Evolutionary Computation"

Journal: ***IEEE Transactions on Evolutionary Computation***
Guest Editors: Pietro S. Oliveto, Anne Auger, Francisco Chicano, and Carlos M. Fonseca
Submission Deadline: October 1, 2018
Further Information: Pietro S. Oliveto (p.oliveto@sheffield.ac.uk)
http://cis.ieee.org/images/files/Documents/Transactions/TEC/CFP_TEVC_TFoEC_FINAL.pdf

Special Issue on "Deep Reinforcement Learning and Games"

Journal: ***IEEE Computational Intelligence Magazine***
Guest Editors: Dongbin Zhao, Simon Lucas, and Julian Togelius
Submission Deadline: October 1, 2018
Further Information: Dongbin Zhao (Dongbin.zhao@ia.ac.cn)
http://people.ucas.ac.cn/~0001598?language=en

Digital Object Identifier 10.1109/MCI.2018.2807159

Conference Calendar

Bernadette Bouchon-Meunier
University Pierre et Marie Curie, FRANCE

* Denotes a CIS-Sponsored Conference
∆ Denotes a CIS Technical Co-Sponsored Conference

IEEE World Congress on Computational Intelligence
08-13 July 2018, Rio de Janeiro, Brazil
www.ieee-wcci.org

*** 2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (IEEE CIBCB 2018)**
May 30–June 2, 2018
Place: Saint Louis, USA
General Chair: Donald Wunsch
Website: http://www.cibcb.org/CIBCB2018/index.html

*** 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (IEEE CIVEMSA 2018)**
June 12–14, 2018
Place: Ottawa, Canada
General Chairs: Ana-Maria Cretu, Dalila Megherbi
Website: http://civemsa2018.ieee-ims.org

*** 2018 IEEE World Congress on Computational Intelligence (IEEE WCCI 2018)**
July 8–13, 2018
Place: Rio de Janeiro, Brazil
General Co-Chairs: Marley Vellasco, Pablo Estevez
Website: http://www.ieee-wcci.org/

*** 2018 IEEE Conference on Computational Intelligence and Games (IEEE CIG 2018)**
August 14–17, 2018
Place: Maastricht, The Netherlands
General Chair: Mark Winands
Website: https://project.dke.maastrichtuniversity.nl/cig2018/

∆ 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP 2018)
September 6–7, 2018
Place: Zaragoza, Spain
General Chairs: Sergio Ilarri, Fernando Bobillo, Raquel Trillo-Lado, Martín López-Nores
Website: http://smap2018.unizar.es

*** 2018 IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA 2018)**
October 1–4, 2018
Place: Turin, Italy
General Chairs: Francesco Bonchi, Foster Provost
Website: https://dsaa2018.isi.it

*** 2018 IEEE Smart World Congress (IEEE SWC 2018)**
October 8–12, 2018
Place: Guangzhou, China
General Chairs: Guojun Wang, Yew Soon Ong
Website: http://www.smart-world.org/2018/

*** 2018 IEEE Latin American Conference on Computational Intelligence (2018 IEEE LA-CCI)**
November 7–9, 2018
Place: Guadalajara, Mexico
General Chairs: Alma Y. Alanis, Marco A. Perez-Cisneros
Website: http://la-cci.org

*** 2018 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2018)**
November 18–21, 2018
Place: Bangalore, India
General Co-Chairs: Sundaram Suresh, Koshy George
Website: http://ieee-ssci2018.org

*** 2019 IEEE Congress on Evolutionary Computation (IEEE CEC 2019)**
June 10–13, 2019
Place: Wellington, New Zealand
General Co-Chairs: Mengjie Zhang, Kay Chen Tan
Website: http://www.cec2019.org

*** 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2019)**
June 23–26, 2019
Place: New Orleans, USA
General Chairs: Timothy C. Havens, James M. Keller
Website: http://www.fuzzieee.org

*** 2019 IEEE International Conference on Games (IEEE CoG 2019)**
August 20–23, 2019
Place: London, United Kingdom
General Chairs: Diego Perez Liebana and Sanaz Mostaghim
Website: TBA

*** 2019 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2019)**
December 6–9, 2019
Place: Xiamen, China
General Chair: Tingwen Huang
Website: TBA

*** 2020 IEEE World Congress on Computational Intelligence (IEEE WCCI 2020)**
July 19–24, 2020
Place: Glasgow, UK
General Chairs: Amir Hussain, Marios M. Polycarpou, Xin Yao
Website: TBA

Digital Object Identifier 10.1109/MCI.2018.2807041
Date of publication: 10 April 2018

2018 IEEE Smart World Congress

October 8-12, 2018, Guangzhou, China

Organizing Committee

General Chairs
Guojun Wang, *Guangzhou Univ., China*
Yew Soon Ong, *Nanyang Tech. Univ., Singapore*

Program Chairs
Md Zakirul Alam Bhuiyan, *Fordham Univ., USA*
Ruixuan Li, *Huazhong Univ. of Sci. and Tech., China*

Workshop Chairs
Manuel Roveri, *Politecnico di Milano, Italy*
Seiichi Ozawa, *Kobe University, Japan*
Qin Liu, *Hunan University, China*

Tutorial Chair
Haibo He, *University of Rhode Island, USA*

Summit Chair
Zhong Chen, *Peking University, China*

Competition Chair
Jerry Gao, *San Jose State University, USA*

Local Arrangement Chair
Jianer Chen, *Guangzhou University, China*

Honorary Chairs
Stephen S. Yau, *Arizona State Univ., USA*
Xin Yao, *Southern Univ. of Sci. and Tech., China*
Cesare Alippi, *Politecnico di Milano, Italy*

Advisory Chairs
Vincenzo Piuri, *Univ. of Milan, Italy*
Hussein Abbass, *Univ. of New South Wales, Australia*
Hong Mei, *Beijing Institute of Technology, China*

Steering Chairs
Jianhua Ma, *Hosei Univ., Japan*
Laurence T. Yang, *St. Francis Xavier Univ., Canada*

IMPORTANT DATES

Workshop Proposal:	**Mar. 8, 2018**
Paper Submission Deadline:	**Apr. 8, 2018**
Authors Notification:	**June 25, 2018**
Camera-ready Submission:	**Aug. 8, 2018**

Sponsored by

IEEE Computational Intelligence Society

IEEE

Ubiquitous Intelligence for Smart World

The smart world is set to enhance everyday things with abilities of sensation, communication, computation and intelligence so that many tasks and processes could be simplified, more efficient and productive. It consists of numerous “smart things” that can be endowed with different levels/forms of computational intelligence and even capable of thinking. Research on smart world is an emerging research field covering many areas, essential problems and crucial issues in truly building the smart world that benefits humanity, and simultaneously safeguards the natural environment for sustainable development and evolution.

The IEEE Smart World Congress originated from the 2005 Workshop on Ubiquitous Smart Worlds (USW, Taipei) and the 2005 Symposium on Ubiquitous Intelligence and Smart World (UISW, Nagasaki). SmartWorld 2018 in Guangzhou is the continuation after the success of SmartWorld 2017 in San Francisco, SmartWorld 2016 in Toulouse, and SmartWorld 2015 in Beijing. IEEE SmartWorld 2018 aims to provide a high-profile, leading-edge platform for researchers and engineers to exchange and explore state-of-art advances and innovations in graceful integrations of Cyber, Physical and Social Worlds with Ubiquitous Intelligence.

IEEE SmartWorld 2018 Topics

- ✧ Smart Industry and Manufacture
- ✧ Smart Agriculture and Aquaculture
- ✧ Smart Materials and Fabric
- ✧ Smart Environment and Ecosystems
- ✧ Smart Earth/Space System
- ✧ Smart Grid and Energy
- ✧ Smart Logistics and Retail
- ✧ Smart Building and Structure
- ✧ Smart Roads and Transportation
- ✧ Smart Vehicles and Networks
- ✧ Smart Machines and Robots
- ✧ Smart Home and Furniture
- ✧ Smart Appliances and Goods
- ✧ Smart Wearables and Implants
- ✧ Smart Medicine and Healthcare
- ✧ Smart Elderly/Kiddy Care
- ✧ Smart Foods and Living
- ✧ Smart Learning and Education
- ✧ Smart Disaster Management
- ✧ Smart Internet of Things
- ✧ Smart Sensing, System and Service
- ✧ Smart Computing & Communication

Six Co-located Conferences

- ◆ The 15th IEEE Int'l Conf. on Ubiquitous Intelligence and Computing (**UIC** 2018)
- ◆ The 15th IEEE Int'l Conf. on Advanced and Trusted Computing (**ATC** 2018)
- ◆ The 18th IEEE Int'l Conf. on Scalable Computing & Communications (**ScalCom** 2018)
- ◆ The 4th IEEE Int'l Conf. on Cloud and Big Data Computing (**CBDCom** 2018)
- ◆ The 4th IEEE Int'l Conf. on Internet of People (**IoP** 2018)
- ◆ The 2nd IEEE Int’l Conf. on Smart City Innovations (**SCI** 2018)

IEEE SmartWorld 2018 Calls for

- ✓ Papers that must be submitted electronically via the 2018 congress website
- ✓ Workshop and Special Session Proposals in smart world related all aspects
- ✓ Tutorial Proposals in addressing the emerging smart world areas
- ✓ Smart AI City Competition Proposals for AI technological applications

Accepted conference papers will be published by IEEE (IEEE-DL and EI indexed) in Conference Proceedings. Best Paper Awards will be presented to high quality papers. Selected papers will be recommended to prestige journal special issues.

IEEE SmartWorld 2018 Website: http://www.smart-world.org/2018/

Digital Object Identifier 10.1109/MCI.2018.2807082

Advertisers Index

IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE REPRESENTATIVE

Mark David
Director, Business Development — Media & Advertising
Phone: +1 732 465 6473
Fax: +1 732 981 1855
m.david@ieee.org

445 Hoes Lane, Piscataway, NJ 08854

Digital Object Identifier 10.1109/MCI.2017.2770617

IEEE

IEEE Computational Intelligence Society

2019 IEEE International Conference on Fuzzy Systems

FUZZ-IEEE 2019

June 23-26, 2019

General Chairs
Tim Havens, USA
Jim Keller, USA

Program Chairs
Alina Zare, USA
Derek Anderson, USA

Special Sessions Chairs
Christian Wagner, UK
Thomas Runkler, Germany
Tutorials Chairs
Qiang Shen, UK
Jesus Chamorro, Spain
Keynotes Chair
Robert John, UK
Posters Chair
Marek Reformat, Canada
Finance Chair
Mihail Popescu, USA
Conflict-of-Interest Chairs
Sansanee Auephanwiriyakul, Thailand
CT Lin, Taiwan
Competitions Chairs
Christophe Marsala, France
Mika Sato-Ilic, Japan
Panel Sessions Chair
Humberto Bustince, Spain
Publications Chairs
Anna Wilbi, Netherlands
Tim Wilkin, Australia
Registrations Chair
Marie-Jeanne Lesot, France
Local Arrangement Chairs
Fred Petry, USA
Paul Elmore, USA
Publicity Chair
Daniel Sanchez, Spain
Web Chair
Tony Pinar, USA

The 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2019), the world-leading event focusing on the theory and application of fuzzy logic, will be held in New Orleans, Louisiana, USA. Nicknamed the "Big Easy," New Orleans is known for its round-the-clock nightlife, vibrant live-music scene, and spicy Cajun cuisine. It is located on the Mississippi River, near the Gulf of Mexico, and is popular tourist destination for all ages.

FUZZ-IEEE 2019 will be hosted at the JW Marriott, a premier conference venue nestled in the heart of the world-famous French Quarter. You will be steps away from some of New Orleans's most iconic nightlife and restaurants. Take a walk outside and visit Jackson Square, shop at the lively French Market, or dance your way through Bourbon Street.

FUZZ-IEEE 2019 will represent a unique meeting point for scientists and engineers, from academia and industry, to interact and discuss the latest enhancements and innovations in the field. The topics of the conference will cover all aspects of theory and applications of fuzzy logic and its hybridisations with other artificial and computational intelligence methods. In particular, FUZZ-IEEE 2019 topics include, but are not limited to:

Mathematical and theoretical foundations of fuzzy sets, fuzzy measures, and fuzzy integrals
Fuzzy control, robotics, sensors, fuzzy hardware and architectures
Fuzzy data analysis, fuzzy clustering, classification and pattern recognition
Type-2 fuzzy sets, computing with words, and granular computing
Fuzzy systems with big data and cloud computing, fuzzy analytics, and visualization
Fuzzy systems design and optimization
Fuzzy decision analysis, multi-criteria decision making and decision support
Fuzzy logical and its applications to industrial engineering
Fuzzy modeling, identification, and fault detection
Fuzzy information processing, information extraction, and fusion
Fuzzy web engineering, information retrieval, text mining, and social network analysis
Fuzzy image, speech and signal processing, vision, and multimedia data analysis
Fuzzy databases and informational retrieval
Rough sets, imprecise probabilities, possibilities approaches
Industrial, financial, and medical applications
Fuzzy logic application in civil engineering and GIS
Fuzzy sets and soft computing in social sciences
Liguistic summarization, natural language processing
Computational intelligence in security
Hardware and software for fuzzy systems and logic
Fuzzy Markup Language and standard technologies for fuzzy systems
Adaptive, hierarchical, and hybrid neuro- and evolutionary-fuzzy systems

The conference will include regular oral and poster presentations, an elevator pitch competition, tutorials, panels, special sessions, and keynote presentations. Full details of the submission process for papers, tutorials, and panels will be made available on the conference website: http://www.fuzzieee.org

Important dates
Deadline for special session, tutorial, competition, and panel session proposals: October 8, 2018
Notification of acceptance for tutorials, special sessions, and panels: November 2, 2018
Deadline for full paper submission: January 11, 2019
Notification of paper acceptance: March 4, 2019
Deadline for camera-ready paper submission: April 1, 2019
Deadline for early registration: April 5, 2019
Conference: June 23-26, 2019

http://www.fuzzieee.org

FUZZ-IEEE NEW ORLEANS 2019

Digital Object Identifier 10.1109/MCI.2018.2807079

IEEE Congress on Evolutionary Computation

10 - 13 June 2019, Wellington, New Zealand

IEEE Computational Intelligence Society

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Wellington
Absolutely Positively

Business
Events
Wellington
WellingtonNZ.com

Advisory Board
Hussein Abbass, Australia
Kalyanmoy Deb, USA
David Fogel, USA
Zbigniew Michalewicz, Australia
Kwong Tak Wu Sam, Hong Kong
Xin Yao, UK
Gary Yen, USA
Simon Lucas, UK

General Co-Chairs
Mengjie Zhang, New Zealand
Kay Chen Tan, Hong Kong

Program Chair
Carlos A. Coello Coello, Mexico

Technical Co-Chairs
Oscar Cordón, Spain
Hisao Ishibuchi, Japan
Jürgen Branke, UK
Jing Liu, China
Gabriela Ochoa, UK
Dipti Srinivasan, Singapore

Plenary Talk Chair
Yaochu Jin, UK

Special Session Chair
Chuan-Kang Ting, Taiwan

Tutorial Chair
Xiaodong Li, Australia

Competition Chair
JiaLin Liu, UK

Publicity Co-Chair
Stefano Cagnoni, Italy
Huanhuan Chen, China
Anna I Esparcia-Alcazar, Spain
Emma Hart, UK
Bin Hu, Austria
Sanza Mostaghim, Germany
Yew Soon Ong, Singapore
Yuhui Shi, China
Jun Zhang, China

Sponsorship Chair
Andy Song, Australia

Finance Chair
Bing Xue, New Zealand

Local Organising Chairs
Will Browne, New Zealand
Hui Ma, New Zealand

Registration Chair
Aaron Chen, New Zealand

Proceedings Chair
Yi Mei, New Zealand

Poster Chair
Kai Qin, Australia

The IEEE Congress on Evolutionary Computation (IEEE CEC) is a world-class event in the field of evolutionary computation. It provides a forum to bring together researchers and practitioners from all over the world to present and discuss their research findings on evolutionary computation and computational intelligence.

IEEE CEC 2019 will be held in Wellington, New Zealand. Wellington is known as the 'Coolest Little Capital'. It is famous for a vibrant creative culture fueled by events and great food. Wellington offers a wide range of cosmopolitan amenities in downtown that is safe, clean, and pedestrian friendly.

Call for Papers

Papers for IEEE CEC 2019 should be submitted electronically through the Congress website at www.cec2019.org, and will be refereed by experts in the fields and ranked based on the criteria of originality, significance, quality and clarity.

Call for Special Sessions

Special session proposals are invited to CEC 2019. All special session proposals should include the title, aim and scope, a short biography of all organizers, and a list of potential contributors. Proposals should be submitted to the Special Session Chair Prof Chuan-Kang Ting (ckting@cs.ccu.edu.tw).

Call for Tutorials

CEC 2019 solicits proposal for tutorials covering specific topics in Evolutionary Computation. If you are interested in proposing a tutorial, would like to recommend someone who might be interested, or have questions about tutorials, please contact the Tutorial Chair Prof Xiaodong Li (xiaodong.li@rmit.edu.au).

Call for Competitions

Competitions will be held as part of the Congress. Prospective competition organizers are invited to submit their proposals to the Competition Chair Dr JiaLin Liu (jialin.liu@qmul.ac.uk).

Call for Workshops

Workshops will be held to provide participants with the opportunity to present and discuss novel research ideas on active and emerging topics in Evolutionary Computation. Prospective workshop organizers are invited to submit their proposals to the Workshop Chair.

Important Dates

- **Special Session Proposal Deadline: 26 Oct, 2018**
- **Competition Proposal Deadline: 26 Nov, 2018**
- **Paper Submission Deadline: 7 Jan, 2019**
- **Workshop Proposal Deadline: 7 Jan, 2019**
- **Tutorial Proposal Deadline: 7 Jan, 2019**
- **Notification Deadline: 7 Mar, 2019**
- **Final Paper Submission & Early Registration Deadline: 31 Mar, 2019**

Find Us At www.cec2019.org contact: cec2019@ecs.vuw.ac.nz

Digital Object Identifier 10.1109/MCI.2018.2807080