Data Mining by Symbolic Fuzzy Classifiers and Genetic Programming– State of the Art and Prospective Approaches

SUHAIL OWAIS[1], *PAVELKRÖMER*[2], *JAN PLATOŠ*[2], *VÁCLAV SNÁŠEL*[2],*IVAN ZELINKA*[2]
[1]Department of Computer Science
Applied Science University
P.O.BOX 166 Amman 11931
JORDAN
dr_suhail@asu.edu.jo

[2]Department of Computer Science,
FEECS, VŠB – Technical University of Ostrava,
17.Listopadu 12, Ostrava
CZECH REPUBLIC
pavel.kromer@vsb.cz, jan.platos@vsb.cz, vaclav.snasel@vsb.cz, ivan.zelinka@vsb.cz

*Abstract***: -** There are various techniques for data mining and data analysis. Data mining is very important in the information retrieval areas especially when the data amounts are very large. Among them, hybrid approaches combining two or more algorithms gain importance as the complexity and dimension of real world data sets grows. In this paper, we present an application of evolutionary-fuzzy classification technique for data mining, outline state of the art of related methods and draw future directions of the research. In the presented application, genetic programming was deployed to evolve a fuzzy classifier and an example of real world application was presented.

Key-Words :-Data mining, fuzzy classifiers, genetic programming, application.

1. Introduction

The recent time has seen a rise in the demand for advanced data mining algorithms. In the real world domains; many applications generates a huge amounts of data. In such data, the hidden information can be extracted and can help in optimization of processes, designs, and algorithms.

The growing dimension and complexity of said data sets represents a challenge for traditional search and optimization methods while the increase of power of widely available computers encourages the deployment of soft computing methods such as the populational meta-heuristic algorithms, artificial neural networks and fuzzy systems. Moreover, soft computing concepts including fuzzy sets allow better modeling of real world problems and more accurate soft decisions. For example, soft computing was used to detect lifetime building thermal insulation failures [1], neural networks were deployed to visualize network traffic data for intrusion detection [2], and soft computing methods were utilized to identify typical meteorological days[3].

Fuzzy sets and fuzzy logic provide means for soft classification of data. In contrast to crisp classification, which states crisp decisions about data samples, fuzzy classification allows to analyze the data samples in a more sensitive way[4]. Fuzzy decision trees and if-then rules are examples of efficient, transparent, and easily interpretable fuzzy classifiers[4], [5].

Genetic programming is a powerful machine learning technique from the wide family of evolutionary algorithms. In contrast to the traditional evolutionary algorithms, genetic programming can be used to evolve complex hierarchical tree structures and symbolic expressions.

In this work, we used genetic programming for data mining by fuzzy classifier evolution. In particular, genetic programming was used to evolve symbolic fuzzy classifiers that are able to describe classes in a data set by means of its features. Such a fuzzy classifier evolved over a training data set can be later used for efficient and fast classification of data samples e.g. for predicting quality of products, and generally to assign labels to data.

Artificial evolution of fuzzy classifiers is a promising approach to data mining because evolutionary methods have proven very good ability to find symbolic expressions in various application domains. The general process of classifier evolution can be used to evolve classifiers for different data classes and data sets with different properties. The resulting classifiers can be used as standalone data labeling tools or participate in collective decision in an ensemble of data classification methods.

2. Evolutionary Computation

Genetic algorithms are probably the most popular and wide spread member of the class of evolutionary algorithms (EA). EAs found a group of iterative stochastic search and optimization methods based on mimicking successful optimization strategies observed in nature [12], [13], [14], [15]. The essence of EAs lies in their emulation of Darwinian evolution, utilizing the concepts of Mendelian inheritance for use in computer science [15]. Together with fuzzy sets, neural networks, and fractals, evolutionary algorithms are among the fundamental members of the class of soft computing methods.

EAs operate with a population (also known as a pool) of artificial individuals (also referred to as items or chromosomes) encoding possible problem solutions. Encoded individuals are evaluated using a carefully selected objective function (fitness function) which assigns a fitness value to each individual. The fitness value represents the quality (ranking) of each individual as a solution to a given problem. Competing individuals explore the problem domain towards an optimal solution [13].

For the purpose of EAs proper encoding is necessary; this represents solutions to a given problem as encoded chromosomes suitable for an evolutionary search process. Finding proper encoding is a non-trivial and problem-dependent task affecting the performance and results of an evolutionary search in a given problem domain. Suggested problem solutions might be encoded into binary strings, real vectors, or more complex, often tree-like, hierarchical structures. The encoding selection is based on the needs of a particular application area.

The iterative phasesfor the evolutionary search process starts with an initial population of individuals.The initial population can be generated randomly or seeded with potentially good solutions with respect of chosen encoded scheme. Artificial evolution consists of the iterative application of genetic operators, introducing to the algorithm evolutionary principles such as inheritance, the survival of the fittest, and random perturbations. Iteratively, the current population of problem solutions is modified with the aim of forming new and, hopefully, better population to be used in the next generation. The evolution of problem solutions ends after specified termination criteria have been satisfied, and especially the criterion of finding an optimal solution. However, the decision as to whether a problem solution is the best one (i.e. a global optimum was reached) is impossible in many problem areas. After several iterations, the termination of the search process, the evolution winner was decoded and presented as the most optimal solution where found.

2.1. Genetic operators

Genetic operators and termination criteria are the most influential parameters of every evolutionary algorithm. All the operators presented bellow have several implementations that perform differently in various application areas.

- A selection operator was used for selecting chromosomes from a population. Through this operator, selection pressure is applied to the population of solutions with the aim of picking promising solutions to form the following generation. The selected chromosomes are usually called parents.
- A crossover operator modifies the selected chromosomes from one population to the next generation by exchanging one or more of their subparts and produce new chromosomes (offspring). Crossover is used for emulating sexual reproduction of diploid organisms with the aim of passing on and increasing the good properties of parents for offspring chromosomes.
- A mutation operator introduces random perturbation into chromosome structure; it is used for changing chromosomes randomly and introducing new genetic material into the population.

Besides genetic operators, a termination criterion is another important factor affecting the search process. Widely used termination criteria include:

- Reaching an optimal solution for the problem (which is often hard or impossible to recognize).
- Processing a certain number of generations.
- Processing a certain number of generations without any significant improvement in the population.

In implementation EAs, several generations will be iterated before terminated with respect to genetic operators were presented in Fig.1, as life cycle of process starting from current population then selection and recombination (crossover) and mutation and finally migration.

EAs are a successful, general, and adaptable soft computing concept with good results in many areas. The class of evolutionary techniques consists of more particular algorithms with numerous variants, forged and tuned for specific problem domains. The family of evolutionary algorithms consists of genetic algorithms, genetic programming, evolutionary strategies and evolutionary programming.

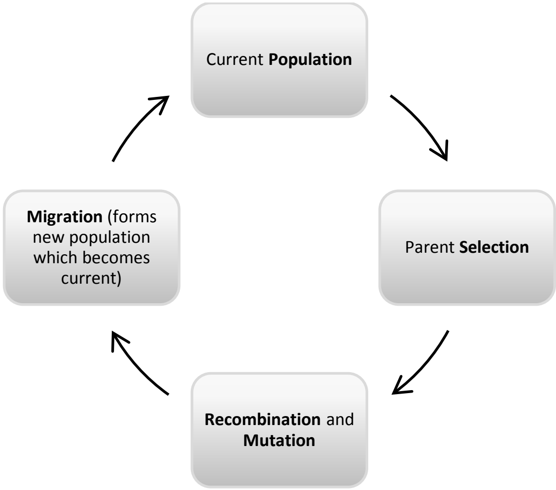

Fig. 1: Iterative Phase of Evolutionary Algorithm

Genetic algorithms (GA) introduced by John Holland and extended by David Goldberg, are widely applied and highly successful EA variant. The basic workflow of the originally proposed standard generational GA is shown in Fig. 2. It presents the steps of EA after choosing the suitable encoded scheme to encode chromosomes, and define the objective function that measures the optimality of chromosomes. Also it demonstrates the process of iterations and the needed operators (selection, crossover and mutation) and finally the termination process.

1. Define the objective function.
2. Encode initial population of possible solutions with fixed-length binary strings.
3. Evaluate the fitness value for all chromosomes in the initial population using the objective function.
4. Create new population (evolutionary search for better solutions):
   a. Select suitable chromosomes for reproduction (parents).
   b. Apply crossover operator on parents with respect to crossover probability to produce new chromosomes (offspring).
   c. Apply mutation operator on offspring chromosomes with respect to mutation probability.
   d. Evaluate the fitness value for the new offspring chromosomes. Add newly constituted chromosomes to new population
   e. While the size of new population is less than the size of current population go back to step a.
   f. Replace current population by the new population.
5. Check termination criteria; if satisfied then the problem solution was founded, but if not go back to step 4.

Fig. 2**:** Genetic Algorithm Workflow

3. Related Work

There were a lot of research efforts on artificial evolution of fuzzy classifiers, predictors and generally symbolic expressions with applications in data mining backed by fuzzy systems. In this section we briefly describe some of them:

Alcalá-Fdez et al. (2008) paper [6], proposed asoftware tool KEEL to assess EA for data miningproblems.Alcalá-Fdez et al. introduce some available data mining software tools, and consider the main strength and weakness of each tool. They categorize KEEL as an alternative to them, as software tool that facilitates the analysis of the behavior of evolutionary learning. KEEL software tool consists of the several function blocks. They developed it to ensemble and use different data mining models. J. Alcalá-Fdez et al. present two case studies functionality and process of creating experiment by KEEL. J. Alcalá-Fdez et al. also developed data visualization tools for the on/off-line modules.

Hüllermeier(2005) paper [7], proposed to convey an impression of the current status and prospects of fuzzy set theory in machine learning, data mining, and related fields. Hüllermeierstudy focused on two of the performancetasks that have attracted much more attention in the fuzzy set theory community. Also, Hüllermeieremphasized the ability of possibility theory to represent partial ignorance as a special advantage in comparison to probabilistic approaches. Also methods for learning graphical models from data and the potential of fuzzy set theory to produce comprehensible and robust models had been pointed out in the paper. It was argued that fuzzy set theoryis especially qualified for data pre- and post- processing, approximation of complex and accurate models, or the presentation of data mining results.

Snášel et al. (2010) [8], evolved a fuzzy classifier in the form of fuzzy search expression by genetic programming. The data mining task was mapped to a fuzzy information retrieval problem and the search for fuzzy classifier was reduced to query optimization problem.

Cordón et al. (1999) [9], proposed a new frame reasoning method to improve the performance of fuzzy rule-based classification systems. Cordón et al. study described different structures for fuzzy rules and a method to learn the parameters of these frame reasoning methods by means of genetic algorithms. Frame reasoning methods fitting the specific problems were evolved.

Muni et al. (2004) [10], proposed a new designed approach with an integratedview of all classes for designing classifiers for a c-class problem by evolving a multi-tree classifier of c trees each representing a classifier for a particular class using a single run of genetic programming. Muni et al. proposed a new concept of unfitness of a tree that was exploited in order to improve artificial evolution. Moreover, new mutation operations where proposed to reduce the destructive nature of mutation operation. Muni et al. used a heuristic rule-based scheme followed by weight-based scheme to resolve conflicting situations.

Muni et al. (2006) [11], proposed a methodology for online feature selection and classifier design using a multi-tree genetic programming based feature selection called GPmtfs. A simultaneous feature selection and classifier design was employed in the GPmtfs. A battery of seven data sets was used for validation of the methodology. Muni et al. compared the performance of the proposed method with the results available in the literature with both filter and wrapper type approaches.

Bridges et al. (2000) [29], proposed an intelligent intrusion detection model to demonstrate the effectiveness of data mining techniques that utilize fuzzy logic and genetic algorithms. Bridges et al. used genetic algorithms to tune the fuzzy membership functions and to select an appropriate set of features. Bridges et al. system architecture allows them to support both anomaly detection and misuse detection components at both the individual workstation level and at the network level.

4. Genetic Programming for Classifier Evolution

The algorithm for fuzzy classifier evolution used in this study was introduced in [8] and builds on the principles of fuzzy information retrieval [16], [17]and evolutionary optimization of search queries [18]. It uses Genetic Programming to find the classifiers that are evaluated with the help of fuzzy set theory.

4.1. Genetic programming

Genetic programming (GP) is an extension to genetic algorithms, allowing work with hierarchical, often tree-like, chromosomes with an unlimited length [19], [20].

Genetic programming was introduced as a tool to evolve whole computer programs and represented a step towards adaptable computers that could solve problems without being programmed explicitly [19], [21].

A genetic programming chromosome takes the form of hierarchical variably-sized expressions, point-labeled structure trees. The trees are constructed from nodes of two types, terminals and functions. More formally, a GP chromosome is a symbolic expression created from terminals t from the set of all terminals T, and functions f from the set of all functions F satisfying the recursive definition [21]:

- $\forall t \in T$: t is the correct expression.
- $\forall f \in F$: $f(e_1, e_2, \ldots, e_n)$ is thecorrect expression if $\forall f \in F$ and $e_1, \ldots, e_n$ are correct expressions. The function *arity*(f) represents the arityof f.
- There are no other correct expressions.

GP chromosomes are evaluated by the recursive execution of instructions corresponding to tree nodes[21].

Terminal nodes are evaluated directly (e.g. by reading an input variable) and functions are evaluated afterleft-to-right depth-first evaluation of their parameters.

Genetic operators are applied to the nodes in tree-shaped chromosomes. A crossover operator is implemented as the mutualexchange of randomly selected sub-trees of the parent chromosomes as shown in Fig. 3.

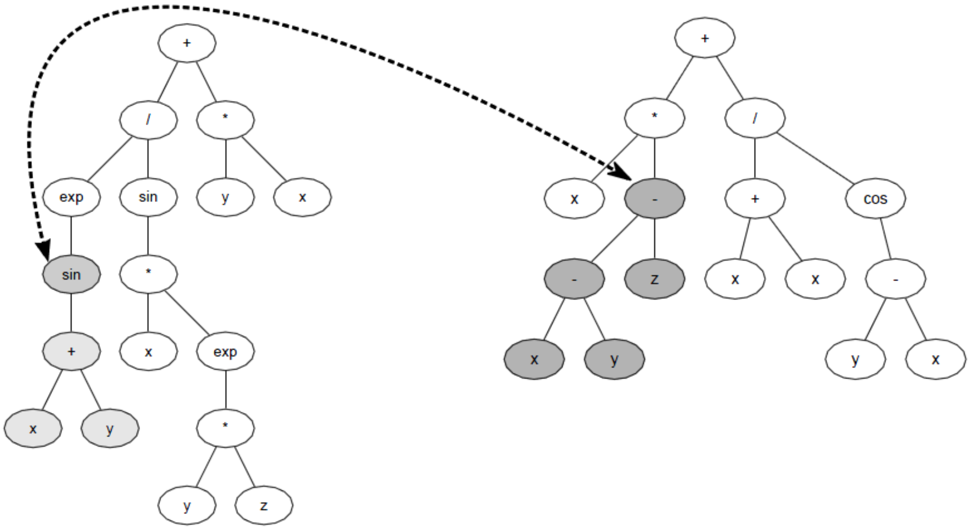

Fig. 3: GP crossover operator

Mutation has to modify the chromosomes by pseudo-random arbitrary changes in order to prevent premature convergence and broaden the coverage of the fitness landscape. Mutation could be implemented as:

- removal of a sub-tree at a randomly chosen node
- replacement of a randomly chosen node by a newly generated sub-tree
- replacement of node instruction by a compatible node instruction (i.e. a terminal can be replaced by another terminal, function can be replaced by another function of the same arity)
- acombination of the above.

4.2. Fuzzy Classifier

Fuzzy set theory has been applied successfully in a variety of application areas. The fuzzy classifier takes form of a symbolic expression with data features (data set attributes) as terminals and operators as non-terminal nodes. Both terminals and non-terminals are weighted, as presented by example in Fig. 4.

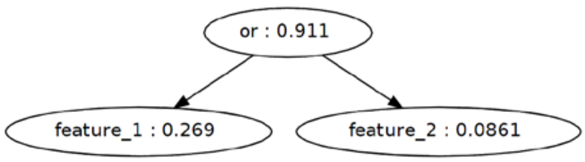

Fig. 4: An example of fuzzy classifier

Fuzzy classifier was evaluated for each data sample in the training collection. For each terminal, the value of corresponding feature is taken. The operators are implemented with the help of fuzzy set operators. The standard implementation of fuzzy set operators were used but any other pair of *t-norm* and *t-conorm* could be used. We also note that additionaloperators (e.g. various ordered weighted averaging aggregation operators) could be added.

5. Data Mining

Data mining (DM) is the process of

automatically exploring large amount of data to extract the interesting knowledge, or extract patter from attributes using pattern recognition [30]. There are two distinct learning's supervised and unsupervised and different tasks for each. Classification and classification rule learners' tasks for unsupervised learning. Clustering and association rule learners' tasks for unsupervised learning.

Data mining can be applied and used in optimization problems handled by a metaheuristic.Decision tree is one of the heuristic methods. So, for aclassical model in data mining classification tasks can be represented using decision trees using GP of the "optimal" decision tree from a training set (or learning set) of data.

Fuzzy sets and fuzzy logic can be used for efficient data classification by fuzzy rules and fuzzy classifiers. Numbers of fuzzy rules are extracted in a heuristic manner based on a rule evaluation criterion which can be viewed as fuzzy data mining [31]. Fuzzy data mining can be used for pattern classification problems.

6. Experiments

Genetic programming was used to evolve fuzzy rules describing faulty products inan industrial plant.During the production, a number of sensory inputs are read to record materialproperties, production flow and product features. The features include the chemicalproperties of the raw material, density, temperature at several processing stages, andmany other values recorded several times during the production process. At the end,the product is classified as either valid or corrupt. The data and classification for anumber of product samples is known and the goal of the genetic programming is tofind a fuzzy classifier that could be used for product quality prediction.

We have obtained data sets from 5 different production lines of a productionplant. The data sets contained readings from 508 sensors for each product. For eachproduction line, the data was divided into training (40%) and test (60%) collection.

We label the data sets D1, D2, D3, D4 and D5 respectively. Selected propertiesof the data sets are shown in Tab.1. All five data sets have the same number offeatures but since they come from different processing lines, their internal structurediffers and the patterns describing faulty products are unique for each of them.

Table.1: Description of the data sets

Name	**D1**	**D2**	**D3**	**D4**	**D5**
Features	508	508	508	508	508
Training samples	562	154	755	4881	2022
Test samples	844	233	1134	73226	3034

The proposed algorithm for classifier evolution was implemented and classifierswere sought for all five training sets. The results of the classification on collectionsare shown in Tab.2. The table shows the overall accuracy (OA), the percent of falsepositives (FP) and the percent of false negatives (FN) obtained by the best classifiers.

Table.2: Results of classification of test data collections

	Data set				
	D1	D2	D3	D4	D5
OA	97.63	97.00	99.50	96.99	99.60
FP	1.30	3.00	0	0.43	0.07
FN	1.7	0	0.53	2.58	0.33

7. Further Possibilities and Techniques

There are other alternative techniques of genetic programming nature. Generally, there are two well-known methods, which can be used for synthesis of various programs by means of computers. The first one is called genetic programming or GP, [19, 22] and the other is grammatical evolution [22],[23].

The idea as how to solve various problems using symbolic regression (SR) by means of EA was introduced by John Koza, who used genetic algorithms (GA) for GP. Genetic programming is basically a symbolic regression, which is done by the use of evolutionary algorithms, instead of a human brain. The ability to solve very difficult problems is now well established, and hence, GP today performs so well that it can be applied, e.g. to synthesize highly sophisticated electronic circuits [24].

In the last decade of the 20thcentury, C. Ryan developed a novel method for SR, called grammatical evolution (GE). Grammatical evolution can be regarded as an unfolding of GP due to some common principles, which are the same for both algorithms. One important characteristic of GE is that it can be implemented in any arbitrary computer language compared with GP, which is usually done (in its canonical form) in LISP. In contrast to other evolutionary algorithms, GE was used only with a few search strategies, for example with a binary representation of the populations in [25]. Another interesting investigation using symbolic regression was carried out by [26] working on Artificial Immune Systems or/and systems, which are not using tree structures like linear genetic programming.

But simply, evolutionaryalgorithm simulates Darwinian evolution of individuals (solutions of given problem) on a computer and are used to estimate-optimize numerical values of defined cost function. Methods of GP are able to synthesize in an evolutionary way complex structures like electronic circuits, mathematical formulas etc. from basic set of symbolic (nonnumeric) elements.

Another alternative method of symbolic regression called Analytic programming (AP) in [27].This method can be used as well as GP or GE for various model syntheses and has been successfully compared to many standard problems with GP, with very good results. Our next step is to use AP and other alternative methods to create more complex comparative study, focused on problems, mentioned here.

8. Conclusions

This work presents an application of evo-fuzzy data mining technique to the classification of data samples in a real world industrial data set. Genetic programming has been used to evolve fuzzy classifiers in form of weighted symbolic expressions aggregating data features with the help of a set of operators. In contrast to previous efforts in this area (see e.g. [28], [9], [10], [11]), this approach is inspired by information retrieval. The information retrieval inspired fuzzy rules containing the operators *and*, *or*, and *not* provided a rich and flexible tool to express detailed soft classification criteria.

Data classes were interpreted as membership degrees of a fuzzy set and the algorithm sought for a classifier that would describe such a fuzzy set. In this sense, the described approach also differs from most of the traditional rule-based fuzzy classifiers that aim to mine the if-then relations from data.

The evolution of fuzzy classifier takes a number of parameters. The set of classifier operators, the interpretation of classifier weights and the fitness function can all be altered.

Last but not least, the alternative methods for evolutionary symbolic regression will be employed to find fuzzy classifiers.

Acknowledgements

This paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07 /2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References:

[1] Javier Sedano, Leticia Curiel, Emilio Corchado, Enrique de la Cal, and José R. Villar. A soft computing method for detecting lifetime building thermal insulation failures.*Integr.Comput.-Aided Eng.*, Vol.17,April 2010, pp.103–115.

[2] Emilio Corchado and Alvaro Herrero.Neural visualization of network traffic data for intrusion detection.*Applied Soft Computing*, In Press, Corrected Proof, 2010.

[3] Angel Arroyo Emilio Corchado and VerÄ‚Ĺ‚nicaTricio. Soft computing models to identify typical meteorological days.*Logic Journal of thel IGPL*, July 2010.

[4] James C. Bezdek, James Keller, Raghu Krisnapuram, and Nikhil R. Pal. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing *(The Handbooks of Fuzzy Sets). Springer-Verlag New York, Inc., Secaucus, NJ, USA*, 2005.

[5] AntanasVerikas, Jonas Guzaitis, AdasGelzinis, and MarijaBacauskiene.A general framework for designing a fuzzy rule-based classifier.*Knowledge and Information Systems*, 10.1007/s10115-010-0340-x, pp. 1-19.

[6] J. Alcalá-Fdez, 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, October 2008, pp.307–318, .

[7] EykeHüllermeier. Fuzzy methods in machine learning and data mining: Status and prospects. *Fuzzy Sets and Systems*, Vol.156, No.3,2005, pp.387 – 406.

[8] Václav Snásel, Pavel Krömer, Jan Platos, and Ajith Abraham.The evolution of fuzzy classifier for data mining with applications.*In Kalyanmoy Deb, Arnab Bhattacharya, NirupamChakraborti, ParthaChakroborty, Swagatam Das, JoydeepDutta, Santosh K. Gupta, Ashu Jain, VarunAggarwal, Jürgen Branke, Sushil J. Louis, and Kay Chen Tan, editors, SEAL, of Lecture Notes in Computer Science, Springer,* Vol. 6457, 2010,pp.349–358.

[9] Oscar Cordón, María Jose del Jesus, and Francisco Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems.*International Journal of Approximate Reasoning*, Vol.20, N.1, 1999, pp.21 – 45.

[10] D.P. Muni, N.R. Pal, and J. Das.A novel approach to design classifiers using genetic programming.*Evolutionary Computation, IEEE Transactions on*, Vol.8, No.2, April 2004, pp.183 – 196.

[11] D.P. Muni, N.R. Pal, and J. Das.Genetic programming for simultaneous feature selection and classifier design.*Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, Vol.36, No.1, February 2006, pp.106 –117.

[12] MehrdadDianati, Insop Song, and Mark Treiber.An introduction to genetic algorithms and evolution strategies.*Technical report, University of Waterloo, Ontario, N2L 3G1, Canada*, July 2002.

[13] Gareth Jones. Genetic and evolutionary algorithms.*In Paul von Rague, editor, Encyclopedia of Computational Chemistry.John Wiley and Sons*, 1998.

[14] Melanie Mitchell. An Introduction to Genetic Algorithms.*MIT Press, Cambridge, MA,* 1996.

[15] Ulrich Bodenhofer. Genetic Algorithms: Theory and Applications. *Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter* 2003/2004.

[16] Oscar Cordón, Félix de Moya, and Carmen Zarco.Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments.*In IEEE International Conference on Fuzzy Systems,Budapest, Hungary,* 2004, pp.571–576.

[17] Fabio Crestani and Gabriella Pasi. Soft information retrieval: Applications of fuzzy set theory and neural networks. *In N. Kasabov and R. Kozma, editors, Neuro-Fuzzy Techniques for Intelligent Information Systems,Springer Verlag, Heidelberg, DE,* 1999, pp.287–315.

[18] DusanHúsek, Václav Snášel, Roman Neruda, Suhail S. J. Owais, and Pavel Krömer.Boolean queries optimization by genetic programming.*WSEAS Transactions on Information Science and Applications,*Vol.3, No.1, 2006, pp.15–20.

[19] J. Koza. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. *Technical Report STAN-CS-90-1314, Dept. of Computer Science, Stanford University,* 1990.

[20] John R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection. *MIT Press, Cambridge, MA, USA,* 1992.

[21] Michael Affenzeller, Stephan Winkler, Stefan Wagner, and Andreas Beham. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. *Chapman & Hall/CRC*, 2009.

[22] John R. Koza, David Andre, Forrest H. Bennett, and Martin A. Keane. Genetic Programming III: Darwinian Invention & Problem Solving. *Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition,* 1999.

[23] Conor Ryan, J. J. Collins, and Michael O'Neill. Grammatical evolution: Evolving programs for an arbitrary language. *In Proceedings of the First European Workshop on Genetic Programming, EuroGP '98, London, UK, Springer-Verlag*, 1998, pp.83–96,

[24] John R. Koza, Martin A. Keane, and Matthew J. Streeter.Evolving inventions.*Scientific American*, February 2003.

[25] John O'Sullivan and Conor Ryan.An investigation into the use of different search strategies with grammatical evolution.*In Proceedings of the 5thEuropean Conference on Genetic Programming, EuroGP '02, London, UK, Springer-Verlag, 2002*.pp.268–277

[26] Colin G. Johnson. Artificial immune system programming for symbolic regression.*In Proceedings of the 6th European conference on Genetic programming, EuroGP'03, Berlin, Heidelberg, 2003.Springer-Verlag*.pp. 345–353.

[27] Ivan Zelinka, Donald Davendra, Roman Senkerik, Roman Jasek, and ZuzanaOplatkova.Analytical programming - a novel approach for evolutionary synthesis of symbolic structures.

[28] Brian Carse and Anthony G. Pipe.A framework for evolving fuzzy classifier systems using genetic programming.*In Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2001*,pp.465–469.

[29] Susan M. Bridges and Rayford B. Vaughn. Fuzzy data mining and genetic algorithms applied to intrusion detection. *Inproceeding of the 23rd National Information Systems Security Conference, October 2000,* pp.13-31.

[30] Al-GhazaliTalbi. *Metaheuristics from design to implementation*.John Wiley & Sons, Inc., Hoboken, New Jersey.USA. 2009.

[31] HisaoIshibuchi, Yusuke Nojima, and Isao Kuwajima.Fuzzy data mining by heuristic rule extraction and multiobjective genetic rule selection.*IEEE International conference on fuzzy systems,* July 2006,pp.1633-1640.

About Authors

Suhail OWAIS received the M.Sc. in Computer Science from the Al al-Bayt University, Mafraq, JORDAN, 2003, and Ph.D. in Computer Science and Applied Mathematics from the department of Computer Science, VŠB –Technical University of Ostrava in 2006. Suhail interest in research areas include Evolutionary Algorithms (GA, GP), information retrieval, World Wide Web, data mining, fuzzy logic, and search engines. He has authored 9 papers published in journals, and 12 papers published in conferences, and 8 papers published in workshops, and 2 book chapters. Suhail had several activities such as Regular Member in the Machine Intelligence Research Scientific Network for Innovation and Research Excellence Labs, and as External Examiner Committee Member for the Ph.D. students, and acted as an Evaluator of Research Proposals.

Pavel KRÖMERreceived the M.Sc. in computer science from the VSB - Technical University of Ostrava in 2006 and Ph.D in Applied Mathematics from the same institute in 2010. Pavels areas of interest include computational intelligence, information retrieval, World Wide Web, and data mining. He coauthored more than 90 papers published in computer science proceedings and journals. He also presented award winning papers at international computer science conferences in 2007, 2008, and 2011.

JanPLATOŠreceived the M.Sc.and Ph.D. degree from the department of Computer Science, VŠB –Technical University of Ostrava in 2006 and 2010 respectively. His research interests cover data compression, computer graphics, data mining, bio-inspired computing, and GPGPU computing. He is the member of several professional organizations including the IEEE and SDIWC. He has served as program co-chair and organizer of several international conferences and workshops, e.g. the ICDIPC 2011, Ostrava, Czech Republic. He co-authored conference papers awarded in 2008 and 2011.

Václav SNÁŠELreceived the M.Sc. degree fromPalacky University, Olomouc, Czech Republic, andthe Ph.D. degree in algebra and geometry fromMasaryk University, Brno, Czech Republic.In 2001, he was a Visiting Scientist at the Instituteof Computer Science, Academy of Sciences, CzechRepublic. Since 2003, he has been a Vice Dean for researchand science at Faculty of Electrical Engineeringand Computer Science, VŠB—Technical University of Ostrava, Ostrava, Czech Republic,where he has been a Dean since 2010 and aFull Professor since 2006. He was also with industrial company, where he wasinvolved in different industrial research and development projects for more than10 years. He has been engaged in research and development for more than 27years in the industry and academia. He has given more than 12 plenary lecturesand conference tutorials in various areas. He has authored/coauthored severalrefereed journal/conference papers, book chapters, and more than 350 papers(138 are recorded at Web of Science). He was also the Supervisor for manyPh.D. students from Czech Republic, Jordan, Yemen, Slovakia, Ukraine, andVietnam. He is the Editor-in-Chief of two journals. He has also been engagedwith the editorial board of some reputed international journals. His researchinterests include multidisciplinary environment involving artificial intelligence,multidimensional data indexing, bioinformatics, information retrieval, semanticWeb, knowledge management, data compression, machine intelligence, neuralnetwork, Web intelligence, data mining and applied to various real-worldproblems.

Ivan ZELINKA (ivanzelinka.eu) received the M.Sc. degree from Technical University of Brno, Czech Republic, and the Ph.D.

degree in Technical Cybernetics from Tomas Bata University in Zlin (awarded by Siemens award). In 2004, he got associated professor of Technical Cybernetics at the Technical University of Brno and professor of Informatics at the VSB-TU Ostrava. Ivan Zelinka is a member of numerous program committees of various conferences and founder and editor in chief of new Springer book series Emergence, Complexity and Computation (www.ecc-book.eu). He is also member of British Computer Society CITP (Certified IT Professional), IEEE and local IEEE Computational Intelligence committee member http://www.ieee.cz/ci.