Lotfi A. Zadeh
Dan Tufiş
Florin Gheorghe Filip
Ioan Dziţac
(Eds.)

From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence

Exploratory Workshop on NL-Computation
Băile Felix; Oradea, Romania, May 15-17, 2008

EDITURA ACADEMIEI ROMÂNE

Editing House of Romanian Academy

2008

Editors:

1. **Lotfi A. Zadeh**, University of California- Berkeley, CA 94720-1776, USA, E-mail: zadeh@cs.berkeley.edu

2. **Dan Tufiş**, Research Institute for Artificial Intelligence of the Romanian Academy, Calea 13 Septembrie, No. 13, Casa Academiei, Rooms 1236 - 1245, Bucharest 050711, Romania, E-mail: tufis@racai.ro

3. **Florin Gheorghe Filip**, Romanian Academy, Bucharest, 125, Calea Victoriei, 010071 Bucharest-1, Romania, E-mail: ffilip@acad.ro

4. **Ioan Dziţac**, Agora University, Oradea, Romania Piaţa Tineretului, 8, Oradea - 410526, Bihor, România, E-mail: idzitac@univagora.ro

Descrierea CIP a Bibliotecii Naţionale a României

FROM NATURAL LANGUAGE TO SOFT COMPUTING: NEW PARADIGMS IN ARTIFICIAL INTELLIGENCE. EXPLORATORY WORKSHOP ON NL-COMPUTATION
(2008, Băile Felix)
From natural language to soft computing: new paradigms in artificial intelligence: exploratory workshop on NL-computation: International Conference on Computers, Communications and Control - ICCCC 2008: Băile Felix,
May, 15-17 / ed.: Lotfi A. Zadeh, Dan Tufiş, Florin Gh. Filip, Ioan Dziţac. - Bucureşti: Editura Academiei Române, 2008
Bibliogr.
Index.
ISBN 978-973-27-1678-6

I. Zadeh, Lotfi A. (ed.)
II. Tufiş, Dan (ed.)
III. Filip, Florin Gheorghe (ed.)
IV. Dziţac, Ioan (ed.)
V. International Conference on Computers, Communications and Control (2008: Băile Felix)

004(063)

Contents

From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence 5
Editors

World Knowledge for Controllers by Fuzzy-Interpolative Systems 11
Marius M. Bălaş and Valentina E. Bălaş

A Survey on the Constant Time to Collision Techniques 22
Valentina E. Bălaş and Marius M. Bălaş

Non-Negative Matrix Factorization Methods and their Applications 32
Ioan Buciu and Ioan Nafornişă

A Multi-Agent Stigmergic Model for Complex Optimization Problems 51
Camelia Chira, Camelia M. Pintea and D. Dumitrescu

Conceptual Model of an Intelligent Supervising System for Evaluating Policies in the Operation of an Underground Mine 63
Felisa M. Cordova and Luis E. Quezada

Neuro-Fuzzy Modeling of Event Prediction using Time Intervals as Independent Variables 84
Simona Dziţac, Ioan Felea, Ioan Dziţac and Tiberiu Vesselenyi

Towards to Colony of Robots 100
Gastón Lefranc

Constructing Multi-dimensional Utility Function under Different Types of Preferences over the Fundamental Vector Attributes 129
Natalia Nikolova, Daniela Toneva, Sevda Ahmed and Kiril Tenekedjiev

Real Sets as Consistent Boolean Generalization of Classical Sets 150
Dragan G. Radojevic

Simulating NEPs in a Cluster with jNEP 172
Emilio del Rosal, Rafael Nuñez, Carlos Castañeda and Alfonso Ortega

Soft Computing for Intelligent Systems 185
Imre J. Rudas and János Fodor

**New Parallel Programming Language Design:
A Bridge between Brain Models and Multi-Core/Many-Core Computers?** 196
Gheorghe Stefanescu and Camelia Chira

Playing with Word Meanings 211
Dan Tufiş

**An Extended Abstract of Workshop's Keynote Lecture:
"A New Frontier in Computation - Computation with Information
Described in Natural Language"** 224
Lotfi A. Zadeh

Author index 226

**APPENDIX 1
Slides of PowerPoint Presentation of Workshop's Keynote Lecture:
"A New Frontier in Computation - Computation with Information
Described in Natural Language"** 227
Lotfi A. Zadeh

**APPENDIX 2
Pictures from EWNLC 2008** 268

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 5-10

From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence

Editors

Abstract: In this volume we publish some papers presented at the Exploratory Workshop on Natural Language Computation (EWNLC 2008), entitled ”From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence”, during May 15-17, 2008, Baile Felix, Oradea, Romania. This workshop was financed by the Romanian Ministry of Education, Research and Youth/ National Authority for Scientific Research, based a project directed by Prof. dr. Dan Tufiş, corresponding member of Romanian Academy, Code of project: PO-01-Ed1-R0-F37, contract no. 8/ 2008. EWNLC 2008 was organized as satellite-event of conference ICCCC 2008[a] (General Chair: Prof. dr. Ioan Dziţac).

[a] International Conference on Computers, Communications & Control, founded by I. Dziţac, F.G. Filip & M.-J. Manolescu, http://www.iccc.univagora.ro

Keywords: Natural Language Computation (NL-Computation), Generalized Constraint Language (GCL), Granular Computing (GC), Generalized Theory of Uncertainty (GTU), Artificial Intelligence (AI).

Key Idea: ”Computation with information described in natural language (NL) is closely related to Computing with Words. NL-Computation is of intrinsic importance because much of human knowledge is described in natural language. This is particularly true in such fields as economics, data mining, systems engineering, risk assessment and emergency management. It is safe to predict that as we move further into the age of machine intelligence and mechanized decision-making, NL-Computation will grow in visibility and importance.” (Zadeh, 2007)

1 Introduction

The Exploratory Workshop “*From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence*”(EWNLC 2008) was organized as satellite-event of ICCCC 2008 by Dr. Dan Tufiş - director of Institute of Artificial Intelligence of Romanian Academy - and Dr. Ioan Dziţac - Agora University of Oradea, and was addressed to the 30 participants, from Romania and other 6 countries, selected from over 100 participants at ICCCC 2008: scientific researchers, PhD and PhD students, from universities and research units. The selection of the participants was based on the analysis of the scientific experience and the capacity to actively participate to the debates of each of the 100 participants who submitted papers for ICCCC 2008. Some other criteria were also used: geographical distribution (Chile, France, Greece, Hungary, Romania (Arad, Bucuresti, Cluj-Napoca, Iasi, Oradea, Sibiu, Trgoviste), Serbia, US); repartition according to the age (experienced researchers - 24 PhDs and young researchers - 6 PhD candidates) and sex (half of the participants are women, both experienced and young researchers); the repartition among universities and other research centers; repartition based on the domain of expertise of potential participants (electronics and telecommunications, computer science, artificial intelligence, economical informatics, automatics, mathematics, economy, communication sciences, robotics, medical informatics).

The main keynote speaker at EWNLC 2008 was Lotfi A. Zadeh, the founder of the Fuzzy Set Theory and Fuzzy Logic. In the last years professor Zadeh developed a theory concerning the representation of natural language like a computing language. One fundamental concept of this theory is ”precisiation” of natural language, in the sense of transforming it in a precise, formal construct. The precisiated natural language (PNL) is the result of the transformations of natural language constructions into constructions of a Generalized Constraint Language (GCL). The expressive power of GCL is far greater than that of other AI languages (LISP, PROLOG) or other conventional logic-based meaning-representation languages (predicate logic, modal logic, a.s.o.). The main reason of this research is that most of the applications based on Natural Language Processing (semantic document categorisation, automatic text summarization, human-machine dialogue, machine translation) could be redefined in terms of GCL representations, with the advantage of a more precise processing of the perceptual information and of a more direct approach to the cognitive objectives of AI.

2 Scientific contents of the event

2.1 Objectives achieved

The most important objective reached during the workshop is that it opened the *way for an interdisciplinary collaboration between researchers in different countries* (Romania, USA, France, Serbia, Chile, Greece and Hungary), with different professional experience (scientific researchers, doctors and Ph.D. students from universities and research units), in order to apply in jointly international research projects. Specialists in the field of natural language processing and computation based on precisiated natural language (a concept introduced by prof. Zadeh) proposed new topics, discussed current and new research directions, and they made oral agreements for future collaboration in principle.

Another achieved objective was the identification and strengthening of an ISI Journal (International Journal of Computers, Communications and Control - IJCCC) for the dissemination of future research results in the domain of the workshop.

The papers presented at the workshop were very interesting and appreciated, so that, together with the members of the program committee members of the workshop, we decided, besides publishing the abstracts and/or their reduced forms in the ICCCC 2008 volumes, to prepare a distinct volume, to be published by the Romanian Academy Publishing House. This volume will include the extended versions of the key lectures, along with other papers selected from ICCCC 2008, with topics related to the workshop, in an extended form. Professor Zadeh was very pleased and agreed to participate to the editing this volume.

2.2 General topics

Natural Language Computation

- Lotfi A. ZADEH, University of California, USA
- Vasile BALTAC, SNSPA Bucharest, Romania
- Boldur BĂRBAT, Lucian Blaga University of Sibiu, Romania
- Pierre BORNE; Ecolle Centrale de Lille, France
- Dan TUFIŞ, Institute for Research in Artificial Intelligence, Romanian Academy

Intelligent Systems

- Florin FILIP, Romanian Academy, Romania
- Stephan OLARIU, Old Dominion University, USA
- Athanasios STYLIADIS, ATEI Thessaloniki, Greece
- Janos FODOR, Budapest Tech, Hungary

Artificial Intelligence

- Ioan BUCIU; University of Oradea, Romania
- Gaston LEFRANC, Pontifical Catholic University of Valparaiso, Chile
- Gheorghe PĂUN, Institute for Mathematics of the Romanian Academy & University of Seville
- Dragan RADOJEVIC, Institute "Mihailo Pupin" from Belgrade, Serbia
- Gheorghe ŞTEFĂNESCU, University of Bucharest

2.3 Conclusions / results

The keywords of the workshop were: *natural language computation, language based on generalized constraints, granular calculation, generalis theory of uncertainty, soft computing and artificial intelligence.*

The imprecision of natural language is one of the main difficulties in the intelligent man-machine interaction. Prof. Lotfi Zadeh has proposed a theory that treats exactly this problem, as well as an extremely powerful calculation formalism, built on the very imprecise natural language constructions.

The abstracts of the presented lectures, respectively a part of the full text of the key lectures have been or are in course to be published in:

- Ioan Dziţac, Simona Dziţac, Loredana Galea, Horea Oros (eds.), *Abstracts of ICCCC Papers: ICCCC 2008*, ISSN 1844 - 4334, p. 88
- Ioan Dziţac, Florin Gheorghe Filip, Mişu-Jan Manolescu (eds.), - *Proceedings of ICCCC 2008*, in *International Journal of Computers, Communications and Control* (IJCCC), Vol.

I (2006), supplementary issue, ISSN 1841-9836, 548 p.

• Lotfi A. Zadeh, Dan Tufiş, Florin Gheorghe Filip, Ioan Dziţac (eds.), *From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence* , Editing House of Romanian Academy, ISBN: 978-973-27-1678-6, 2008 (this volume).

2.4 Contributions to the development of future directions in the domain of natural language computation

Some of the workshop participants were members of the Editorial Board of the "International Journal of Computers, Communications and Control" - IJCCC Journal edited by Agora University and recently listed by CNCSIS in category A journals (ISI journal), more precisely:

- Florin Gheorghe Filip (Founder and Editor in Chief);
- Ion Dziţac (founder and Associate Editor in Chief);
- Mişu-Jan Manolescu (Founder and Managing Editor);
- Ioan Buciu (Associate Executive Editor);
- Pierre Borne (Associate Editor);
- George Păun (Associate Editor);
- Dan Tufiş (Associate Editor);
- Athanasios Styliadis (Associate Editor);
- Horea Oros (Editorial Secretary).

During the workshop, two very important decisions on how to increase the quality and specificity of the IJCCC journal have been taken:

- Granting a larger area to the topic of natural language computation;
- Cooptation in the IJCCC Editorial Board, as Associate Editors, of two of the workshop participants, currently in the United States: Stephan Olariu (Norfolk) and Lotfi A. Zadeh (San Francisco).

3 Information on the event organization

3.1 Identifying the opportunity, the organizing institution and the workshop theme

The opportunity to organise an exploratory workshop was identified on a CNSIS web page, http://www.cncsis.ro/PN2_idei_2008_wexplor.php by the ICCCC 2008 organizers (I. Dziţac, and F.G. Filip and M.-J. Manolescu). They sent an invitation to the Research Institute for Artificial Intelligence of the Romanian Academy (ICIA-AR) to organize such a workshop as a satellite event of ICCCC 2008 on a topic of great interest today.

The director of the institute, Dr. Dan Tufiş, proposed the theme "From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence" which was immediately agreed by the ICCCC 2008 organizers, the Agora University and the Romanian Academy Forum for the Knowledge Society.

3.2 Aspects of preliminary management and marketing

The project proposal for organizing the workshop was written by dr. Dan Tufiş in collaboration with dr. Ioan Dziţac, cf. Collaboration Agreement 300/25.03.2008.

There were identified the relevant lecturers and they were invited to participate. One of the initially invited key lecturer, prof. H.-N. Teodorescu, could not participate in the end, but dr. Gheorghe Păun and prof. Gheorghe Ştefănescu have accepted to deliver invited lectures.

Besides the 14 experts, we selected 16 participants out of the ICCC2008 who explicitly expressed their interest in the workshop topics. From these, prof. dr. Marius Guran, prof. dr. Ştefan Iancu and drd. Văleanu Emma, could not participate and they were replaced by prof. dr. Grigore Albeanu, prof. dr. Adriana Manolescu and drd. Ciprian Popescu, whose profile were similar to that of the initial participants.

Initial contacts were established and a permanent connection was kept with the key-lecturers through letters, e-mail, fax and telephone.

The event was mediatised through posters, media, programs, and web pages:

- http://www.iccc.univagora.ro/?page=workshop
- http://www.cncsis.ro/mai2008i.php
- http://www.racai.ro/events/events.html
- http://www.academiaromana.ro/forum_info/fpsc_anunturi.htm

The agreements and conventions for accommodation and meals were established with the Termal Hotel, Felix Spa (through the Agora University).

The agreements for publishing and printing were established with the Publishing House of the Agora University, the Publishing House of the Romanian Academy and the Metropolis Ltd. Printing House.

3.3 During the event

On May 14, 2008, the participants were met at the Hotel Termal in Felix Spa.On May 15, 2008, the opening ceremony took place, having the members of the presidium as speakers: in picture 3, from left to right: Dan Tufiş, Ioan Dziţac, Lotfi A. Zadeh, Mişu Manolescu and Florin Gheorghe Filip. On this occasion the vice president of the Romanian Academy F.G. Filip handed to Lotfi A. Zadeh a Diploma of Excellence for his entire contribution to the development of the scientific field of Artificial Intelligence, on behalf of the Section of Science and Information Technology of the Romanian Academy.

After the opening ceremony, the scientific part of the workshop, according to the final program, was opened by the lecture "*A New Frontier in Computation - Computation Described in Natural Language*", given by professor Zadeh.

After the presentation of other six key-lectures (Baltac, Borne; Filip, Fodor, Olariu, Păun), the participants were offered a Romanian evening, with traditional Romanian dishes and a folk show delivered by the band "Nuntaşii Bihorului".

On May 16, 2008, other seven key-lectures were presented (Bărbat, Lefranc, Radojevic, Ştefănescu, Tufiş, Buciu and Styliadis).

On May 17, 2008, a roundtable was moderated by I. Dziţac, F.G. Filip, M.-J. Manolescu, D. Tufiş, L.A. Zadeh. Besides the members of the presidium, the following participants addressed to the public: B. Bărbat, A. Styliadis, I. Moisil, G. Păun, S. Olariu, V. Baltac, G. Lefranc, S. Dziţac, V. Judeu, T. Vesselenyi, M. Văleanu and others. On this occasion a new orientation of the IJCCC journal's editorial policy was discussed and two highly reputed scholars, prof Olariu and prof. Zadeh, accepted to join in the IJCCC Editorial Board.

During the afternoon a meeting was held with all the workshop participants and various networks of collaboration have been set up, their main connection point being the IJCCC journal.

On May 18, after 4 nights of accommodation at Hotel Termal, the participants have left the Felix Spa with many new experiences and pleasant memories, as proved by the many thankful messages received from the participants after their returning home.

Oradea, May, 2008
Editors

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 11-21

World Knowledge for Controllers by Fuzzy-Interpolative Systems

Marius M. Bălaş and Valentina E. Bălaş

Abstract: The paper is discussing the necessity of providing the close loop controllers with incipient elements of world knowledge: general knowledge on system theory and specific knowledge on the processes. This can be done with fuzzy-interpolative systems, allied with simulation models and/or planners. Some planned controller case studies are illustrating the idea.

Keywords: fuzzy-interpolative controllers, fuzzy-interpolative expert systems, knowledge embedding by computer models, planners.

1 Introduction

In ref. [1] Lotfi A. Zadeh affirmed that the main weakness of the Question-Answering Systems is the absence of the world knowledge. *World knowledge* WK is the knowledge acquired through experience, education and communication.

The components of WK are [1]:

- Propositional: Paris is the capital of France;
- Conceptual: Climate;
- Ontological: Rainfall is related to climate;
- Existential: A person cannot have more than one father;
- Contextual: Tall.

Some of the main characteristics of WK are:

- Much of WK is perception-based;
- Much of WK is negative, i.e., relates to impossibility or nonexistence;
- Much of WK is expressed in a natural language.

Obviously KW is highly necessary to the human emulating AI products. Nevertheless this approach must overcome lots of difficulties: WK need huge memory capacity, the representation techniques must be in the same time comprehensive, specific and portable, the selection of the knowledge, the learning and the forgetting processes need further fundamental conceptual investigations, etc.

Our interest in linked to the automate control of the nonlinear and time varying processes, one particular domain where AI is extremely needed but not easy to apply. Sometimes, unexpected accidents, as the Ariane 5 launch failure (caused essentially by a human confusion between the guidance system software of Ariane 5 with the old one, belonging to Ariane 4), are proving that our control technologies, although very sophisticated in many aspects, are facing obstacles in the real world.

A crucial feature for controllers would be the ability to recognize on-line the environment (the type and the technical characteristics of the process) and the operating regime, in order to extend as much as possible its self-adaptive capabilities.

The aim of this paper is to answer the following question: "Can low level computing devices: μP,μC, DSP, etc. benefit of WK, when even the sophisticated modern AI software, running on powerful workstations, is encountering difficulties?"

A positive answer to this question is essential, because we think that only this way the automate systems will reach the performances and the safety capabilities we really need. Besides other possible ways to solve this problem, we have already proposed a methodology in this sense: the Fuzzy-Interpolative.

This paper is a development of the ref. [2] paper.

2 The Fuzzy-Interpolative Systems

A *fuzzy-interpolative controller* FIC is a fuzzy controller that can be equaled with a corresponding look-up table with linear interpolations. The FIC concept must not be confounded with *the fuzzy rule interpolation*, originally introduced by L.T. Kóczy and K. Hirota [3, 4].

A typical FIC is a Sugeno controller with triangular or trapezoidal fuzzy partitions, prod-sum inference and COG defuzzyfication [5, 6, 7, 8], etc. The interpolative counterpart of this controller is the look-up table with linear interpolations (as the corresponding Simulink-Matlab block). FICs started from the practical observation that the Matlab FIS (Fuzzy Inference System) toolkit is demanding notable resources and occasionally it encounters computational problems, while an equivalent look-up-table performs almost instantly, although they are producing the same control surface.

The fundamental advantage of FICs is the easiness of their implementation. In high level programming languages the look-up tables bring effectiveness, resources saving and quick developments. In fact the interpolative implementations are immediate in any possible software technology (even ASM) since the interpolation networks can be directly associated to addressable memories. This way fuzzy interpolative expert systems can be implemented virtually in any software technology. Digital hardware circuits (μC, DSPs) can also implement FICs due to their memory type architecture. However the most outstanding feature of the interpolative systems is their compatibility with the analog hardware technologies. Some possible analog technologies were mentioned, such as the translinear analog CMOS [5] and even nanometric circuits [9].

In the same time, using the fuzzy theoretical perspective, sophisticated applications become feasible. This is the case of the *fuzzy self-adaptive interpolative controllers* FSAIC [5, 6].

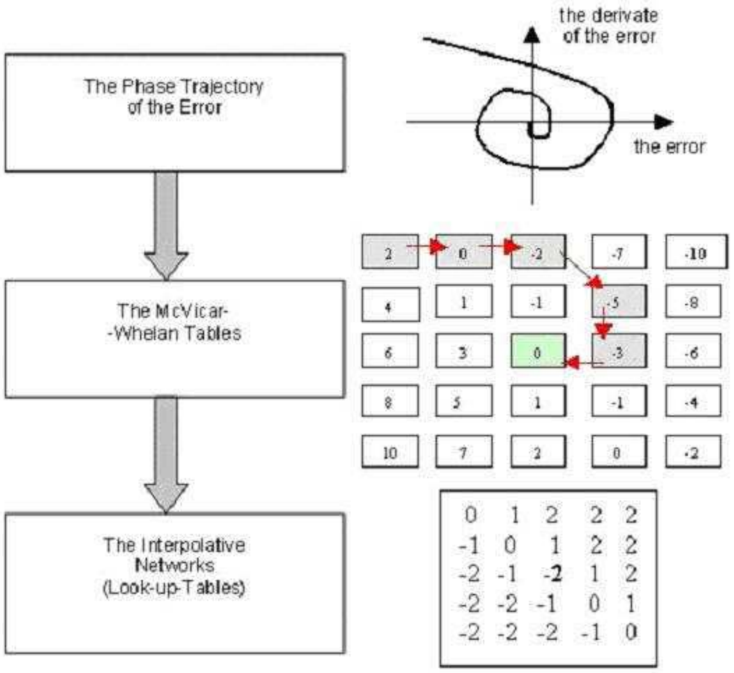
In close loop control applications FICs are perfectly matching a fundamental time analyze tool: *the phase trajectory of the error* and their specific analyze method, the qualitative analyze. We can rely on the figure 1 succession of theoretical tools that are involved into the conception, the development and the implementation of FICs.

The FIC's conception, development and implementation can be achieved by a set of operations that will be generically called *the fuzzy-interpolative methodology* FIM. FIM is taking advantage of both linguistic and interpolative nature of the fuzzy systems, combining the advantages of their both sides:

a) The fuzzy sets and fuzzy logic theory will be applied during the *conception* and the *development* stages of the control algorithms;

b) The linear interpolations based methods will ensure the *implementation* stage. The steps of the FIM are the following [2, 4, 7]:

- **a1)** the identification of the control solution;
- **a2)** the building of the control rule base of the corresponding fuzzy expert system, represented by McVicar-Whelan tables, in a linguistic manner;
- **a3)** the designing of the Sugeno controller with triangular fuzzy partitions, prod-sum inference and COG defuzzyfication, equivalent to the fuzzy expert system;
- **b1)** the designing of the corresponding look-up table with linear interpolations;
- **b2)** the implementation of the look-up table;



the derivate of the error
The Phase Trajectory of the Error
the error
The McVicar--Whelan Tables
The Interpolative Networks (Look-up-Tables)

2	0	-2	-7	-10
4	1	-1	-5	-8
6	3	0	-3	-6
8	5	1	-1	-4
10	7	2	0	-2

0	1	2	2	2
-1	0	1	2	2
-2	-1	-2	1	2
-2	-2	-1	0	1
-2	-2	-2	-1	0

Figure 1: The theoretical tools that are supporting the fuzzy-interpolative methodology

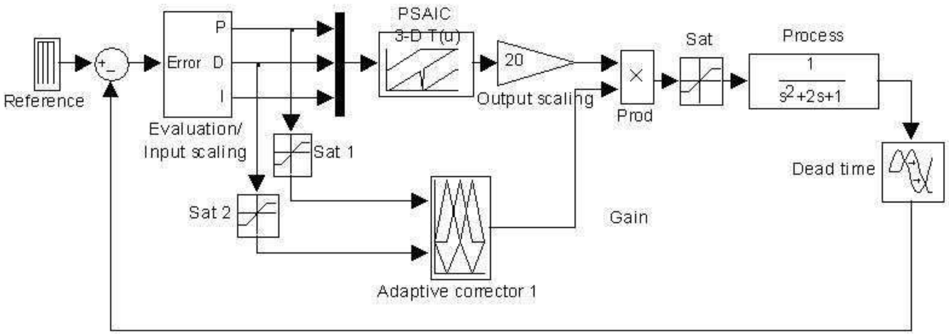
The pragmatic finality of FIM is that for simulating and implementing a large variety of fuzzy systems we do not necessarily need specific software and/or hardware. The same results can be cheaply and quickly obtained using just look-up-tables and common use controllers or DSPs.

FIM is a typical *time analysis method*, although knowledge acquired by frequency analysis methodology (the Nyquist stability criterion for instance) may be handled. From the AI perspective, since FIM is only a particular fuzzy technique, it should be classified as belonging to the Soft Computing.

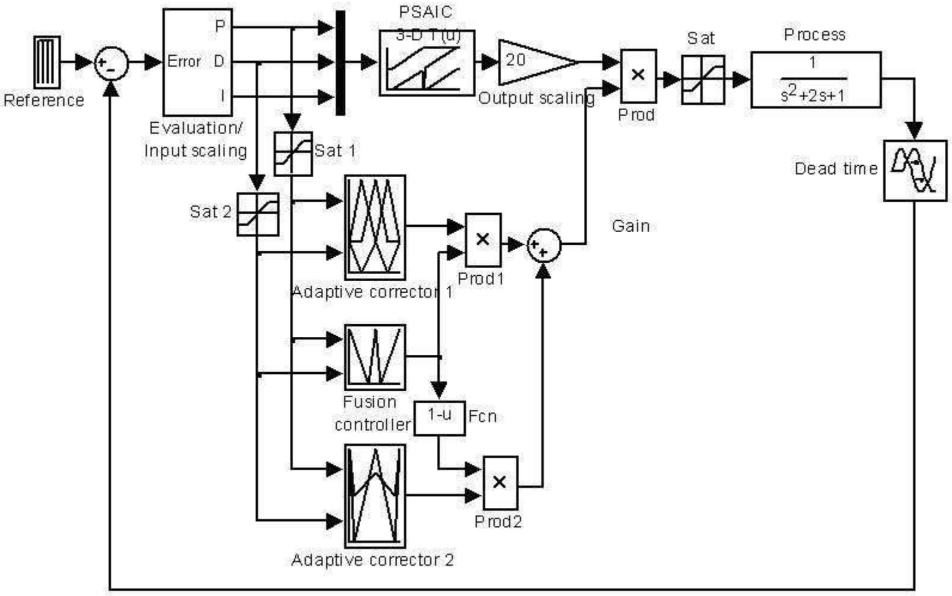
Due to its heuristic side FIM is the counterpart of a numerical algorithm: it has a low specificity but a high generality.

3 The Fuzzy Self Adaptive Interpolative Controllers

J.J. Buckley launched the paradigm of the *universal controller*, that could control a wide class of processes without any manual adjustments. Aiming to approach such an ideal structure, the family of *fuzzy self adaptive interpolative controllers* FSAIC presented in fig. 2 was introduced in references [5] and [6].



a. The FSAIC architecture



b. A Fusioned Fuzzy Self Adaptive Interpolative Controller

Figure 2: The theoretical tools that are supporting the fuzzy-interpolative methodology

FSAIC has a variable structure. During transient regimes the main controller is a

PD one (a 2D look-up-table). Its control surface is almost plane, in order to avoid the distortion of the phase trajectory of the error. During the steady regime an integrative effect is gradually introduced, the structure becoming a PID one. This functionality is achieved with a 3D look-up table having as inputs the control error ε, its derivate ε' and its integrative $\int\varepsilon$. The different PD tables that are creating the $\int\varepsilon$ dimension differ only at the central rule, that is activated when $\varepsilon = zero$ and $\varepsilon' = zero$. Thus the integrative effect is gradually activated, only when steady regimes occur. This controller is called *plane surface adaptive interpolative controller* PSAIC.

The adaptive feature that is creating the FSAIC is introduced by a PD FIC corrector that is acting by mean of a multiplicative correction factor *Gain*.

What is important for our issue is that the design of the adaptive corrector includes a set of *general knowledge on linear PID controllers' adjustment* and on *linear systems' stability*.

The FSAIC operation relies on the qualitative analyze of the phase trajectory of the error. The strategy is to push the phase trajectory towards $\varepsilon = zero$ and $\varepsilon' = zero$ point, in a sliding mode like manner. The tactic is to maintain the phase trajectory into quadrants II or IV as long as possible and to avoid quadrants I and III. This task is performed by PSAIC that also needs adaptive capabilities in the case of highly nonlinear or time varying plants. The relevant operating regimes that might occur (transient, steady, oscillating and unstable) need specific tunings of the controller. Their on-line identification can be performed by detecting the activation of the rules that are the most relevant signatures of each regime. Thanks to the tabular organization of the rule base (McVicar-Whelan) this operation is simple and can be related to the phase trajectory of the error, as shown in fig. 3.

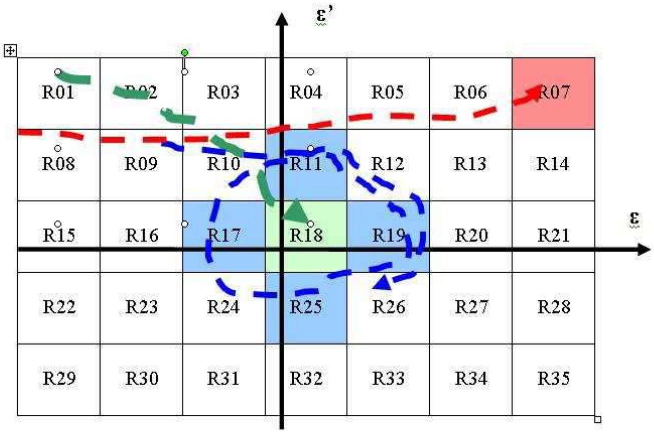

ε'
R01 R02 R03 R04 R05 R06 R07
R08 R09 R10 R11 R12 R13 R14
R15 R16 R17 R18 R19 R20 R21
R22 R23 R24 R25 R26 R27 R28
R29 R30 R31 R32 R33 R34 R35
ε

Figure 3: The most significant signatures of the operating regimes in terms of phase trajectory and their corresponding control rules, in a 7 x 5 rule base

The most relevant situations are the following:

1. if $\varepsilon = zero$ and $\varepsilon' = zero$ the regime is *steady* and the gain is *great*
   This rule, R18, is identifying the installation of the steady regime and its effect is to increase the action of PSAIC. This way the control precision is increasing, as well as the sensitivity, for the best possible rejection of the minor perturbations.

2. if $\varepsilon = zero$ and $\varepsilon' = medium$ or $\varepsilon = medium$ and $\varepsilon' = zero$, as for the rules R11, R17, R19 and R25, the regime is *transitory* or *oscillatory* and the gain is *medium* This situation may appear either in oscillatory regimes or when overshoots are producing, in both situations the gain must be decreased.

3. 3. if $sign(\varepsilon \cdot \varepsilon') > 0$, as for R07, the regime is *unstable* and the gain is *small*; The system is now firmly installed into quadrants I or III and the best measure against this situation is to reduce the gain at a minimum value, according to the Nyquist stability criterion.

This way system theory knowledge on identification, correction and stability can be embedded, constituting essential pieces of WK for the closed loop controllers. According to our experience adaptive nonlinear PID controllers can cope to almost any technical application if provided with self-adaptive algorithms with relevant WK. An interesting conclusion of this approach is that *instead of developing new control laws we should better concentrate on how WK can be embed into nonlinear PID controllers and to select the relevant pieces of knowledge that are worse to be considered.*

The FIC's capability to embed and to process knowledge is explained by the expert system side of any fuzzy system. As detailed in [11] and some related papers, using FIM in the expert system design generates *fuzzy-interpolative expert systems*, able to cope with the linguistic nature of WK.

We are presenting only one illustration of the operation, in Fig. 4, where the same FFSAIC is able to control a large variety of plants, some of them unstable for an usual linear controller [5, 6].

4 The Planned Fuzzy Interpolative Controllers

Besides the fundamental system theory knowledge that is governing all the control theory and that is useful for any controller, each application has its own features that are personalizing it. Keeping in sight the specific details of an application may make the difference between success and failure.

Trying to find techniques that are compatible with FIC and allow the representation of the specific WN concerning the controlled process, we have so far considered the *internal models* and the *planners*.

The internal functional models would be the ideal solution for this problem, but unfortunately their implementation in the industrial on-line control is yet very difficult because of the high computational demands. On the other hand, the planned systems in the sense of ref. [12], that are also solving the specific knowledge handling problem, may be fully compatible with FICs if realizable with mappings or look-up-tables.

The planning systems can harmonize a controller to each specific process if the planners' design is assisted by *functional computer models of the processes.*

The structure of a world knowledge embedding planned controller is presented in fig. 5. The WN fuzzy-interpolative expert system is common to all the applications while the planner is personalized. Although the nonlinear PID controller could be realized in any possible technology, the fuzzy-interpolative is the first choice, in order to perfectly match the adaptive part.

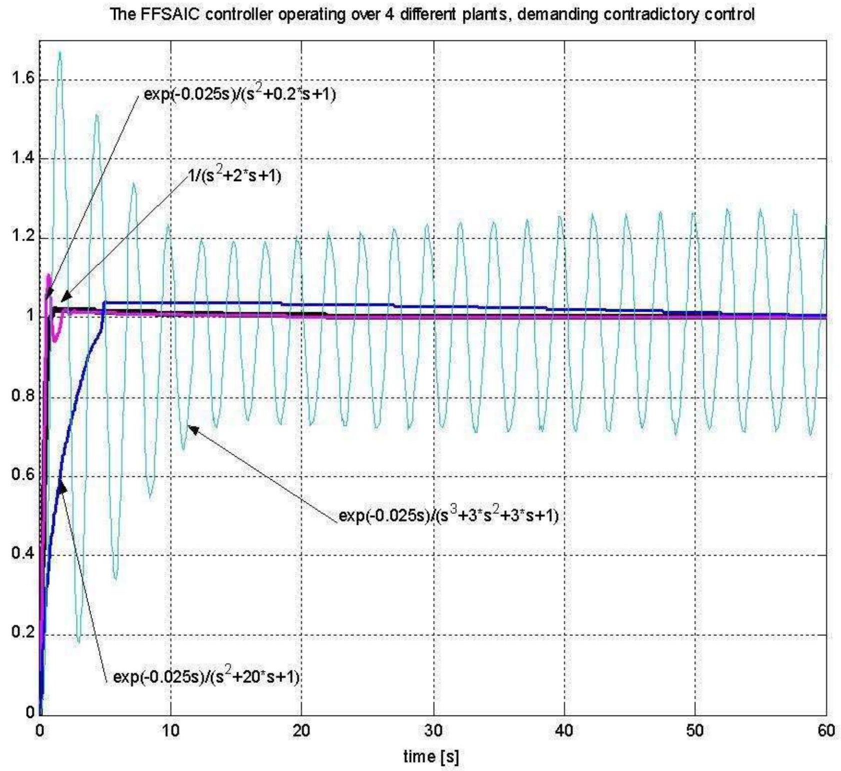


Figure 4: The FFSAIC with the on-line identification of the operating regime.

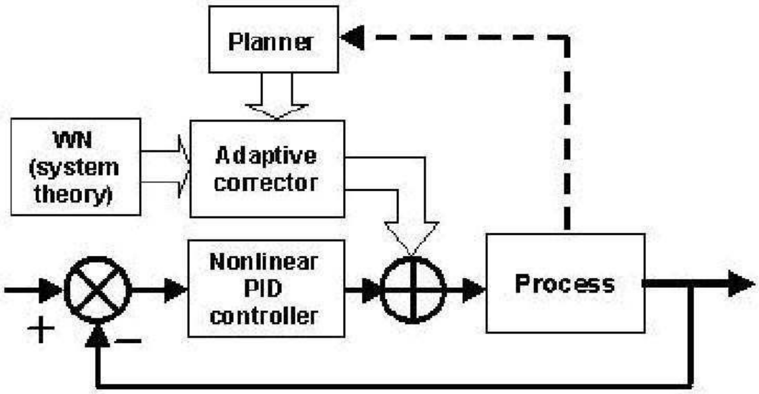


Figure 5: The world knowledge embedding planned fuzzy-interpolative controller

Some WK topics that were used so far by our team in different works are: technical data about vehicles, sensors, psychological behavior, physiology (humans and plants), etc. A case study is presented in order to illustrate this technique.

5 The imposed distance braking

Using the *Imposed Distance Braking Method* IBDM for railway coaches or urban vehicles, the braking action is following the basic rule "*STOP at distance d from here*". During IBDM the resources of the vehicle (the braking force, the friction of the disk brake system, the adherence between the wheels and the rails) are used in an optimal and smooth manner. The effort is constantly distributed along the braking distance, avoiding the strong variations or even the shocks that are inherent to the conventional sequential braking algorithms [14].

The main role in the braking process is played by a *planned position controller*. The controller's input is fed by an imposed program of the vehicle's position that is following the natural evolution in which the same vehicle would brake under the constraint of a constant braking force. This positioning program is implemented by means of a position-velocity mapping, as shown in fig. 6. This mapping is obtained with the help of a computer model of the railway vehicle. The experience achieved using different simulation parameters indicates that the particular parameters of the mapping, (the initial velocity and the overall braking distance) are not critical.

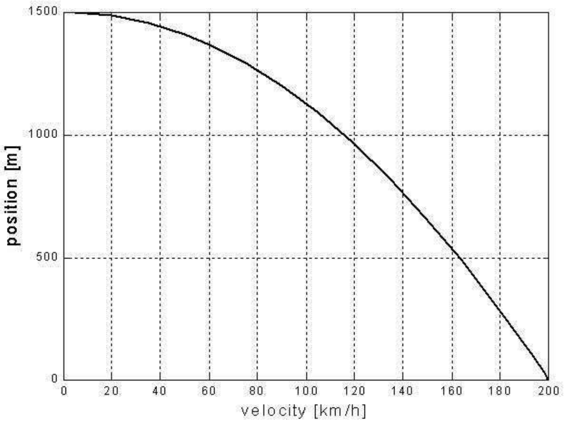


Figure 6: The position-velocity mapping of a railway car

The position-velocity mapping $p(v)$ is connected to the controller by means of two coefficients: the *initial velocity coefficient* k_v and the *distance coefficient* k_d as shown in Fig. 7. The coefficients are able to actualize the $p(v)$ mapping for the initial velocity of the vehicle at the beginning of the braking and to the desired braking distance.

The typical behavior of the IMDM brakes is illustrated in Fig. 8 for 80km/h initial velocity and 120m imposed braking distance. One observe that after the initial error peak caused by the inertia of the pneumatic braking cylinder, the vehicle is smoothly following the desired $p(v)$ planner.

This way no violent actions are needed and the ABS is not activated, which would not be the case, if we would have being applied a simplified linear position velocity planner, as is the case for many actual sequential braking controllers.

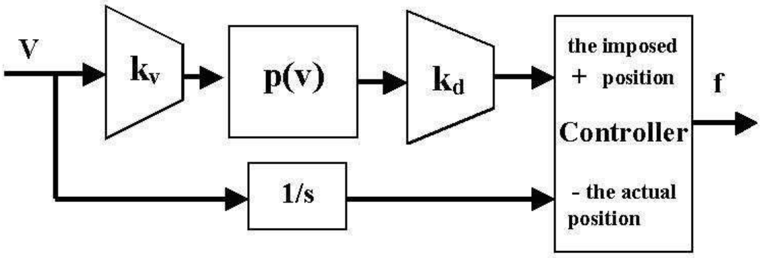


Figure 7: The planned position controller

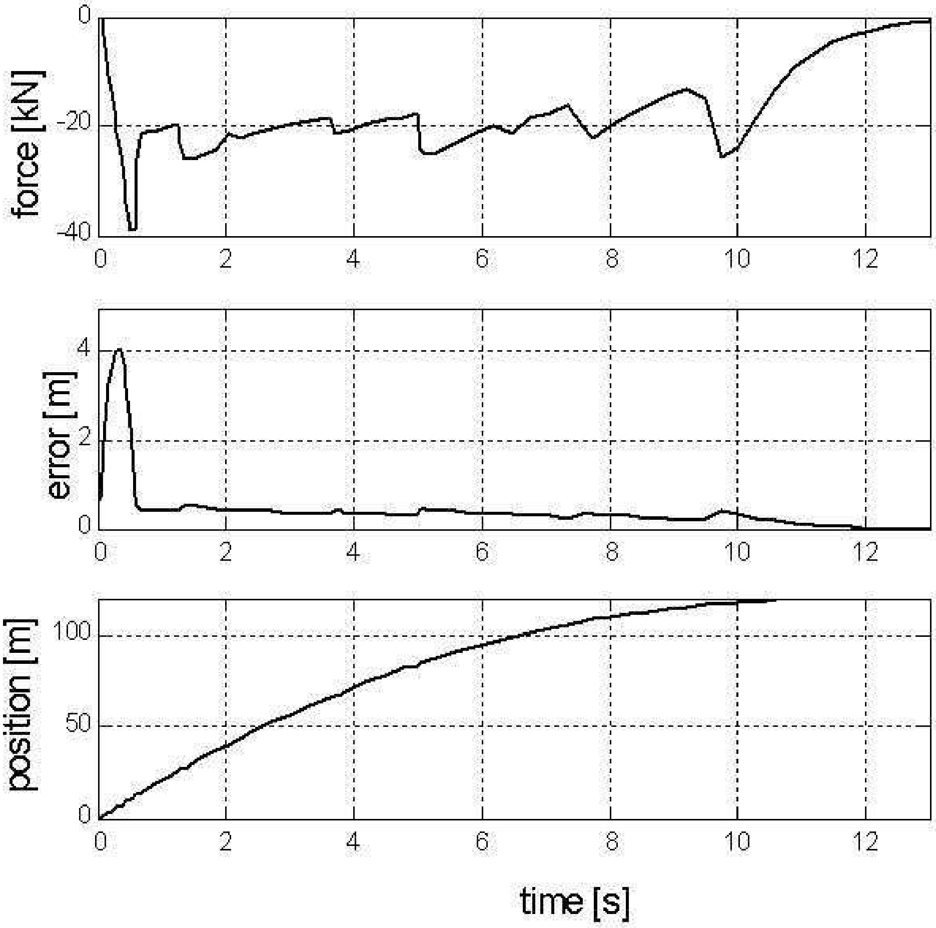


Figure 8: An IBDM braking

6 Conclusions

The world knowledge represents a fundamental resource for the future close loop controllers. If embedded into control algorithms, the general knowledge on the system theory as well as the specific knowledge on the controlled process is able to significantly improve the control performance in any possible sense (precision, robustness, speed, smoothness, etc.) A fundamental theoretical and applicative tool that enables us to provide low level computing devices - μCs, DSPs, etc. with WK is the planned fuzzy-interpolative controller.

References

[1] Lotfi A. Zadeh. From Search Engines to Question-Answering Systems - The Problems of World Knowledge, Relevance, Deduction and Precisiation, *Invited Conference*, SOFA'05, Arad, Romania, August 30, 2005.

[2] M. M. Bălaş, V. E. Bălaş. World Knowledge for Applications by Fuzzy-Interpolative Systems. *International Journal of Computers, Communications and Control, Vol III, May*, pp. 28-32, 2008.

[3] L. T. Kóczy, K. Hirota, Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, no. 71, pp. 169-201, 1993.

[4] L. T. Kóczy, K. Hirota, Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, no. 9, pp. 197-225, 1993.

[5] M. Bălaş. Regulatoare fuzzy interpolative. *Politehnica* Eds., Timisoara, 2002.

[6] M. M. Bălaş, V. E. Bălaş. The Family of Self Adaptive Interpolative Controllers. *Proc. of Information Processing and Management of Uncertainty in Knowledge-Based Systems* IPMU'04, Perugia, July, 2004, pp. 2119-2124.

[7] L. T. Kóczy, M. M. Bălaş, M. Ciugudean, V. E. Bălaş, J. Botzheim. On the Interpolative Side of the Fuzzy Sets. *Proc. of the IEEE International Workshop on Soft Computing Applications* SOFA'05, Szeged-Arad, 27-30 Aug. 2005, pp. 17-23.

[8] M. Bălaş, V. Bălaş. World Knowlwdge for Control Applications. *Proc. of 11th International Conference on Intelligent Engineering Systems* INES 2007, Budapest, 29 June - 1 July, 2007, pp. 225-228.

[9] M. Bălaş, V. Bălaş. Another Possible Way towards the Intelligent Nano-Systems the Fuzzy-Interpolative. *Proc. of Information Processing and Management of Uncertainty in Knowledge-Based Systems* IPMU'06, Paris, July, 2006, 2135-2141.

[10] V. Bălaş, M. Bălaş. Observing Control Systems by Phase Trajectory of the Error. *WSEAS Transactions on Systems.* Issue 7, vol. 5, July 2006. pp. 1717-1722.

[11] M. Bălaş. Le flou-interpolatif, present et perspectives. *Seminaire LSIS St.-Jerome*, Marseille, France, 21 sept. 2006.

[12] K. M. Passino, P. J. Antsaklis. Modeling and Analysis of Artificially Intelligent Planning Systems. *Introduction to Intelligent and Autonomous Control, by P.J. Antsaklis and K.M. Passino*, Eds., Kluwer, 1993, pp. 191-214.

[13] M. Bălaş, V. Bălaş, J. Duplaix. Optimizing the Distance-Gap between Cars by Constant Time to Collision Planning. *Proc. of IEEE International Symposium on Industrial Electronics* ISIE 2007, June 2007, Vigo, pp. 304-309.

[14] M. M. Bălaş, V. E. Bălaş, C Barna. The Constant Effort Imposed Distance Braking for Urban Railway Vehicles. EUROCON 2005 - *The International Conference on "Computer as a Tool"*, Belgrade, Serbia and Montenegro, November 21-24, 2005, pp. 365-368.

Marius M. Bălaş, Valentina E. Bălaş
Aurel Vlaicu University of Arad, Romania
E-mail: balas@inext.ro

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 22-31

A Survey on the Constant Time to Collision Techniques

Valentina E. Bălaş and Marius M. Bălaş

Abstract: The paper is presenting a new method for the management of the traffic flow on highways, based on the constant time to collision criterion. The criterion is applied for each car implied in traffic, and for the whole highway. Each car is provided with a constant time to collision cruise controller, that is maintaining optimal distance-gaps between cars, adapted to the speed and to the technical data of the cars. The highway's traffic management center has the possibility to impose the same time to collision to all the cars. This way the traffic is organizing itself, by distributing the cars such way that the collision risk is uniformly distributed. Simulations are illustrating how the cars are behaving when they are forming highway platoons and how the traffic flow may be controlled by imposing the time to collision.

Keywords: knowledge embedding by computer models, optimal distance gap between cars, constant time to collision, fuzzy-interpolative cruise controllers.

1 Introduction

The automat driving is expected to enhance the driving performance and to reduce the crash risks. The Advanced Driver Assistance Systems ADAS are such systems [1], [2]. These systems can be linked to cruise control system, allowing the vehicle to slow when catching up the vehicle in front and accelerate again to the preset speed when the traffic allows. A key problem in this issue is the control of the distance gap be-tween cars. In some previous papers [3], [4], [5], we introduced a fuzzy-interpolative distance-gap control method that is using a Constant Time to Collision Planner CTCP, in the sense of the Planning System concept [6]. This approach was also dis-cussed in ref. [7]. The present work is presenting a survey on the Constant Time to Collision criterion CTTC and its application in the domain of the traffic management. A model of a CTTC platoon is discussed. The simulations are focused on the way in which the cars are forming the platoons, and on the relationship between the imposed CTTC and the traffic intensity. An association of ideas can be established with the new domain of the swarm systems, where the CTTC criterion can be used as swarm's aggregation law.

The paper is synthesizing the results of the first researches performed on the CTTC techniques that are already communicated [4, 5, 13, 15, 16], in order to offer a new impulse for further developments and successful applications.

2 The Constant Time to Collision Criterion

Several indicators measure the characteristics of the traffic flow: the Time-to-Collision TTC, the Time-to-Accident, the Post-Encroachment-Time, the Deceleration-to-Safety-Time, the Number of Shockwaves, etc. [1], [2]. TTC is the time before two following cars (Car2 is following Car1) are colliding, assuming unchanged speeds of both vehicles:

$$TTC = \frac{d}{v_2 - v_1} \tag{1}$$

TTC is linked to the longitudinal driving task. Negative TTC implies that Car1 drives faster, i.e. there is no danger, while positive TTC is leading to unsafe situations. By assessing TTC values at regular time steps or in continuous time, a TTC trajectory of a vehicle can be determined. Doing this for all vehicles present on a road segment one can determine the frequency of the occurrence of certain TTC values, and by comparing these distributions for different scenarios, one can appreciate the traffic safety [2].

The central issue in cars' safety is to impose an appropriate distance between cars, d_i. The Autonomous Intelligent Cruise Control AICC is imposing a particular polynomial $d_i(v_2)$ law:

$$d_i(v_2) = z_0 + z_1 \cdot v_2 + z_2 \cdot v_2^2 = 3 + z_1 \cdot v_2 + 0.01 \cdot v_2^2 \tag{2}$$

Several settings are recommended, for example $z_1 = 0.8s$ or $z_1 = 0.6s$. Two objections can be drawn against this polynomial $d_i(v_2)$ law:

- no effective adaptation to the traffic intensity is offered: if (3) is tuned for intense traffic, when the traffic is decreasing, the following cars will continue to maintain the same short distance-gaps between them. The driving rules used on highways today are even weaker: "keep distance above 100m" for instance.

- z_1 and z_2 are artificially introduced parameters, they have no significance for humans - highway operators or drivers - and they are not linked to the physical features of the system.

The *Constant Time to Collision* criterion CTTC consists in imposing *stabilized* TTCs by means of the Car2 cruise controller.

The on-line TTC control is not convenient because when the two cars have the same speed the denominator of TTC is turning null: $v_2 - v_1 = 0$. That is why CTTC must be implemented off-line, with the help of $d_i(v_2)$ mappings. The CTTC implementation by $d_i(v_2)$ distance-gap planners is possible because *a distance gap planner using* TTC *will produce* CTTC. We studied this method by computer simulations, using a Matlab-Simulink model of the tandem Car1-Car2, introduced in other previous papers [3, 4, 5, 9, 13] and presented in Fig. 1.

Since the design of the planners is performed with the help of functional models of the cars, accurate knowledge about the specific behavior and parameters of each car (traction and braking forces, weight, aerodynamic coefficient, etc.) can be taken into account, which is not possible to the simplified and leveling analytic model (2).

The application of this method imposes the car manufacturers to provide each type of automobile with a functional computer model. However this investment could be very useful as well for other purposes, say for instance the automate diagnosis.

The distance-gap planners are designed as follows. The simulation scenario consists in braking Car1 until the car is immobilized, starting from a high initial speed. A TTC controller is driving the Car2 traction/braking force such way that during the whole simulation TTC is stabilized to a desired constant value. A linear PID controller can be

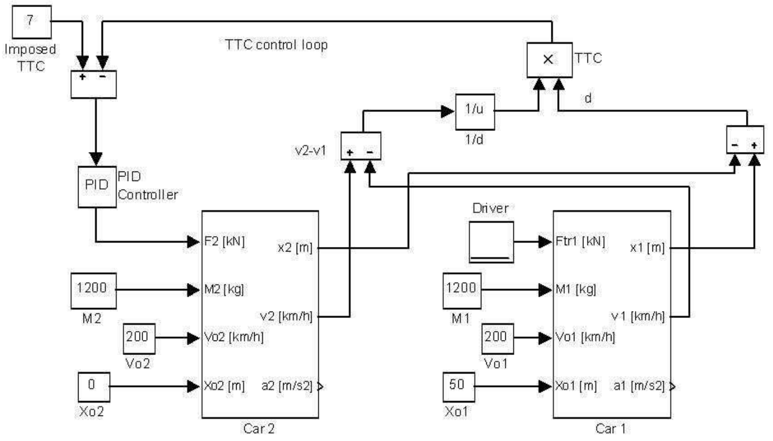


Figure 1: The SIMULINK-MATLAB model of the tandem Car1-Car2

used for this. The continuous braking allow us to avoid the $v_2 - v_1 = 0$ case. We will use the recorded d mapping as the desired $d_i(v_2)$ planner for the given TTC.

The Fig. 2 planners are determined for three TTC values: 4, 7 and 10s. They can be easily implemented with the help of the look-up tables with linear interpolation.

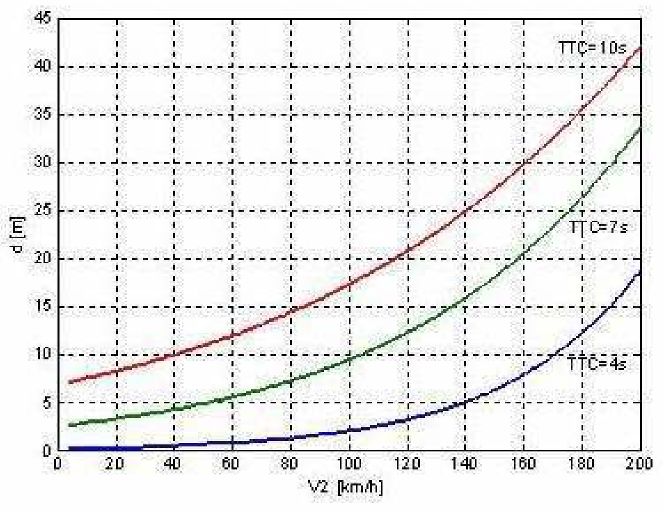


Figure 2: The recorded $d_i(v_2)$ mappings for three different TTC

The use of the CTTC planning technique is essentially facilitating the task of the distance controller that is actually driving the traction/braking force of a real car during the cruise regime, as shown in Fig. 3. Very simple fuzzy-interpolative PD con-trollers or even linear controllers can such way cope with the car following task [15].

The implementations can be basically achieved by look-up-table techniques.

Applying CTTC brings two obvious advantages:

- a constant distribution of the collision risk over all the vehicles involved;

- the possibility to control the traffic flow on extended road sections, if each vehicle will apply the same TTC that is currently recommended by the Traffic Management Center [14]: a long TTC means *low traffic flow and higher safety* while a short TTC means *high traffic flow* and *higher risk*.

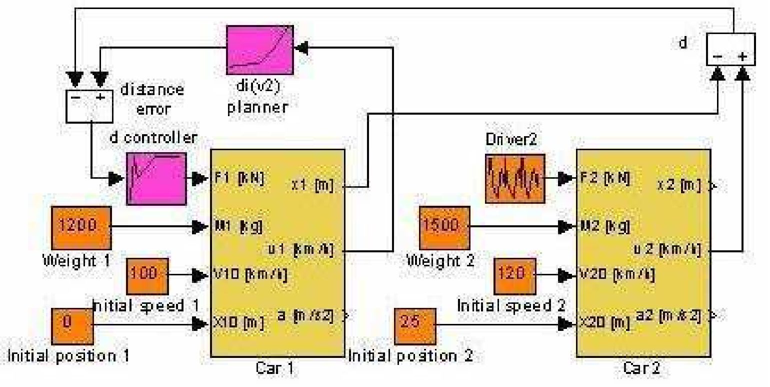


Figure 3: A cruise control system with distance controller and CTTC $d_i(v_2)$ planner

3 The Traffic Management by Constant Time to Collision

Besides the control of following paired cars, a superior application level for the CTTC criterion is the management of the highway traffic. Assuming each car provided with a cruise controller with CTTC planner, the Traffic Management Center TMC [14] has the possibility to impose the same TTC to all the cars. This way the highway system becomes a distributed one. Each car is trying to reach and to maintain the position that respects the imposed TTC to the previous car. This trend has as major advantage a constant distribution of the collision risk for each car.

Lets consider that TMC is imposing a 7s TTC. If the traffic is not too intense, the tendency of the cars will be to form platoons that are able to maintain TTC=7s. It is to remark that the distance between the cars belonging to the same platoon are not necessarily identical, even for constant speeds, because each type of cars has its particular $d_i(v_2)$ planner, in accordance to its technical parameters (weight, aerodynamics, engine power, brakes, etc.) If the traffic is beginning to decrease the number of the cars that are included into platoons will decrease too, and empty zones will develop on the highway. In this case TMC should increase the imposed TMC value, say TMC=10s. As a consequence the distance gaps between cars will increase, the disposable space of the highway will be better covered and the collision risk will decrease for each car.

In the opposite case, if the traffic is increasing, the cars will not be able to maintain the desired TTC and the corresponding distance-gaps. TMC will be forced to reduce the imposed TTC, say TMC=4s. This way the density of the traffic will increase and the collision risk will increase too, but this will happen in a smooth and controlled manner, the risk continuing to be equally distributed over each car.

Our research on this matter is only at the initial stage, but the preliminary simulations are confirming that CTTC criterion is potentially able to cope with the highway traffic.

4 A CTTC platoon model

The CTTC platoons are highway car formations composed by automobiles provided with CCTC cruise controllers. In some previous papers, namely [15], a five cars CTTC platoon was simulated. The Fig. 4 Simulink-Matlab model, addressed to a five car group, was used. Each car has its own technical parameters: weights between 1000 and 1400kg (variables M), engine powers between 100% and 180% of the generic Car1 power (variables Gain) and its own CTTC planner (see the $D_i(V_2, TTC)$ look-up-tables). The initial speed v_o and position x_0 of each car can be as well adjusted. The model is offering the time variation of the aimed parameters: speeds and positions of each car, distance-gaps between cars, the length of the platoon, etc.

The next figures are illustrating the behavior of this platoon, for a generic simulation scenario, presented in Fig. 5. The scenario is imposing plausible variations of the speed and of the imposed TTC. It is to remark the notable TTC steps that appear for $t = 470$ and $t = 500s$, that has the purpose to test the dynamics of the CTTC cruise controllers. Such fast variations of the imposed TTC should not appear during the usual exploitation.

The first simulation, presented in Fig. 6, is illustrating the global behavior of the five car TTC platoon, for TTC=7s. One can also observe the continuous variation of the platoon's length with the speed that is presented in Fig. 7.

In figures 8 and 9 another simulation, executed for TTC=15s, details the formation of the platoon. One can observe the behavior of the five cars that are starting separated with 10m, and are forming the platoon in less than 10s, with no notable errors or oscillations. This behavior was obtained with the help of a nonlinear fuzzy-interpolative PD cruise controller [16], that is superior to the linear PD [15].

Of course, the maneuvers that are needed to enter into or to exit out of a high speed platoon need a careful attention and serious experimental tests.

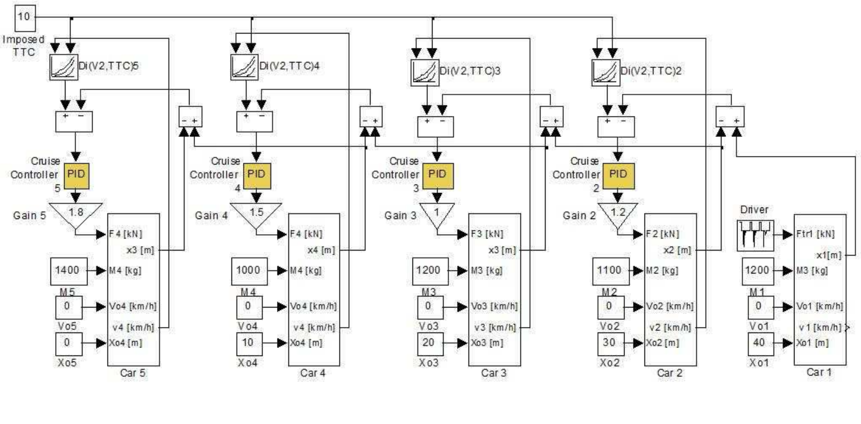


Figure 4: The SIMULINK-MATLAB model of a five car CTTC platoon

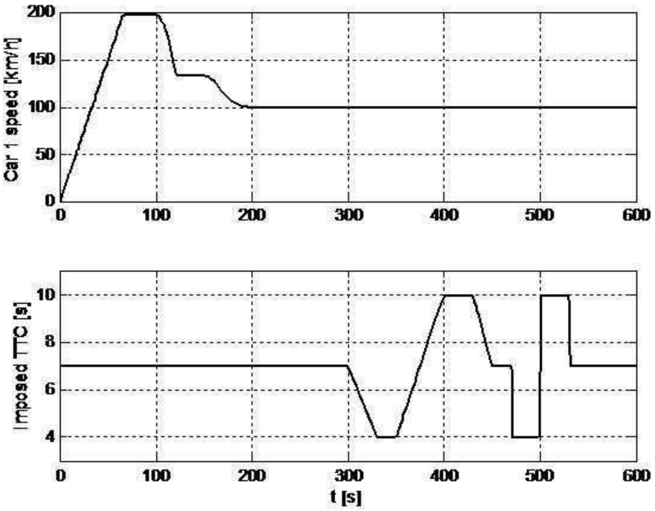


Figure 5: A CTTC platoon simulation, with v_1 and the sum of the distance gaps

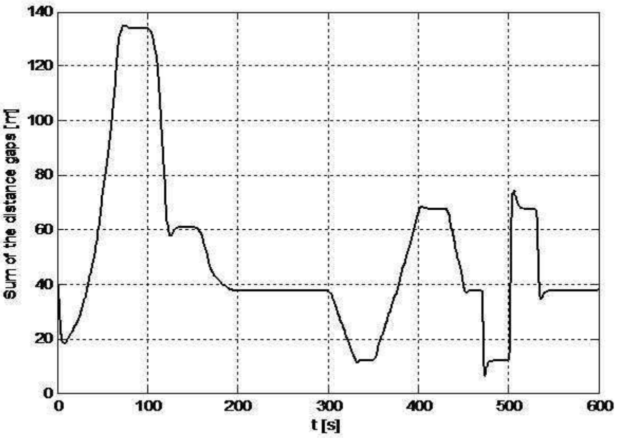


Figure 6: The sum of the distance gaps for the previous simulation

5 The PD fuzzy-interpolative cruise controller

The CTTC cruise controller is an minimal PD interpolative one, as shown in Fig. 10.

The 2D look-up-table that is implementing the controller is the following:

$$Row(distanceerror) \ : \ [-10-50510] \tag{3}$$

$$Column(errorderivate) \ : \ [-10010] \tag{4}$$

$$Output \ : \ [-1-1-1;-1-0.30;-0.200.2;00.31;111] \tag{5}$$

This controller is extremely simple and it has multiple tuning options: the look-up-table values, as well as the input and output scalar factors.

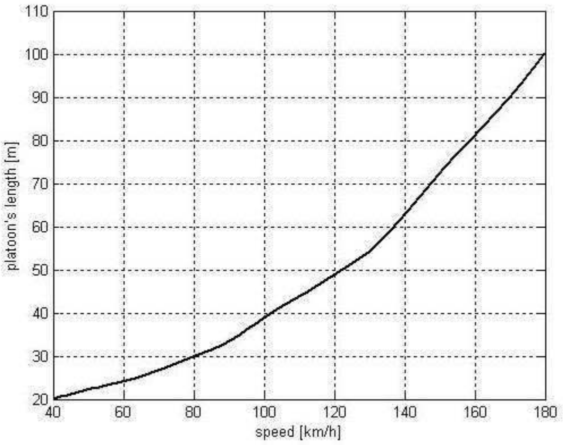


Figure 7: The dependence with the speed of the sum of the distance gaps, for TTC=7s

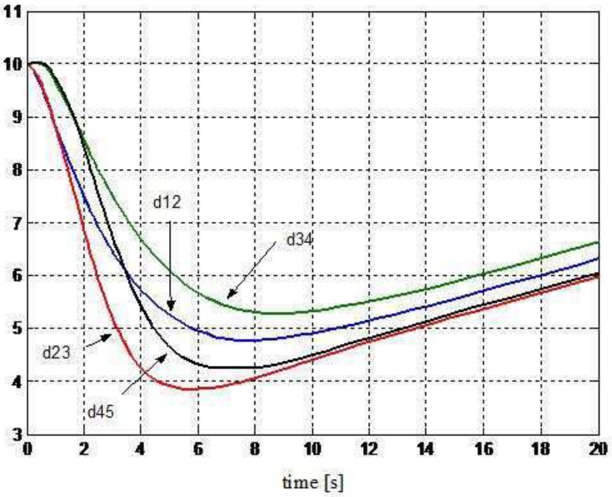


Figure 8: The platoon's aggregation

6 Driver assisting by Inverse Time to Collision

In a previous paper [5] we introduced the inverse of TTC: *The Inverse Time to Colli-sion* TTC^{-1}:

$$TTC^{-1} = \frac{v_2 - v_1}{d} \tag{6}$$

Besides avoiding the disadvantage of the infinite value when $v_2 = v_1$, this index is directly proportional with the collision risk: the higher TTC-1 is the higher will be the risk. Negative $TTC^{-1}s$ have the same significance as negative TTCs. The neighbor-hood of $TTC - 1 = 0$ is corresponding to the TTC's saturation so it is not sensitive.

TTC^{-1} can be used in the driver assistance, as shown in Fig. 11 [5].

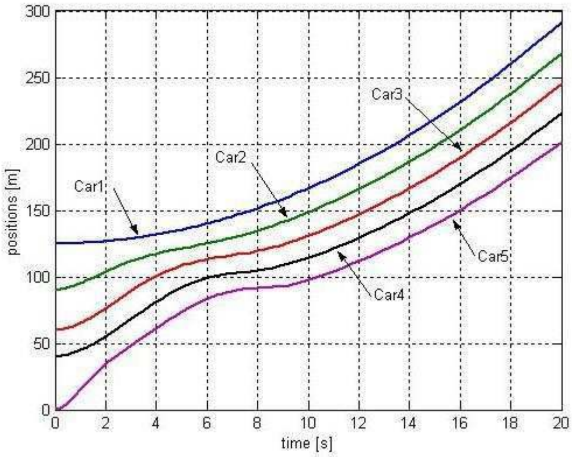


Figure 9: The positions of the cars during the platoon's aggregation

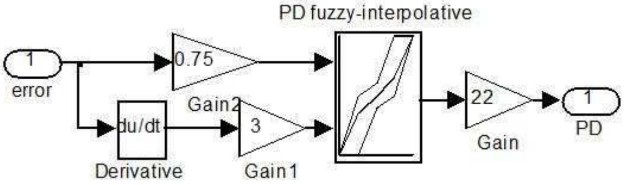


Figure 10: The PD fuzzy-interpolative cruise controller

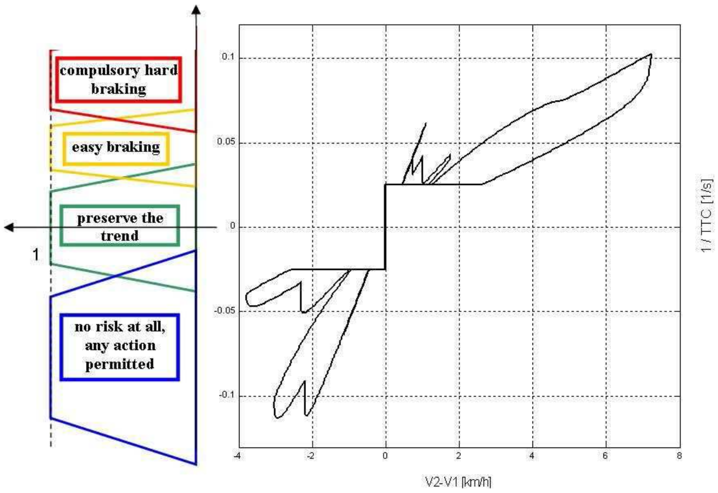


Figure 11: A $TTC^{-1}(v_2 - v_1)$ trajectory and a corresponding fuzzy partition able to assist the driver

7 Conclusions

The time to collision criterion can be used in the highway traffic management. If each car is provided with a constant time to collision cruise controller, the traffic management center can impose the same time to collision to all the cars. Such way the highway system becomes a distributed one, each car trying to reach and to maintain the position that respects the imposed time to collision to the previous car. The method keeps constant the collision risk for over all the cars of the highway. Besides the simplicity and the advantageous interpolative implementation, all the time to collision based tools have a common feature: they are embedding precise knowledge about the technical data of the automobiles thanks to the functional computer model that stands behind their design. This adaptive capability is promising to improve the future highway traffic. The inverse time to collision is an index of the collision risk and may assist the human driver.

References

[1] A.R. Girard, J. Borges de Sousa, J.A. Misener and J. K. Hendrick. A Control Architecture for Integrated Cooperative Cruise Control and Collision Warning Systems. *Berkeley Uni-versity of California*, http://path. berkeley.edu/ anouck/papers/cdc01inv3102. pdf.

[2] M.M. Minderhoud and S.P. Hoogendoorn. "Extended Time-to-Collision Safety Measures for ADAS Safety Assessment", *Delft University of Technology*, http://www. Delft 2001.tudelft. nl/paper %20 files/ paper1145. doc

[3] M. Balas, C. Barna. Using CCD cameras for the car following algorithms. *Proc. of IEEE International Symposium on Industrial Electronics* ISIE'05, Dubrovnik, 20-23 June, 2004, pp. 57-62.

[4] M. Balas, V. Balas. Optimizing the Distance-Gap between Cars by Fuzzy-Interpolative Control with Time to Collision Planning. *Proc. of The IEEE International Conference on Mechatronics*, Budapest, Hungary ICM'06 Budapest, 3-5 July, 2006, pp. 215-218.

[5] V. Balas, M. Balas. Driver assisting by Inverse Time to Collision. *Proc. of The IEEE International Conf. on Mechatronics*, Budapest, Hungary WAC'06 Budapest, 24-27 July, 2006.

[6] K.M. Passino, P.J. Antsaklis. Modeling and Analysis of Artificially Intelligent Planning Systems. *Introduction to Intelligent and Autonomous Control*, by P.J. Antsaklis and K.M. Passino, Eds., Kluwer, 1993, pp. 191-214.

[7] M. Balas. Le flou-interpolatif, present et perspectives. *Seminaire LSIS St.-Jerome*, Marseille, France, 21 sept. 2006.

[8] M. Balas. Regulatoare fuzzy interpolative. *Editura Politehnica Timisoara*, 2002.

[9] L.T. Kóczy, M.M. Balas, M. Ciugudean, V.E. Balas, J. Botzheim. On the Interpolative Side of the Fuzzy Sets. *Proc. of the IEEE International Workshop on Soft Computing Applications* SOFA'05, Szeged-Arad, 27-30 Aug. 2005, pp. 17-23.

[10] R.E. Precup, S. Preitl, M. Balas, V. Balas. Fuzzy Controllers for Tire Slip Control in Anti-Lock Braking Systems. *Proc. of The IEEE International Conference on Fuzzy Systems* FUZZ-IEEE'04, Budapest, 25-29 July, 2004, pp. 1317-1322.

[11] M. Balas. World knowledge for controllers. *International Symposium Research and Education in Innovation Era*, Arad, 16-18 Nov. 2006, Section 3, pg. 531-534.

[12] Y. Zhang, E.B. Kosmatopoulos, P.A. Ioannou, C.C. Chien. Autonomous Intelligent Cruise Control Using Front and Back Information for Tight Vehicle Following Maneuvers. *IEEE Trans. on Vehicular Technology*, vol. 48, no. 1, January 1999, pp. 319-328.

[13] M. Balas, V. Balas, J. Duplaix. Optimizing the Distance-Gap between Cars by Constant Time to Collision Planning. *Proc. of IEEE International Symposium on Industrial Electronics* ISIE 2007, June 2007, Vigo, pp. 304-309.

[14] ITS Decision. Traffic Management Centers http://www.calccit.org/itsdecision/serv_and_tech/Traffic_management/TMC/tmc_summary.html

[15] V.E. Balas and M.M. Balas. Constant Time to Collision Platoons. Int. J. of Computers, Communications & Control, Vol. III, Suppl. Issue, Oradea, pp. 33-39.ITS Decision. Traffic Management Centers http://www.calccit.org/itsdecision/serv_and_tech/Traffic_management/TMC/tmc_summary.html

[16] 16. V.E. Balas and M.M. Balas. The Traffic control by constant time to collision platoons. The 11th Mini Conference on Vehicle System Dynamics, Identification and Anomalies, Budapest, 10-12 Nov. 2008.

Valentina E. Bălaş, Marius M. Bălaş
Aurel Vlaicu University of Arad, Romania
E-mail: balas@inext.ro

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 32-50

Non-negative Matrix Factorization Methods and their Applications

Ioan Buciu and Ioan Nafornită̆

Abstract: Nonnegative Matrix Factorization (NMF) is a relatively recent method used to decompose a given data set into two nonnegative more or less sparse factors. The computer vision scientists have shown an impressive interest for NMF in recent years, leading to a large number of scientific papers in this field. Since its first original development, the method had suffered various modifications and improvements. Most improvements addressed the sparseness issue to allow intrinsic or user based sparseness degree variation. The sparseness issue was not only considered from the feature representation point of view, but also in conjunction with other aspects, such as feature classification or compression. Some works dealt with the classification issue, modifying the standard method in order to achieve superior classification performance. Being applicable to both 1D and 2D signals, NMF and its variants have been successfully used for various applications, including image classification, chemometry, sound recognition, musical audio separation or extraction of summary excerpts from audio and video, air emission quality studies, identification of object materials from spectral reflectance data at different optical wavelengths, or text mining. This paper presents an overview of the standard NMF method along with its most representative and recent variants, followed by NMF's applications.

Keywords: Overview, nonnegative matrix factorization, feature extraction, pattern recognition.

1 Introduction

The data decomposition paradigm has multiple meanings and goals, arising from many applications. We can mention those related to data compression, transmission or storage. An important application comes from the pattern recognition field, where the purpose is to automatically cluster the data samples into distinct classes. This task usually requires the extraction of discriminant latent features from the initial data prior to classification. This preprocessing step is typically applied for increasing the classification accuracy. Data decomposition through feature extraction is an important step for high dimensional data. When applied, it removes redundant data components and, consequently, reduces data dimensionality. Working on a lower dimensionality space leads to several benefits, such as reduced computational load and possible discovery of task-relevant hidden variables.

Given a matrix $\mathbf{X}$ of size $m \times n$ whose columns contain data samples, the data decomposition task can be described by factoring $\mathbf{X}$ into two terms $\mathbf{W}$ and $\mathbf{H}$ of size $m \times p$ and $p \times n$, respectively, where $p <\min(m, n)$. The decomposition is performed so that the product $\mathbf{WH}$ should approximate as best as possible the original data $\mathbf{X}$, for $p <\min(m, n)$. Obviously, when $p = n$, a perfect data recovery is obtained. The columns of $\mathbf{W}$ are usually called basis vectors and the rows of $\mathbf{H}$ are called decomposition (or encoding) coefficients. Thus, the original data are represented as linear combinations of these basis vectors.

This chapter aims at describing a particular data decomposition approach, termed *Non-negative Matrix Factorization* (NMF), where both **W**, **H** terms have only non-negative entries. Several important variations of this technique are also described along with their applications in signal and image processing. To date, three survey papers on NMF subject exist in the literature. In the first one [1], the readers can find issues related to the NMF numerical stability and various algorithms for solving its associated cost functions, while the second survey [2] rather addresses its applications in pattern recognition. This chapter extends the work presented in the third survey [3], providing an overview of the entire domain, complements the missing parts of the three aforementioned papers, and presents novel works devoted to the NMF in terms of both mathematical extensions and applications. The chapter ends up with discussions where open problems are pointed out.

2 Matrix decomposition into non-negative factors

Formally, the NMF problem can be stated as follows:

Given a non-negative matrix $\mathbf{X}_{m\times n}$ *and a positive integer* $p < min(m,n)$*, find two non-negative matrices* $\mathbf{W}_{m\times p}$ *and* $\mathbf{H}_{p\times n}$ *that minimize the following Least Square (LS) cost function* $f(\mathbf{W},\mathbf{H})$*:*

$$f_{NMF}^{LS}(\mathbf{X},\mathbf{WH}) = \frac{1}{2}\|\mathbf{X}-\mathbf{WH}\|_F^2 = \frac{1}{2}\sum_{ij}(x_{ij}-\sum_k w_{ik}h_{kj})^2 \tag{1}$$

subject to $\mathbf{W},\mathbf{H} \geq 0$. Here F denotes the Frobenius norm, $i = 1,\dots,m$, $j = 1,\dots,n$, and $k = 1,\dots,p$. We should note that the above NMF problem is too restrictive for a general definition due to the particular cost function involved. Other error norms can be considered. However, we referred to this cost function as it was originally mentioned in the standard NMF, and is ,probably, the most intuitive error measurement. The lower decomposition rank p can be much less than either n or m, and its value is sometimes critical for certain applications. To date, no general rule exists for choosing the appropriate value prior to experiments. Rather, its value is a data-dependent issue.

Non-negative matrix decomposition can be traced back in time to 1994, when Paatero and Tapper [4] proposed a positive matrix factorization (PMF) to perform factor analysis on environmental data. The goal of their work was motivated by finding explanations of large set of experimental measurements, where a particular factor might be present having a positive effect, or it might not be present in which case it has no effect (value zero). However, the interest for this problem exploded only after Lee and Seung [5] published their work in *Nature*. Their paper describes two NMF applications. The first application contained human faces in the columns of **X**, yielding basis vectors describing facial features such as eyes, eyebrows, nose or mouth. In the second NMF application, the columns of the input matrix contained word counts from documents and the decomposition produces basis vectors corresponding to semantic text categories.

Apart from the cost function expressed in (1), Lee and Seung proposed a second cost function [6] based on Kullback-Leibler (KL) divergence [7]:

$$f_{NMF}^{KL}(\mathbf{X}\,\|\,\mathbf{WH}) \triangleq \sum_{i,j}\left(x_{ij}\ln\frac{x_{ij}}{\sum_k w_{ik}h_{kj}} + \sum_k w_{ik}h_{kj} - x_{ij}\right). \tag{2}$$

Interestingly, this KL based cost function is closely related to a work of Bregman dated 1967[8], who derived a frequently used distance measure between two positive vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}_+^m$. When these vectors are associated with a convex function $\phi : \Delta \rightarrow \mathbb{R}$, $\Delta \subseteq \mathbb{R}_+^m$, the Bregman distance between $\mathbf{a}$ and $\mathbf{b}$ is expressed as follows:

$$B_\phi(\mathbf{a} \,\|\, \mathbf{b}) \triangleq \phi(\mathbf{a}) - \phi(\mathbf{b}) - \nabla\phi(\mathbf{a})(\mathbf{a} - \mathbf{b}) \tag{3}$$

where $\nabla\phi(\mathbf{a})$ is the gradient of ϕ at $\mathbf{a}$. When $\phi(\mathbf{a}) = \sum_i a_i \ln a_i$, for $i = 1, \ldots, m$, the Bregman distance recasts into Kullback-Leibler (KL) divergence between $\mathbf{x}$ and $\mathbf{b}$, i.e.:

$$KL(\mathbf{a} \,\|\, \mathbf{b}) = \sum_i \left(a_i \ln\left(\frac{a_i}{b_i}\right) + b_i - a_i \right). \tag{4}$$

Obviously, relations (2) and (4) are equivalent for $\mathbf{b} = \mathbf{Wh}$, where $\mathbf{h}$ is a column of $\mathbf{H}$.

In order to find the factors that minimize the NMF cost function, Lee and Seung [6] utilized a technique similar to the Expectation-Maximization approach. While keeping one factor fixed, the other one is updated using a multiplicative rule. The factors are then interchanged and the procedure is employed again. This procedure is applied either for a certain number of iterations or until the factorial product approximates the original data within a specified error range. Thus, for the KL based cost function, the factors are updated at each iteration t as follows:

$$h_{kj}^t = h_{kj}^{t-1} \frac{\sum_i w_{ki} \frac{x_{ij}}{\sum_k w_{ik} h_{kj}^{t-1}}}{\sum_i w_{ik}}, \tag{5}$$

$$w_{ik}^t = w_{ik}^{t-1} \left(\sum_j \frac{x_{ij}}{\sum_k w_{ik}^{t-1} h_{kj}} h_{jk} \right). \tag{6}$$

For the Euclidean distance cost function, the NMF updating relations are:

$$h_{kj}^t = h_{kj}^{t-1} \frac{\sum_i w_{ki} x_{ij}}{\sum_i \sum_k w_{ki} w_{ik} h_{kj}}, \tag{7}$$

$$w_{ik}^t = w_{ik}^{t-1} \frac{\sum_j x_{ij} h_{jk}}{\sum_k \sum_j w_{ik} h_{kj} h_{jk}}. \tag{8}$$

$$w_{ik}^t = \frac{w_{ik}^t}{\sum_i w_{ik}^t}, \quad \text{for all } k. \tag{9}$$

3 NMF Extensions

3.1 Local NMF

As the NMF method permits only additive factors in its decomposition, theoretically, it should lead to basis vectors containing sparse image features. Although this expectation was verified for the facial image database used by Lee and Seung in [5], this is not always the case. It was found that the NMF decomposition rather retrieves more global (spatially distributed) image features for other image databases. To enhance the decomposition sparseness, Li et al [9] have developed the Local Non-negative Matrix Factorization (LNMF) algorithm, imposing more constraints to the KL cost function to get more localized image features. The associated cost function is then given by:

$$f^{KL}_{LNMF}(\mathbf{X}||\mathbf{WH}) \triangleq f^{KL}_{NMF}(\mathbf{X}||\mathbf{WH}) + \alpha \sum_{ij} u_{ij} - \beta \sum_{i} v_{ii}, \tag{10}$$

where $[\mathbf{u}_{ij}] = \mathbf{U} = \mathbf{W}^T\mathbf{W}$ and $[\mathbf{v}_{ij}] = \mathbf{V} = \mathbf{HH}^T$. Here, α and $\beta > 0$ are constants. By maximizing the third term in (10), the total squared projection coefficients over all training images is maximized. The second term can be further split into two parts:

1. $\sum_i u_{ii} \longrightarrow$ min. This term guarantees the generation of more localized features on the basis images $\underline{\mathrm{Z}}$, than those resulting from NMF, since the basis image elements are constrained to be as small as possible.

2. $\sum_{i \neq j} u_{ij} \longrightarrow$ min. This enforces basis orthogonality, in order to minimize the redundancy between image bases.

The following factors updating rules were found for the KL-based LNMF cost function:

$$h^t_{kj} = \sqrt{h^{t-1}_{kj} \sum_i w_{ki} \frac{x_{ij}}{\sum_k w_{ik} h^{t-1}_{kj}}} \tag{11}$$

$$w^t_{ik} = \frac{w^{t-1}_{ik} \sum_j \frac{x_{ij}}{\sum_k w^{t-1}_{ik} h_{kj}} h_{jk}}{\sum_j h_{kj}}. \tag{12}$$

3.2 Discriminant NMF

LNMF was further extended by Buciu and Pitas [10], who developed a NMF variant that takes into account class information. Their algorithm, termed Discriminant Non-negative Matrix Factorization (DNMF), leads to a class-dependent image representation. The KL-based DNMF cost function is given by:

$$f^{KL}_{DNMF}(\mathbf{X}||\mathbf{WH}) \triangleq f^{KL}_{LNMF}(\mathbf{X}||\mathbf{WH}) + \gamma \mathbf{S}_w - \delta \mathbf{S}_b, \tag{13}$$

where γ and δ are constants. The new terms are the within-class $\mathbf{S}_w$ and the between-class $\mathbf{S}_b$ scatter matrix, respectively, expressed as following:

1. $\mathbf{S}_w = \sum_{c=1}^{\mathcal{Q}} \sum_{l=1}^{n_c} (\mathbf{h}_{cl} - \boldsymbol{\mu}_c)(\mathbf{h}_{cl} - \boldsymbol{\mu}_c)^T$. Here, $\mathcal{Q}$ are the image classes and n_c is the number of training samples in class $c = 1, \ldots, \mathcal{Q}$. Each column of the $p \times n$ matrix $\mathbf{H}$ is viewed as image representation coefficients vector $\mathbf{h}_{cl}$, where $c = 1, \ldots, \mathcal{Q}$ and $l = 1, \ldots, n_c$. The total number of coefficient vectors is $n = \sum_{c=1}^{\mathcal{Q}} n_c$. Further, $\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{l=1}^{n_c} \mathbf{h}_{cl}$ is the mean coefficient vector of class c, and $\boldsymbol{\mu} = \frac{1}{n} \sum_{c=1}^{\mathcal{Q}} \sum_{l=1}^{n_c} \mathbf{h}_{cl}$ is the global mean coefficient vector.

2. $\mathbf{S}_b = \sum_{c=1}^{\mathcal{Q}} (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T$ defines the scatter of the class mean around the global mean μ.

By imposing these terms in the cost function, the decomposition coefficients now encode class information and they are updated according to the following expression:

$$h^{(t)}_{kl(c)} = \frac{2\mu_c - 1 + \sqrt{(1 - 2\mu_c)^2 + 8\xi h^{(t-1)}_{kl(c)} \sum_i w^{(t)}_{ki} \frac{x_{ij}}{\sum_k w^{(t)}_{ik} h^{(t-1)}_{kl(c)}}}}{4\xi} \tag{14}$$

where $\xi = \gamma - \beta$ is a constant. The elements h_{kl} are then concatenated for all $\mathcal{Q}$ classes as:

$$h_{kj}^{(t)} = [h_{kl(1)}^{(t)} \,|\, h_{kl(2)}^{(t)} \,|\, \dots \,|\, h_{kl(\mathcal{Q})}^{(t)}] \tag{15}$$

where "|" denotes concatenation. The expression for updating the basis vectors remains the same as in the LNMF approach.

A related work to DNMF has been conducted in parallel by Wang et al. [11], who proposed an algorithm named Fisher non-negative Matrix Factorization (FNMF). However, contrary to the DNMF method, they modified the NMF rather then the LNMF cost function, thus leading to different expressions for decomposition factor updating.

3.3 Non-negative sparse image coding

Spareness is an important issue for image decomposition and representation in the Human Visual System (HVS). Many research studies have been carried out for understanding the way the HVS encodes visual data. Starting with the work of Hubel and T. N.Wiesel [12], many other theoretical papers and experiments brought evidences that the response of the mammalian primary visual cortex (know also as V1 neurons) can be described by localized, oriented and bandpass filters (also known as receptive fields). When applied to natural images, these filters decompose the images into features that are very similar to those obtained by HVS receptive fields. Viewed within this light, Hoyer [13] proposed a new NMF version called Non-negative Sparse Coding (NNSC) where auxiliary constraints are used to impose factor sparseness. The sparseness measure is based on the relation between the L_1 norm and L_2 norm, i.e., sparseness($\mathbf{x}$) $= \frac{\sqrt{m} - \|\mathbf{x}\|_1 / \|\mathbf{x}\|_2}{\sqrt{m} - 1}$. Furthermore, a penalty term of the form $J(\mathbf{W}) = (\zeta \|resh(\mathbf{W})\|_2 - \|resh(\mathbf{W})\|_1)^2$ is introduced in the standard NMF problem, where $\zeta = \sqrt{km} - (\sqrt{km} - 1)\eta$, and $resh(.)$ is the operator which transforms a matrix into a column vector in a column-wise fashion. Here, the desired sparseness for the basis vectors is controlled by η, which can vary from 0 to 1. By replacing $\mathbf{W}$ with $\mathbf{H}$, the sparseness control can be applied to the encoding coefficients.

3.4 Nonsmooth NMF

A sparse NMF variant called nonsmooth NMF (nsNMF) was proposed by Montano et al. in [14], which allows a controlled sparseness degree in both factors. Like LNMF and NNSC methods, nsNMF also leads to a local image decomposition. However, unlike the LNMF approach, nsNMF explicitly modifies the sparseness degree. Also, unlike NNSC, this variant applies the sparseness concept directly to the model, achieving global sparseness. Imposing sparseness in one of the NMF factors (as NNSC does) will almost certainly force smoothness in the other in an attempt to reproduce the data as best as possible. Additionally, forcing sparseness constraints on both the basis and the encoding vectors decreases the data variance explained by the model. The new variant nsNMF seems to be more robust to this effect. The nsNMF decomposition is given by $\mathbf{X} = \mathbf{WOH}$. The matrix $\mathbf{O}_{p \times p}$ is a square positive symmetric "smoothing" matrix defined as:

$$\mathbf{O} = (1 - \upsilon)\mathbf{I} + \frac{\upsilon}{p}\mathbf{1}\mathbf{1}^T, \tag{16}$$

with **I** the identity matrix and **1** is a vector of ones. The parameter $0 \leq \upsilon \leq 0$ controls the extent of smoothness of the matrix operator **O**. However, strong smoothing in **O** will force strong sparseness in both the basis and the encoding vectors, in order to maintain faithfulness of the model to the data. Accordingly, the parameter υ controls the model sparseness. The suitability of the proposed method over NMF, NNSC and LNMF is investigated with respect to the deterioration of the goodness of fit between the data and the model. The nsNMF model maintained almost perfect faithfulness to the data, expressed by a variance (of goodness) value greater than 99 % for a wider range of sparseness degree, compared with the other NMF variants whose variance decreases with sparseness degree modification.

3.5 Projective NMF

A novel method to decompose the input matrix into non-negative factors named projective NMF (ProjNMF) was proposed by Yuan and Oja [15]. The idea is derived from the Singular Value Decomposition (SVD) approach. ProjNMF based on LS minimization can be written as:

$$\begin{aligned} \text{minimize} \quad & f_{ProjNMF}^{LS}(\mathbf{X},\mathbf{W}) = \frac{1}{2}\|\mathbf{X}-\mathbf{W}\mathbf{W}^T\mathbf{X}\|_F^2 \\ \text{subject to} \quad & \mathbf{W} \geq 0, \end{aligned} \tag{17}$$

Note that the encoding matrix **H** is now replaced by the product $\mathbf{W}^T\mathbf{X}$. The LS and KL minimization leads to the following rule for updating **W**:

$$w_{ik} = w_{ik}\frac{\sum_j \sum_i x_{ij}x_{ji}w_{ik}}{\sum_k \sum_j \sum_i w_{ik}w_{ki}x_{ij}x_{ji}w_{ik} + \sum_j \sum_k \sum_i x_{ij}x_{ji}w_{ik}w_{ki}w_{ik}} \tag{18}$$

and

$$w_{ik} = w_{ik}\frac{\sum_j (w_{ki}x_{ij} + \sum_i w_{ki}x_{ij})}{\sum_j x_{kj}\left(\frac{\sum_i w_{ki}x_{ij}}{\sum_k \sum_i w_{ik}w_{ki}x_{ij}} + \sum_i w_{ki}\frac{x_{ij}}{\sum_k \sum_i w_{ik}w_{ki}x_{ij}}\right)} \tag{19}$$

Experimentally, it was found that projNMF produces sparser basis vectors than the standard NMF. However, the reconstruction accuracy (defined as the Euclidean distance between the original input matrix and the factors product) is lower then the one obtained by the standard NMF for the same number of iterations.

3.6 NMF with Temporal Smoothness and Spatial Decorrelation Constraints

The standard NMF was altered in [16] by imposing additional constrains to tackle in particular time varying signals. Here, the rows of **H** represents temporal signal of length n samples. Then the $k-th$ source can be denoted by $h_k(t) \equiv h_{kt}$ and the $k-th$ row vector of **H** by $\mathbf{h}_k$. The temporal smoothness is described by:

$$R = \frac{1}{n}\|\mathbf{h}_k^n - \overline{\mathbf{h}}_k^n\|^2 = \frac{1}{n}\|\mathbf{h}_k^n - \mathbf{T}\mathbf{h}_k^n\|^2, \tag{20}$$

where $\overline{\mathbf{h}}_k^n$ denotes the short-term exponentially weighted average of temporal signal $h_k(t)$ and $\mathbf{h}_k^n$ is the corresponding long-term. As described above, the short-term is expressed as the convolution product between $h_k(t)$ and a template operator $\mathbf{T}$, defined through the following Toeplitz matrix:

$$\mathbf{T} = \begin{bmatrix} \tau & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \rho\tau & \beta & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \rho^2\tau & \rho\tau & \tau & 0 & 0 & 0 & 0 & \dots & 0 \\ \rho^3\tau & \rho^2\tau & \rho\tau & \tau & 0 & 0 & 0 & \dots & 0 \\ \rho^4\tau & \rho^3\tau & \rho^2\beta & \rho\tau & \tau & 0 & 0 & \dots & 0 \\ 0 & \rho^4\tau & \rho^3\tau & \rho^2\tau & \rho v & \tau & 0 & \dots & 0 \\ 0 & 0 & \cdots & & & & & \ddots & \dots \\ 0 & \cdots & & 0 & \rho^4\tau & \rho^3\tau & \rho^2\tau & \rho\tau & \tau \end{bmatrix}$$

, where $\rho \in (0,1)$ is a forgetting factor that determines the local smoothness range, and $\tau = 1 - \rho$.

Finally, the NMF LS and KL based cost function of NMF with temporal smoothness constraint are written as:

$$f_{tNMF}^{LS} = f_{NMF}^{LS} + \varphi \sum_k \mathbf{R}_k, \tag{21}$$

and

$$f_{tNMF}^{KL} = f_{NMF}^{KL} + \frac{\varphi}{2} \sum_k \mathbf{R}_k, \tag{22}$$

respectively. Here, φ is a small regularization coefficient that balances the trade-off between the reconstruction error and temporal smoothness constraint. The corresponding factor updating rules for LS minimization are as follows:

$$h_{kj} = h_{kj} \frac{\sum_i w_{ki} x_{ij}}{\sum_i \sum_k w_{ki} wik h_{kj} + \varphi \sum_j h_{kj} q_{jj}} \tag{23}$$

$$w_{kj} = w_{kj} \frac{\sum_j x_{ij} h_{jk}}{\sum_k \sum_j w_{ik} h_{kj} h_{jk}}, \tag{24}$$

where q_{jj} are the elements of the symmetric square matrix $\mathbf{Q} = \frac{1}{n}(\mathbf{I} - \mathbf{I})^T(\mathbf{I} - \mathbf{I})$. The corresponding factor updating rules for KL minimization are given by:

$$h_{kj} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{25}$$

$$w_{kj} = w_{kj} \frac{\sum_j x_{ij} h_{jk} / \sum_k w_{ik} h_{kj}}{\sum_j h_{kj}}, \tag{26}$$

where $a = \varphi \sum_l q_{jl}$, $b = \sum_i w_{ik}$, and $c = -\sum_i x_{ij} \frac{w_{ik} h_{kj}}{\sum_k w_{ik} h_{kj}}$.

The spatial decorrelation constraint implies that the column vectors s_t are uncorrelated and that the sample correlation matrix $\mathbf{V} = \mathbf{H}\mathbf{H}^T/n$ is diagonally dominant. Notice that this constraint is exactly the same as the one in the LNMF approach, leading to the same cost function, except the constraint associated to the basis vectors.

3.7 NMF Deconvolution

Non-negative matrix factor deconvolution [17] is another particular application derived from the standard NMF to cope with temporal information extracted from sound signals. Here, the decomposition takes the form $\mathbf{V} = \sum_{t=0}^{T-1} \mathbf{W}_t \overrightarrow{\mathbf{H}}$, where t is the time interval and the symbol $\overrightarrow{(\cdot)}$ over the matrix denotes the shift operator that shifts the columns of the matrix by i places to the right. The leftmost columns of the matrix are appropriately set to zero, so as to maintain the original size of the input.

3.8 Shifted NMF

Mørup et al [18] further extended NMF algorithm to allow delays among factor components utilized in source separation. Within this particular application, each row of the input matrix $\mathbf{X}$ represents a sensor, while each column of $\mathbf{W}$ denotes a source to be extracted from the mixture. The decomposition of the so-called shifted NMF (sNMF) is given by $x_{i,j} = \sum_k w_{i,k} h_{k,n-\nu_{m,k}} + e_{i,j}$, where $e_{i,j}$ represents the noise and $\nu_{m,k}$ denotes an arbitrary delay from the $k-th$ source to the $m-th$ sensor. For the LS based cost function, the updating rule for the basis vectors is the one derived from the standard NMF. The encoding matrix $\mathbf{H}$ and the delay term $\nu_{m,k}$ is modified according to:

$$h_{k,j} = h_{k,j} \frac{g^-_{k,j}}{g^+_{k,j}} \tag{27}$$

$$\nu_{m,k} = \nu_{m,k} - \eta b_{m,k} \frac{\partial f^{LS}_{sNMF}}{\partial \nu_{m,k}}. \tag{28}$$

In the above expressions, $g^-_{k,j}$ and $g^+_{k,j}$ are the inverse DFT (time domain) of the quantities $\widetilde{g}^-_{k,j} = \frac{1}{p}\sum_i \widetilde{w}^{(f)}_{ki}\widetilde{x}^{(f)}_{ij}$ and $\widetilde{g}^+_{k,j} = \frac{1}{p}\sum_i \sum_k \widetilde{w}^{(f)}_{ki}\widetilde{w}^{(f)}_{ik}\widetilde{h}^{(f)}_{kj}$, respectively, expressed in the frequency domain. These quantities are computed by taking the derivative of the sNMF LS based function with respect to $\widetilde{\mathbf{H}}$ in frequency domain. The quantities $b_{m,k}$ represent the elements of the Hessian matrix, i.e., the second order partial derivatives of the cost function with respect to $\nu_{m,k}$. η is a step parameter.

3.9 Incremental NMF

An incremental version of the NMF algorithm was proposed in [19]. This extension is very useful when new data arrives, i.e., when adding new columns to $\mathbf{X}$. For standard NMF, this batch-mode operation implies re-running the algorithm from scratch, thus increasing the processing time. Incremental NMF (INMF) allows concatenation of additional columns to the initial input matrix without increasing the computational load. This approach was applied to model the dynamic content of a surveillance video. Here, p denotes the initial number of frames. With a new video frame sample $n+1$, the INMF cost function is expressed as follows:

$$f_{INMF(n+1)} = (1-\theta) f_{INMF(n)} + \theta \sum_{i=1}^{m} \left(x_i - \sum_k w_{ik} h_k \right) \tag{29}$$

The INMF minimization leads to the following updating rules:

$$h_k^t = h_k^{t-1} \frac{\sum_i w_{ki(n)} \frac{x_{ij}}{\sum_k w_{ik(n)} h_k^{t-1}}}{\sum_i w_{ik(n)}}, \tag{30}$$

$$w_{ik}^t = w_{ik}^{t-1} \frac{\sum_j (1-\theta) x_{ij(n)} h_{jk(n)} + \sum_j \theta x_{ij} h_{jk}^t}{\sum_k \sum_j (1-\theta) w_{ik(n)} h_{kj(n)} h_{jk(n)} + \sum_j \theta h_{kj} h_{jk}}. \tag{31}$$

The parameter $\theta \in (0,1)$ controls its ability to adapt to dynamic content changes. Note that each video frame in $\mathbf{X}_{n+1}$ is reconstructed with the help of the corresponding column of the encoding matrix $\mathbf{H}_{n+1}$. Consequently, the derivative for computing the updating rules for the factors is taken with respect to the columns of $\mathbf{H}_{n+1}$. Also, whenever a new video frame $(n+1)$ arrives, INMF does not need to update all elements of the matrix $\mathbf{H}_{n+1}$, but only the elements corresponding to the new updated video frame. Therefore, the number of factor updating per iteration is fixed and no additional computational load is required.

3.10 Sparse higher order NMF

An interesting NMF extension which allows higher order matrix (tensor) decomposition into non-negative factors is developed by Mørup et al [20]. Tensors (also known as multidimensional matrices) are generalization of vectors (first order tensors) or matrices (second order tensors), i.e., $\mathbf{X} \in \mathcal{C}^{I_1 \times I_2 \times \ldots I_N}$. The tensors can be decomposed according to:

$$\mathbf{X}_{i_1,i_2,\ldots i_N} = \sum_{j_1,j_2,\ldots j_N} \mathcal{G}_{j_1,j_2,\ldots j_N} \mathbf{A}^{(1)}_{i_1,j_1} \mathbf{A}^{(1)}_{i_1,j_1} \cdot \ldots \cdot \mathbf{A}^{(N)}_{i_N,j_N}, \tag{32}$$

where the core tensor $\mathcal{G} \in \mathcal{C}^{J_1 \times J_2 \times \ldots J_N}$ and $\mathbf{A}^n \in \mathcal{C}^{I_N \times J_N}$. Furthermore, the decomposition can be written as:

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{Z}_{(n)} = \mathbf{A}^{(n)} \mathbf{G}_{(n)} (\mathbf{A}^{(N)} \otimes \ldots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \ldots \otimes \mathbf{A}^{(1)})^T. \tag{33}$$

The $n-th$ modality is spanned by the vectors comprised by the columns of $\mathbf{A}^{(n)}$, while the vectors of each modality interact with the strength expressed by the core tensor to reconstruct the data. By exchanging $\mathbf{W}$ with $\mathbf{A}^{(n)}$ and $\mathbf{H}$ with $\mathbf{Z}_{(n)}$, the HONMF recasts into the standard NMF problem. The LS and KL cost functions are expressed as:

$$f^{LS}_{HONMF}(\mathbf{X}, \mathbf{WH}) \triangleq f^{LS}_{NMF}(\mathbf{X}, \mathbf{WH}) + \lambda\, C_{sparse}(\mathbf{H}), \tag{34}$$

$$f^{KL}_{HONMF}(\mathbf{X}||\mathbf{WH}) \triangleq f^{KL}_{NMF}(\mathbf{X}||\mathbf{WH}) + \lambda\, C_{sparse}(\mathbf{H}). \tag{35}$$

Here, $C_{sparse}(\mathbf{H})$ is a function used to controls the sparseness degree of $\mathbf{H}$. The HONMF updates the factors $\mathbf{A}$ and $\mathcal{G}$, until the convergence is achieved. For the LS minimization problem, the updating rules are described as follows:

$$\begin{gathered}
\mathbf{A}^{(n)} = \mathbf{A}^{(n)} \otimes \left(\mathbf{X}_{(n)} \mathbf{Z}^T_{(n)}\right) \oslash \left(\mathbf{A}^{(n)} \mathbf{Z}_{(n)} \mathbf{Z}^T_{(n)}\right) \\
\mathcal{R} = \mathcal{G} \times_1 \mathbf{A}^{(1)^T} \times_1 \mathbf{A}^{(1)^T} \times_3 \ldots \times_N \mathbf{A}^{(N)^T} \\
\mathcal{B} = \mathcal{X} \times_1 \mathbf{A}^{(1)^T} \times_1 \mathbf{A}^{(1)^T} \times_3 \ldots \times_N \mathbf{A}^{(N)^T} \\
\mathcal{C} = \mathcal{R} \times_1 \mathbf{A}^{(1)^T} \times_1 \mathbf{A}^{(1)^T} \times_3 \ldots \times_N \mathbf{A}^{(N)^T} \\
\mathcal{G} = \mathcal{G} \otimes \mathcal{B} \oslash \mathcal{C}.
\end{gathered} \tag{36}$$

Here, $\otimes$ and $\oslash$ denote elementwise multiplication and division, respectively. The updating rules for the KL-based cost function are:

$$\begin{aligned} \mathbf{A}^{(n)} &= \mathbf{A}^{(n)} \otimes \left(\mathbf{X}_{(n)} \oslash (\mathbf{A}^{(n)}\mathbf{Z}_{(n)}) \otimes \mathbf{Z}^T_{(n)}\right) \oslash \left(\mathbf{I}^{(n)}\mathbf{Z}^T_{(n)}\right) \\ \mathcal{R} &= \mathcal{G} \times_1 \mathbf{A}^{(1)^T} \times_1 \mathbf{A}^{(1)^T} \times_3 \ldots \times_N \mathbf{A}^{(N)^T} \\ \mathcal{D} &= \mathcal{X} \oslash \mathcal{R} \times_1 \mathbf{A}^{(1)^T} \times_1 \mathbf{A}^{(1)^T} \times_3 \ldots \times_N \mathbf{A}^{(N)^T} \\ \mathcal{F} &= \mathcal{E} \times_1 \mathbf{A}^{(1)^T} \times_1 \mathbf{A}^{(1)^T} \times_3 \ldots \times_N \mathbf{A}^{(N)^T} \\ \mathcal{G} &= \mathcal{G} \otimes \mathcal{D} \oslash \mathcal{F}, \end{aligned} \tag{37}$$

where $\mathbf{I}$ and $\mathcal{E}$ is the identity matrix and the tensor of ones in all entries, respectively. Synthetic data consisting of 5 images of logical operators mixed through two modalities was used for experiments. HONMF was found to identify almost all components, while the standard NMF fails to estimate the proper components.

3.11 Polynomial NMF

Recently, a kernelized NMF extension termed polynomial NMF (PNMF) was proposed in [21] to handle nonlinear non-negative factor decomposition. The input matrix $\mathbf{X}$ is firstly transformed through a nonlinear polynomial kernel mapping into a higher dimensional space $\mathcal{F} \subseteq \mathbb{R}_{l \times n}$, $l \gg m$ (called reproducing kernel Hilbert or feature space), followed by a nonnegative decomposition within this feature space. The proposed nonlinear mapping enables higher-order correlation between input variables, thus discovering features which posses non-linear dependencies. If the transformed input data is denoted by $\mathbf{F} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n)]$, with l - dimensional vector
$\phi(\mathbf{x}_j) = [\phi(\mathbf{x})_1, \phi(\mathbf{x})_2, \ldots, \phi(\mathbf{x})_s, \ldots, \phi(\mathbf{x})_l]^T \in \mathcal{F}$, one can find a matrix
$\mathbf{Y} = [\phi(\mathbf{z}_1), \phi(\mathbf{z}_2), \ldots, \phi(\mathbf{z}_p)]$, $\mathbf{Y} \in \mathcal{F}$, that approximates the transformed data set, so that $p < n$. Consequently, each vector $\phi(\mathbf{x})$ can be written as a linear combination $\phi(\mathbf{x}) \approx \mathbf{Yb}$. The PNMF LS based cost function is expressed as:

$$f^{LS}_{PNMF} = \|\phi(\mathbf{X}) - \mathbf{YB}\|^2. \tag{38}$$

The minimization is subject to $b_r, Z_{ir} \geq 0$, and $\sum_{i=1}^{m} Z_{ir} = 1$. The updating rules for the non-negative factors $\mathbf{B}$ and $\mathbf{Z}$ are given above:

$$\mathbf{B} = \mathbf{B} \otimes \mathbf{K}_{zx} \oslash (\mathbf{K}_{zz}\mathbf{B}) \tag{39}$$

$$\mathbf{Z} = \mathbf{Z} \otimes [(\mathbf{X}\mathbf{K}^{'}_{xz}) \oslash (\mathbf{Z}\boldsymbol{\Omega}\mathbf{K}^{'}_{zz})] \tag{40}$$

$$\mathbf{Z} = \mathbf{Z} \oslash \mathbf{S} \tag{41}$$

where $\mathbf{K}_{zx} := \langle\phi(\mathbf{z}_i), \phi(\mathbf{x}_i)\rangle$ and $\mathbf{K}_{xz} := \langle\phi(\mathbf{x}_i), \phi(\mathbf{z}_i)\rangle$ are kernel matrices of dimensions $p \times n$ and $n \times p$, respectively, containing values of kernel functions of $\mathbf{z}_i \in \mathbf{Z}$ and $\mathbf{x}_i \in \mathbf{X}$, and $\mathbf{K}_{zz} = \langle\phi(\mathbf{z}_i), \phi(\mathbf{z}_o)\rangle$ is a $p \times p$ kernel matrix of any vectors $\mathbf{z}_i, \mathbf{z}_o \in \mathbf{Z}$. $\boldsymbol{\Omega}$ is a diagonal matrix whose diagonal elements are $\omega_{rr} = \sum_{j=1}^{n} B_{rj}$, $r = 1, \ldots, p$. The columns of $\mathbf{S}$ are given by $\mathbf{s}_r = \sum_{i=1}^{m} Z_{ir}$, $r = 1, \ldots, p$. The sign " $'$ " denotes the derivative of matrix elements. For the polynomial kernel $k^{'}(\mathbf{x}_i, \mathbf{x}_j) = d \cdot k(\mathbf{x}_i \cdot \mathbf{x}_j)^{d-1}$.

4 Applications

The standard NMF approach and its variants have been extensively used as feature extraction techniques for various applications, especially for high dimensional data analysis. The newly formed low dimensionality subspace represented by the basis vectors should capture the essential structure of the input data as best as possible. Although, theoretically, NMF could be applied to data compression, not much work was carried out in this regard. Rather, the computer vision community focused its attention to the application of NMF to pattern recognition applications, where the extracted NMF features are subsequently clustered or classified using classifiers. The NMF applications can be characterized according to several criteria. We provide the following application classes:

- *1D signal* applications (including sounds and EEG data), where the input matrix $\mathbf{X}$ contains in its columns one-dimensional data varying over time.

- *2D signal* applications (face object images, etc.), where the input matrix $\mathbf{X}$ contains in its columns a vectorized version of the 2D signals (basically 2D images) obtained by lexicographically concatenating the rows of the two-dimensional images.

- *Other applications*, including text or e-mail classification.

4.1 1D signal applications

The separation of pitched musical instruments and drums from polyphonic music is one application, where NMF was considered by Helén and Virtanen in [22]. The NMF splits the input data spectrogram into components which are further classified by an SVM to be associated to either pitched instruments or drums. Within this application, each column of the input matrix $\mathbf{X}$ represents a short-time spectrum vector $\mathbf{x}_t$. The non-negative decomposition takes the form $\mathbf{x}_t = \sum_{i=1}^{n} \mathbf{s}_n a_{i,t}$, where $\mathbf{s}_n$ is the spectrum of $n-th$ component, $a_{i,t}$ is the gain of $n-th$ component in frame t, and n is the component number. Individual musical instrument sounds extraction using NMF was exploited by Benetos et al. in [23]. A number of 300 audio files are used, corresponding to 6 different instrument classes (piano, violin, cello, flute, bassoon, and soprano saxophone) [24]. Two sorts of features are used to form the input matrix. The first feature set is composed of 9 audio specific features and MPEG-7 specifications (such as zero-crossing rate, delta spectrum, mel-frequency cepstral coefficients, etc). The second feature set is given by the rhythm pattern described by several other audio characteristics (such as power spectrum, critical bands, modulation amplitude, etc).

One particular NMF application is on spectral data analysis. Source spectra separation from magnetic resonance (MR) chemical shift imaging (CSI) of human brain using constrained NMF analysis was investigated by Sajda et al. [25]. In CSI, each tissue is characterized by a spectral profile or a set of profiles corresponding to the chemical composition of the tissue. In tumors, for instance, metabolites are heterogeneously distributed and, in a given voxel, multiple metabolites and tissue types may be present, so that the observed spectra are a combination of different constituent spectra. Consequently, the spectral amplitudes of the different coherent resonators are additive, making the application of NMF reasonable. The overall gain with which a tissue type contributes to this addition is proportional to its concentration in each voxel, such that **X** is the observed spectra, the columns of **W** represents concentration of the constituent materials, and the rows of **H** comprises their corresponding spectra.

The spectral data analysis was also investigated in [26] by proposing a constraint NMF algorithm to extract spectral reflectance data from a mixture for space object identification. In this application, each observation of an object is stored as a column of a spectral trace matrix **X**, while its rows correspond to different wavelengths. Each column of **W**, called endmember, is a vector containing nonnegative spectral measurements along p different spectral bands, where each row of **H** comprises the fractional concentration.

Multichannel EEG signals have been analyzed via NMF concept by Rutkowski et al. in [27]. The signals are firstly decomposed into intrinsic modes in order to represent them as a superposition of components with well defined instantaneous frequencies called IMF. The resulting trace IMF components form the input matrix **X**, while **W** is the mixing matrix containing the true sub-spectra.

Finally, NMF has been tailored to address the plant-wide oscillation detection problem by Tangirala et al. [28]. This industrial application deals with the presence of oscillations in control loop measurements which can drastically affect the control loop performance and plant productivity. The input matrix **X** represents the total power which characterizes the overall spectral plant behavior. The total power is the normalized power spectrum sums at certain frequencies over the number of measurements n. Thus, NMF can be viewed as the decomposition of the total power by each basis shape.

4.2 Image applications

One of the first NMF applications in images is on face recognition tasks. Li et al [9] explored this issue for both NMF and LNMF techniques, when a simple Euclidean distance is used as classifier. Their experiments revealed the superiority of LNMF over the standard NMF for the ORL face database [29], especially for occluded faces. Guillamet and Vitrià [38] also applied NMF to a face recognition task. A third framework dealing with the face recognition task is described in [31], where the DNMF is employed along NMF and LNMF for comparison. Also, besides the Euclidean distance, two other classifiers (cosine similarity measure and SVMs) are utilized. Two databases, namely ORL and YALE [32] are utilized here. The experiments showed that the NMF seems to be more robust to illumination changes than LNMF and DNMF, since the variation of illumination conditions for the faces pertaining to Yale database is much more intense than for images from the ORL database. Contrary to the results obtained for ORL, where LNMF gave the highest recognition rate, when face recognition is performed on the YALE database, the best results are obtained by the NMF algorithm. Although the ORL database, generally, contains frontal faces or slightly rotated facial poses. This can contribute to the superior performance of LNMF, since this algorithm is rotation invariant (up to some degree), because it generates local features in contrast to NMF which yields more distributed features.

Buciu and Pitas applied NMF and LNMF for facial expression recognition in [33] and compared them with the DNMF algorithm in [10] for the same task. It was found that, for the facial expression recognition task, the DNMF method outperforms the other two techniques for the Cohn-Kanade AU-coded facial expression database [34].

The NMF for object recognition is investigated by Spratling [35], where an empirical investigation of the NMF performance with respect to its sparseness issue for occluded images is reported. The experiments were conducted for the standard bars problem, where the training data consists of 8×8 pixel images in which each of the 16 possible (one-pixel wide) horizontal and vertical bars can be present with a probability of $1/8$. The occlusion was simulated by overlapping between horizontal and vertical bars. Several NMF variants (i.e., NMF, LNMF, and NNSC) have been tested. It was found that no NMF method was able to identify all the components in the unequal bars (occlusion) problem for any value of the sparseness parameter. To overcome this situation, the author proposed a non-negative dendritic inhibition neural network, where the neural activations identified in the rows of $\mathbf{H}$ reconstruct the input patterns $\mathbf{X}$ via a set of feedforward (recognition) weights $\mathbf{W}$. When applied to face images, the proposed NMF neural network learns representations of elementary image features more accurately than other NMF variants. Within the same object recognition framework, empirical results for NMF, LNM, and DNMF techniques were reported in [36]. The techniques were applied to the Columbia Object Image Library (COIL-100) database [37] comprising color images of 100 objects. The object images were taken at pose intervals of 5 degrees, resulting in 72 poses per object, producing a total of 7200 images. Once the NMF (LNMF, DNMF) features were extracted, the feature vector used for classification is built up either by projecting the input data onto the pseudo-inverse of $\mathbf{W}$, or by projecting the input data onto the transpose of $\mathbf{W}$. The analysis was performed to investigate its classification accuracy with respect to the basis images sparseness degree. As far as the sparseness issue is concerned, the experiments showed a direct relatively strong correlation between sparseness and the recognition performance in the case of LNMF, a moderate correlation in the case of DNMF, and an opposite correlation for NMF, when the inverse of the basis images matrix was employed. A much stronger correlation between the sparseness degree and the recognition rate was noticed for all three algorithms, when the feature vectors is formed using the transpose of the basis image matrix. Guillamet et al. experimentally compared NMF to the Principal Component Analysis (PCA) for image patch classification in [38]. In these experiments, 932 color images from the Corel Image database were used. Each of these images belongs to one of 10 different classes of image patches (clouds, grass, ice, leaves, rocky mountains, sand, sky, snow mountains, trees and water). NMF outperformed PCA. Finally, a face detection approach based on LNMF was proposed by Chen et al. in [39].

4.3 Other applications

The application of NMF for text classification was undertaken in [40]. This application is characterized by a large number of classes and a small training data size. In their formulation, the elements $w_{ik} \geq 0$ represent the confidence score of assigning the k-th class label to the i-th example. Furthermore, $\mathbf{H} = \mathbf{B}\mathbf{W}^T$, where the non-negative matrix $\mathbf{B}$ captures the correlation (similarity) among different classes. However, the linear constraint that restricts the matrix $\mathbf{H}$ to be linearly dependent on the matrix $\mathbf{W}$ delineates the algorithm from the standard NMF approach. For experiments, 3456 documents were picked up from the textual data of the Eurovision St Andrews Photographic Collection (ESTA) in ImageCLEF collection [41]. On average, each document is assigned to 4.5 classes. As baseline, for comparison, three methods namely Spectral Graph Transducer, Multi-label

Informed Latent Semantic Indexing, and Support Vector Machines, were employed. The constrained NMF-based algorithm showed superiority in correctly assigning the document to the proper class, over the aforementioned three methods.

Document clustering using NMF was also investigated by Shahnaz et al. in [42]. Two databases are used for experiments. The first one, Reuters data corpus [43], contains 21578 documents and 135 topics or document clusters created manually. Each document in the corpus has been assigned one or more topics or category labels based on its content. Seven values are picked up for p and three different document collections are generated by filtering the documents according to different topic files. Thus, NMF was performed on all 21 matrices. For any given $\mathbf{X}$, with p topics and n documents, the matrix $\mathbf{W}$ has p columns or basis vectors that represent p clusters, while matrix $\mathbf{H}$ has n columns representing the number of documents. A column vector in $\mathbf{H}$ has p components, each of which denoting the contribution of the corresponding basis vector to that column or document. The classification or clustering of documents is performed based on the index of the highest value of p for each document. The same procedure was employed for the second database, named TDT2 [44].

The extraction and detection of concepts or topics from electronic mail messages is a NMF application proposed by Berry and Browne in [45]. The input matrix $\mathbf{X}$ contains n messages indexed by m terms (or keywords). Each matrix element $x_{i,j}$ defines a weighted frequency at which the term i occurs in message j. Furthermore, $x_{i,j}$ is decomposed as $x_{i,j} = l_{i,j} g_i d_j$, where $l_{i,j}$ is the local weight for the term i occurring in message j, g_i is the global weight for i - th term in the subset, and d_j is a document normalization factor, which specifies whether or not the columns of $\mathbf{X}$ (i.e., the documents) are normalized. Next, a normalized term $p_{i,j} = f_{i,j} / \sum_j f_{i,j}$ is defined, where $f_{i,j}$ denotes frequency that term i appears in the message j. Then, two possible definitions exist for $x_{i,j}$. The first one sets $l_{i,j} = f_{i,j}$, $g_{i,j} = 1$, while the second interpretation sets $l_{i,j} = \log(1 + f_{i,j})$ and $g_{i,j} = 1 + (\sum_j p_{i,j} \log(p_{i,j}) / \log n)$, respectively. After NMF decomposition, the semantic feature represented by a given basis vector $\mathbf{w}_k$ (k - th column of the matrix) by simply sorting (in descending order) its i elements and generating a list of the corresponding dominant terms (or keywords) for that feature. In turn, a given row of $\mathbf{H}$ having n elements can be used to reveal messages sharing common basis vectors $\mathbf{w}_k$, i.e., similar semantic features or meaning. The columns of $\mathbf{H}$ are the projections of the columns (messages) of $\mathbf{X}$ onto the basis spanned by the columns of $\mathbf{W}$.

A chemometric application of the NMF method is proposed by Li et al. [46] where several NMF variants are used to detect chemical compounds from a chemical substances represented through Raman spectroscopy. The input matrix contains the observed total mixture chemical spectra, the basis vectors denote the contribution of chemical spectra, while the spectra is encoded into $\mathbf{H}$.

5 Conclusions

Typically, NMF decomposes an input matrix $\mathbf{X}$ into two nonnegative factors $\mathbf{W}$ and $\mathbf{H}$, where $\mathbf{X} \approx \mathbf{WH}$. Solving the NMF problem implies finding these decomposition factors, so that their product approximate the original input matrix as best as possible. This is equivalent to building an associated cost function to be minimized. The NMF algorithms can be categorized into three classes: multiplicative update algorithms, gradient descent algorithms and alternating least squares algorithms. We did not intend to describe the NMF approach from this view point. Interested readers may consult [1], where the numerical stability of NMF classes is investigated. Rather, this paper aims at describing novel recent NMF variants, where, apart from the mandatory nonnegativity constraints on both factors, other additional application-based constraints are imposed. Some relevant NMF applications were also considered.

As far as the NMF open problems are concerned, several challenges exist, as follows:

- The optimization problem. All the described NMF variants suffer from the same drawback: no global minimum is guaranteed; they only lead to a local minimum, thus several algorithm runs may be necessary to avoid getting stuck in an undesired local minimum. However, in practice, even local minima can provide satisfactory results. Having an approach which conducts to a global minimum will greatly improve the numerical NMF stability.

- Initialization of $\mathbf{H}$ and $\mathbf{W}$. Basically, the factors are initialized with random nonnegative values. A few efforts were undertaken in order to speed up the convergence of the standard NMF. Wild [47] proposed a spherical k-means clustering to initialize $\mathbf{W}$. More recently, Boutsidis and E. Gallopoulos [48] employed an SVD-based initialization. For the DNMF algorithm, Buciu et al. [49] constructed initial basis vectors, whose values are not randomly chosen but contain information taken from the original database. The redundant information of initial basis images is then minimized by imposing orthogonality "a priori". Additionally, the original basis vectors are projected into a sparse and non-negative feature space. Finally, by using a least squares approach, a first approximation for the $\mathbf{X} \approx \mathbf{WH}$ problem leads closer to the final solution, speeding up the DNMF convergence. However, this issue is an open problem and needs further improvements for the standard NMF approach and it variants.

- Speed up of the algorithmic convergence. Apart from starting with non-random values for the factors, several other attempts were carried out to lower the number of iterations needed for the standard NMF to get to the solution. This includes the use of a projected gradient bound constrained optimization technique which was found to be more efficient in convergence terms [50]. Gonzales and Zhang [51] proposed an interior-based approach to accelerate the standard NMF, while Zdunek and Cichocki [52] employed a second-order optimization constraint in the NMF cost function. However, the proposed optimization technique provides slightly worse results compared to the standard NMF, when applied to unmix a set of image mixture. The issue remains open.

- Subspace selection. To date, there is no approach suggesting, a priori, the optimal choice of p for the best performances. The issue is difficult and data-dependent. Typically, the algorithms run for several values of p and the subspace dimension corresponding to the highest recognition rate is reported. Also, before data projection, the resulting basis vectors may be re-ordered according to some criteria (descending order of sparseness degree, discriminative capabilities, etc).

- Nonlinear nonnegative features. Standard NMF linearly decomposes the data. The kernel-based NMF approach proposed in [21] tends to retrieve nonlinear negative features. However, the way the factors are decomposed limits the kernel type. More precisely, the method only allows the polynomial kernel function. Finding an approach to incorporate other kernel functions for more flexibility is a future challenge.

References

[1] A. N. Langville, M. W. Berry, M. Browne, V. P. Pauca, and R. J. Plemmons,"Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vil. 52, no. 1, pp. 155–173, 2007.

[2] W. X. Liu, N. N. Zheng, and Q. B. You," Nonnegative matrix factorization and its applications in pattern recognition," *Chinese Science Bulletin*, vol. 51, pp. 7–18, 2006.

[3] I. Buciu, "Non-negative matrix factorization, a new tool for feature extraction: Theory and Applications," *IEEE 2nd International Conference on Computers, Communications and Control*, pp. 45–52, 2008.

[4] P. Paatero and U. Tapper, "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, pp. 111–126, 1994.

[5] D. D. Lee and H. S. Seung, "Learning the parts of the objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[6] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[7] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, no. 22, pp. 79–86, 1951.

[8] L. M. Bregman, "The relaxation method of findind the common point of convex sets and its application to the solution of problems in convex programming," *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 1, no. 7, pp. 200–217, 1967.

[9] S. Z. Li, X. W. Hou and H. J. Zhang, "Learning spatially localized, parts-based representation," *Int. Conf. Computer Vision and Pattern Recognition*, pp. 207–212, 2001.

[10] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," in Proc. *IEEE Workshop on Machine Learning for Signal Processing*, pp. 539–548, 2004.

[11] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher non-negative matrix factorization for learning local features," in *Asian Conference on Computer Vision*, Korea, January 27-30, 2004.

[12] D. H. Hubel and T. N.Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *Journal of Physiology*, vol. 195, pp. 215-243, 1968.

[13] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[14] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, nr. 3, pp. 403–415, 2006.

[15] Z. Yuan and E. Oja, "Projective nonnegative matrix factorization for image compression and feature extraction," *14th Scandinavian Conference on Image Analysis*, pp. 333-342, 2005.

[16] Z. Chen and A. Cichocki, "Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints," Preprint, 2005.

[17] Paris Smaragdis, "Non-negative matrix factor deconvolution; Extraction of multiple sound sourses from monophonic inputs," *Int. Cong. on Independent Component Analysis and Blind Signal Separation*, vol. 3195, pp. 494, 2004.

[18] M. Mørup, K. H. Madsen, and L. K. Hansen, "Shifted Non-negative Matrix Factorization," *IEEE Workshop on Machine Learning for Signal Processing*, 2007.

[19] S. S. Bucak, B. Gunsel, and O. Gursoy, "Incremental non-negative matrix factorization for dynamic background modelling," *ICEIS 8th International Workshop on Pattern Recognition in Information Systems*, 2007.

[20] M. Mørup, L. K. Hansen, and S. M. Arnfred, "Algorithms for sparse higher order non-negative matrix factorization (HONMF)," *Technical Report*, 2006.

[21] I. Buciu, N. Nikolaidis, and I. Pitas, "Non-negative matrix factorization in polynomial feature space," *IEEE Trans. on Neural Nerworks*, in Press, 2008.

[22] M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," *13th European Signal Processing Conference*, 2005.

[23] E. Benetos, C. Kotropoulos, T. Lidy, A. Rauber, "Testing supervised classifiers based on non-negative matrix factorization to musical instrument classification," in *Proc. of the 14th European Signal Processing Conference*, 2006.

[24] Univ. of Iowa Musical Instrument Sample Database, http://theremin.music.uiowa.edu/index.html.

[25] P. Sajda, S. Du, T. Brown, R. Stoyanova, D. Shungu, X. Mao, and L. Parra, "Non-negative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain," *IEEE Trans. on Medical Imaging*, vol. 23, no. 12, pp. 1453–1465, 2004.

[26] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative Matrix Factorization for Spectral Data Analysis," *Linear Algebra and its Applications*, vol. 416, pp. 29–47, 2006.

[27] T. M. Rutkowski, R. Zdunek, and A. Cichocki, "Multichannel EEG brain activity pattern analysis in timefrequency domain with nonnegative matrix factorization support," *International Congress Series*, vol. 1301, pp. 266–269, 2007.

[28] A. K. Tangirala, K. Kanodia, S. L. Shah, "Non-negative matrix factorization for detection and diagnosis of plant-wide oscillations," *Industrial Engineering Chemistry Research*, vol. 46, no. 3, pp. 801–817, 2007.

[29] http://www.uk.research.att.com/

[30] D. Guillamet and Jordi Vitrià, "Non-negative matrix factorization for face recognition," *Topics in Artificial Intelligence*, Springer Verlag Series: Lecture Notes in Artificial Intelligence, vol. 2504, pp. 336–344, 2002.

[31] I. Buciu, N. Nikolaidis, and I. Pitas, "A comparative study of NMF, DNMF, and LNMF algorithms applied for face recognition," *2006 Second IEEE-EURASIP International Symposium on Control, Communications, and Signal Processing*, 2006.

[32] http://cvc.yale.edu

[33] I. Buciu and I. Pitas, "Application of non-negative and local non negative matrix factorization to facial expression recognition," *International Conference on Pattern Recognition*, pp. 288–291, 2004.

[34] T. Kanade, J. Cohn and Y. Tian, "Comprehensive database for facial expression analysis," *in Proc. IEEE Inter. Conf. on Face and Gesture Recognition*, pp. 46–53, 2000.

[35] M. W. Spratling, "Learning image components for object recognition," *Journal of Machine Learning Research*, vol. 7, pp. 793–815, 2006.

[36] M. Buciu, "Learning sparse non-negative features for object recognition," *IEEE Third International Conference on Intelligent Computer Communications and Processing*, pp. 73–79, 2007.

[37] http://www.cs.columbia.edu/CAVE.

[38] D. Guillamet, B. Schiele, and J. Vitrià, "Analyzing non-negative matrix factorization for image classification," *International Conference on Pattern Recognition*, vol. 2, pp. 116–119, 2002.

[39] X. Chen, L. Gu, S. Z. Li, and H.-J. Zhang, "Learning representative local features for face detection," *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1126–1131, 2001.

[40] Y. Liu, R. Jin, and L. Yang, "Semi-supervised Multi-label Learning by Constrained Non-negative Matrix Factorization," in *Proc. of The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, vol. 21, pp. 421–426, 2006.

[41] ImageCLEF - 2003, "The CLEF Cross Language Image Retrieval Track (ImageCLEF)," available at *http://ir.shef.ac.uk/imageclef/.*

[42] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Journal on Information Processing & Management*, vol. 42, no. 2, pp. 373-386, 2004.

[43] *http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.*

[44] *http://www.ldc.upenn.edu/.*

[45] M. W. Berry and M. Browne, "Email Surveillance Using Nonnegative Matrix Factorization," *Computational & Mathematical Organization Theory*, vol. 11, pp. 249-264, 2005.

[46] H. Li, T. Adali, W. Wang, D. Emge, A. Cichocki, "Non-negative matrix factorization with orthogonality constraints and its application to Raman spectroscopy," *Journal of VLSI Signal Processing*, vol. 48, no. 1-2, pp. 83-97, 2007.

[47] S. Wild, "Seeding non-negative matrix factorization with the spherical k-means clustering," Master thesis, University of Colorado, 2002.

[48] C. Boutsidis and E. Gallopoulos, "On SVD-based initialization for nonnegative matrix factorization," Tech. Rep. HPCLAB-SCG-6/08-05, University of Patras, Patras, Greece., 2005.

[49] I. Buciu, N. Nikolaidis, and I. Pitas, "On the initialization of the DNMF algorithm," *IEEE International Symposium on Circuits and Systems*, pp. 4671-4674, 2006.

[50] C.-J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756-2779, 2007.

[51] E. Gonzalez and Y. Zhang, "Accelerating the Lee-Seung algorithm for nonnegative matrix factorization," Technical Report TR-05-02, Rice University, 2005.

[52] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, pp. 1904-1916, 2007.

Ioan Buciu
University of Oradea
Faculty of Electrical Engineering and Information Technology
Department of Electronics
Address: Universitatii 1, 410087, Oradea, Romania
http://webhost.uoradea.ro/ibuciu/
E-mail: ibuciu@uoradea.ro

Ioan Nafornita
"Politehnica" University of Timisoara
Electronics and Communications Faculty
Bd. Vasile Parvan, no.2
300223 Timisoara, Romania
E-mail: ioan.nafornita@etc.upt.ro

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 51-62

A Multi-Agent Stigmergic Model for Complex Optimization Problems

Camelia Chira, Camelia M. Pintea and D. Dumitrescu

Abstract: Combinatorial optimization problems arise in many and diverse real-world applications. Systems composed of several interacting autonomous agents are investigated for their potential to efficiently address such complex real-world problems. Agents typically communicate by directly exchanging information and knowledge about the environment and have the ability to learn while acting and reacting in their environment. It is proposed to further endow agents with stigmergic behaviour in order to cope with complex combinatorial problems. This means that agents are able to indirectly communicate by producing and being influenced by pheromone trails. Furthermore, stigmergic agents within a system have different sensitivity levels facilitating a balance between direct and stigmergic communication. For better search diversification and intensification, agents can learn to modify their sensitivity level according to environment characteristics and previous experience. The resulting computational metaheuristic combines sensitive stigmergic behaviour with direct agent communication and learning for combinatorial optimization. The proposed model has been tested for solving various instances of NP-hard problems and numerical experiments indicate the robustness and potential of the new metaheuristic.

Keywords: agent communication, stigmergy, sensitivity, multi-agent system, ant colony optimization

1 Introduction

Metaheuristics can efficiently find high-quality near optimal solutions within reasonable running time for problems of realistic size and complexity [5]. This paper investigates the potential of models based on interacting agents to address combinatorial optimization problems.

A metaheuristic combining stigmergic behaviour and agent direct communication is proposed. The proposed model involves several two-way interacting agents. On one hand, agents are endowed with a stigmergic behaviour [3, 11] similar to that of *Ant Colony Systems* [8, 9]. This means that each agent is able to produce pheromone trails that can influence future decisions of other agents. On the other hand, agents can communicate by directly exchanging messages - a behaviour similar to that of multi-agent systems [13, 15, 19]. The information directly obtained from other agents is very important in the search process and can become critical in a dynamic environment (where the latest changes in the environment can be transmitted to other agents).

Stigmergic agents are characterized by a certain level of sensitivity to the pheromone trail facilitating a balance between direct and stigmergic communication. The agents have different sensitivity levels allowing various types of reactions to a changing environment. Furthermore, the senitivity level of each agent is not static as the agent has the ability to dynamically change sensitivity according to information perceived about the environment. Learning plays a crucial role in the functionality of the system.

The proposed model is tested for solving various instances of $\mathcal{NP}$-hard problems: the *Generalized Traveling Salesman Problem (GTSP)*, the *Asymmetric Traveling Salesman Problem (ATSP)* and a dynamic version of *GTSP*. Numerical results and comparisons indicate the potential of the proposed system.

The structure of the paper is as follows: stigmergic agents are introduced based on a brief review of both multi-agent systems and ant colony systems, agent sensitivity to pheromone trails is explained, the learning mechanism for agents is presented and the proposed model and algorithm are detailed. Numerical experiments and some final remarks conclude the paper.

2 Stigmergic Agents

Agents of the proposed model are able to communicate both directly and in a stigmergic manner using pheromone trails produced by agents. Direct communication is inspired by the paradigm of multi-agent systems while stigmergic communication is based on swarm intelligence behaviour such as that of ant systems.

2.1 Multi-Agent Systems

Characterized by computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility and reuse, multi-agent systems (MAS) promote conceptual clarity and simplicity of design [13, 19]. A multi-agent approach to developing complex systems involves the employment of several agents capable of interacting with each other to achieve objectives [2, 6, 22, 23]. The benefits of such an approach include the ability to solve large and complex problems, interconnection and interoperation of multiple existing legacy systems and the capability to handle domains in which the expertise is distributed [2, 13, 14].

Interoperation among autonomous agents of MAS is essential for the successful location of a solution to a given problem [6, 19]. Agent-oriented interactions span from simple information interchanges to planning of interdependent activities for which cooperation, coordination and negotiation are fundamental. Coordination is necessary in MAS because agents have different and limited capabilities and expertise [14]. Agents have to coordinate their activities in order to determine the organizational structure in a group of agents and to allocate tasks and resources. Furthermore, interdependent activities require coordination (the action of one agent might depend on the completion of a task for which another agent is responsible). Negotiation is essential within MAS for conflict resolution and can be regarded as a significant aspect of the coordination process among autonomous agents [2, 14].

Agents need to communicate in order to exchange information and knowledge or to request the performance of a task as they only have a partial view over their environment [6, 22]. Considering the complexity of the information resources exchanged, agents should communicate through an agent communication language (ACL). Standard ACLs designed to support interactions among intelligent software agents include the Knowledge

Query and Manipulation Language (KQML) proposed by the Knowledge Sharing Effort consortium [10] and FIPA ACL defined by the FIPA organization [25]. Both KQML and FIPA ACLs are designed to be independent of particular application vocabularies.

2.2 Ant Systems

Self-organization [3] and indirect interactions between individuals in a system make possible the identification of intelligent solutions to complex problems. These indirect interactions occur when one individual modifies the environment and other individuals respond to the change at a later time. This process refers to the idea of stigmergy [11] which stays at the core of any ant system model. The indirect interactions in an ant system are based on pheromone trails being inspired by the real behaviour of ants. This approach led to the development of a metaheuristic that has been successfully used to solve combinatorial optimization problems [12].

Ant algorithms are based on the following main ideas [1, 9]:

- Each tour detected by an ant is associated with a candidate solution for a given problem.
- The amount of pheromone deposited on a tour is proportional to the quality of the corresponding candidate solution for the target problem.
- When an ant has to choose between two or more nodes, the edge having higher amount of pheromone has a greater probability of being chosen.

The *Ant Colony System (ACS)* metaheuristic [7] is a particular class of ant algorithms. The behaviour of insects is replicated to the search space. The decisions of the ants regarding the edge to follow are influenced by the corresponding amount of pheromone. Stronger pheromone trails are preferred and the most promising tours receive a greater pheromone trail in time. The result of an *ACS* algorithm is the shortest tour found - potentially corresponding to the optimal or a near-optimal solution of the given problem.

Let α and β be parameters used for tunning the relative importance of edge length in selecting the next node. Let us denote by ${J^k}_i$ the unvisited successors of node i by ant k and $u \in {J^k}_i$. q is a random variable uniformly distributed over $[0, 1]$ and q_0 is a parameter in the unit interval.

The probability p_{iu} of choosing $j = u$ as the next node if $q > q_0$ (the current node is i) is defined as [7]:

$$p_{iu}(t) = \frac{[\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta}}{\Sigma_{o \in {J^k}_i}[\tau_{io}(t)]]^{\alpha}[\eta_{io}(t)]^{\beta}}, \tag{1}$$

where

$\tau_{iu}(t)$ refers to the pheromone trail intensity on edge (i, u) at time t, and

$\eta_{iu}(t)$ represents the visibility of edge (i, u).

If $q \leq q_0$ the next node j is chosen according to the following rule [7]:

$$j = argmax_{u \in J_i^k} \{[\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta}\}. \tag{2}$$

Well known and robust algorithms include *Ant Colony System* [7] and *MAX-MIN* Ant System [21].

2.3 Hybrid Behaviour of Stigmergic Agent Systems

Agents of the proposed model are able to exchange different types of messages in order to share knowledge and support direct interoperation. The content of the messages exchanged refers to environment characteristics and partial solutions obtained. The information about dynamic changes in the environment is of significant importance in the search process. The introduced model inherits agent properties such as autonomy, reactivity, learning, mobility and pro-activeness used in multi-agent systems [13, 15, 23]. The agents that form the system have the ability to operate without human intervention, can cooperate to exchange information and can learn while acting and reacting in their environment.

Furthermore, agents are endowed with the ability to produce pheromone trails that can influence future decisions of other agents within the system. The idea of stigmergic agents was introduced in [5] where a system composed of stigmergic agents is outlined and illustrated by an example. The stigmergic behaviour of agents is similar to that of the ants in the bio-inspired ACS metaheuristic [8, 9].

Every time a decision needs to be made, an agent will first choose one of the strategies available: direct communication or stigmergic communication. If direct communication strategy is selected then the agent will make the decision based on the information about the environment and partial solutions received directly from other agents. If stigmergic strategy is used then the agent will make the decision based on the ACS model using the rules given in (1), (2).

3 Sensitivity in the Proposed Stigmergic Agent Model

Sensitivity in ant systems refers to the idea that not all ants react in the same way to the pheromone trails [4]. In a sensitive ant model, agents are heterogeneous as they are endowed with different levels of sensitivity to pheromone. This variable sensitivity can potentially induce various types of reactions to a changing environment. A better balance between search diversification and search exploitation can be achieved by combining stigmergic communication with heterogeneous agent behaviour.

The model of sensitive ants described in [4] is engaged in the proposed stigmergic agent model for guiding the agent in the selection of the strategy (direct or stigmergic communication)to be used when a decision has to be made.

Within the proposed model each agent is characterized by a pheromone sensitivity level denoted by PSL which is expressed by a real number in the unit interval $[0, 1]$. Extreme situations are:

- If $PSL = 0$ the agent completely ignores stigmergic information;

- If $PSL = 1$ the agent has maximum pheromone sensitivity.

Small PSL values indicate that the agent will normally choose very high pheromone levels moves (as the agent has reduced pheromone sensitivity). These agents are more independent and can be considered environment explorers. They have the potential to autonomously discover new promising regions of the solution space. Therefore, search diversification can be sustained.

Agents with high PSL values will choose any pheromone marked move. Agents of this category are able to intensively exploit the promising search regions already identified. In this case the agent's behaviour emphasizes search intensification.

4 Learning in the Proposed Stigmergic Agent Model

Agents of the proposed model can learn to adapt their PSL according to the environment characteristics (and based on previous experience) furthermore facilitating an efficient and balanced exploration and exploitation of the solution space.

4.1 Learning as a Search Mechanism

The initial PSL values are randomly generated. During their lifetime agents may improve their performance by learning. This process translates to modifications of the pheromone sensitivity. The PSL value can increase or decrease according to the search space topology encoded in the agent's experience. Low sensitivity of agents to pheromone trails encourages a good initial exploration of the search space. High PSL values emphasize the exploitation of previous search results. Several learning mechanisms can be engaged at individual or global level. A simple reinforcing learning mechanism is proposed in the current model. According to the quality of the detected solution, the PSL value is updated for each agent.

Agents with high PSL value (above a specified threshold τ) are environment exploiters and they will be encouraged to further exploit the search region by increasing their PSL value each time a good solution is determined. Agents with small PSL value are good explorers of the environment and good solutions will be rewarded by decreasing agent PSL value (emphasizing space exploration).

4.2 The Learning Rule

Let $PSL(A,t)$ denote the PSL value of the agent A at iteration t and $S(A,t)$ the solution detected. The best solution determined by the system agents (until iteration t) is denoted by $Best(t)$. The proposed learning mechanism works as follows:
Case 1: $PSL(A,t) > \tau$

- If $S(A,t)$ is better than $Best(t)$ then A is rewarded by increasing its PSL value according to the following learning rule:

$$PSL(A,t+1) = min(1, PSL(A,t) + exp(-PSL(t))/(t+1)^2). \tag{3}$$

- If $S(A,t)$ is worse than $Best(t)$ then A is 'punished' by decreasing its PSL value according to the following learning rule:

$$PSL(A,t+1) = max(0, PSL(A,t) - exp(-PSL(t))/(t+1)^2). \tag{4}$$

Case 2: $PSL(A,t) \leq \tau$

- If $S(A,t)$ is better than $Best(t)$ then A is rewarded by decreasing its PSL value according to the following learning rule:

$$PSL(A,t+1) = max(0, PSL(A,t) - exp(-PSL(t))/(t+1)^2). \tag{5}$$

- If $S(A,t)$ is worse than $Best(t)$ then A is 'punished' by increasing its PSL value according to the following learning rule:

$$PSL(A,t+1) = min(1, PSL(A,t) + exp(-PSL(t))/(t+1)^2). \tag{6}$$

Agents learn the characteristics of the search space via a dynamic change in the PSL values. Good explorers of the solution space will be encouraged to more aggressively further explore the environment. Promising solutions already identified will be further exploited by rewarding the corresponding agent.

5 Learning Sensitive Stigmergic Agent System Model

The learning sensitive stigmergic agent-based model is based on a set of interacting agents characterized by different sensitivity levels (see Section 3) and enabled with learning properties (see Section 4).

5.1 Model General Description

The proposed model is initialized with a population of agents that have no knowledge of the environment characteristics. Each path followed by an agent is associated with a possible solution for a given problem. Each agent deposits pheromone on the followed path and is able to communicate to the other agents in the system the knowledge it has about the environment after a full path is created or an intermediary solution is built.

The infrastructure evolves as the current agent that has to determine the shortest path is able to make decisions about which route to take at each point in a sensitive stigmergic manner and based on learn information. Agents with small PSL values will normally choose only paths with very high pheromone intensity or alternatively use the knowledge base of the system to make a decision. These agents can easily take into account ACL messages received from other agents. The information contained in the ACL message refers to environment characteristics and is specific to the problem that is being solved. On the other hand, agents with high PSL values are more sensitive to pheromone trails and easily influenced by stronger pheromone trails. However, this does not exclude the possibility of additionally using the information about the environment received from other agents.

5.2 The Algorithm

The algorithm of the proposed model is sketched below.

```
Algorithm Learning Sensitive Stigmergic Agent System
Begin
Set parameters
Initialize pheromone trails
Initialize knowledge base
While stop condition is false
Begin
  Activate a set of agents
  Place each agent in search space
  Do - For each agent
    Apply a state transition rule to incrementally build a solution.
    Determine next move (stigmergic strategy / direct communication)
    Apply a local pheromone update rule.
    Propagate learned knowledge.
  Until all agents have built a complete solution
  Update $PSL$ value for each agent using learning mechanism according
to (3), (4), (5) and (6).
  Apply a global pheromone update rule
  Update knowledge base (using learned knowledge).
End While
End.
```

After a set of agents determines a set of problem solutions, the proposed model allows the activation of another set of agents with the same objective but having some knowledge about the environment. The initial knowledge base of each agent refers to the information about the path previously discovered by each agent.

6 NP-hard Problems Solved using the Proposed Model

The proposed model has been tested for solving various instances of the well known $\mathcal{NP}$-hard Traveling Salesman Problem. This section presents the numerical results obtained for the *Generalized Traveling Salesman Problem (GTSP)* and *Asymmetric Traveling Salesman Problem (ATSP)*. Furthermore, the performance of the model in changing environments is tested on a dynamic version of *GTSP*.

The numerical experiments and comparisons reported in this section emphasize the potential of the proposed approach to address complex problems and facilitate further connections between multi-agent systems and nature inspired computing.

6.1 Numerical Experiments for GTSP

GTSP is a generalized version of the $\mathcal{NP}$-hard problem TSP. Let $G = (V, E)$ be an n-node undirected graph with edges associated with non-negative costs. Let $V_1, ..., V_p$ be a partition of V into p subsets called clusters. *GTSP* refers to finding a minimum-cost tour

H spanning a subset of nodes such that H contains exactly one node from each cluster $V_i, i = 1, ..., p$.

To address the *GTSP*, the proposed computational model allows agents to deposit pheromone on the followed path. Unit evaporation takes place each cycle. This prevents unbounded intensity trail increasing. The system is implemented using sensitive stigmergic agents with low initial PSL values.

The performance of the proposed model in solving *GTSP* is compared to the results of classical *Ant Colony System (ACS)* technique, the *Nearest Neighbor (NN)* algorithm, the GI^3 composite heuristic [18] and *Random Key Genetic Algorithm (rkGA)* [20]. The algorithm *Ant Colony System* for *GTSP* [16] is based on the ACS [8, 9] idea of simulating the behaviour of a set of agents that cooperate to solve a problem by means of simple communications. In *Nearest Neighbor* algorithm the rule is always to go next to the nearest as-yet-unvisited location. The corresponding tour traverses the nodes in the constructed order. The composite heuristic GI^3 is composed of three phases: the construction of an initial partial solution, the insertion of a node from each non-visited node-subset, and a solution improvement phase [18]. The *Random Key Genetic Algorithm* combines a genetic algorithm with a local tour improvement heuristic. Solutions are encoded using random keys, which circumvent the feasibility problems encountered when using traditional GA encodings [20].

The data set of Padberg-Rinaldi city problems (TSP library [24]) is considered for numerical experiments. $TSPLIB$ provides the optimal objective values (representing the length of the tour) for each problem. Comparative results obtained are presented in Table 1.

Problem	Opt.val.	NN	GI^3	ACS	rkGA	Proposed Model
16PR76	64925	76554	**64925**	**64925**	**64925**	**64925**
22PR107	27898	28017	**27898**	27904.4	**27898**	**27898**
22PR124	36605	38432	36762	36635.4	**36605**	**36605**
28PR136	42570	47216	43117	42593.4	**42570**	**42570**
29PR144	45886	46746	**45886**	46033	**45886**	**45886**
31PR152	51576	53369	51820	51683.2	**51576**	**51576**
46PR226	64007	68045	**64007**	64289.4	**64007**	**64007**
53PR264	29549	33552	29655	29825	**29549**	29549.2
60PR299	22615	27229	23119	23039.6	22631	**22628.4**
88PR439	60099	67428	62215	64017.6	60258	**60188.4**

Table 1: Numerical results for the Padberg-Rinaldi *GTSP* data set

The proposed model gives the optimal solution for 7 out of the 10 problems engaged in the numerical experiments. For two other problems, the solutions reported by the proposed model are very close to the optimal value and better than those supplied by the other methods considered.

For statistical analysis purposes, the Two Paired Sample Wilcoxon Signed Rank Test is engaged on the Padberg-Rinaldi *GTSP* results (presented in Table 1) for comparing ACS and the proposed model. The results of this test are indicated in Table 2.

Wilcoxon Signed-Rank Statistic	45,000
E(W+), Wilcoxon Signed-Rank Score	22,500
Var(W+), Variance of Score	71,250
Wilcoxon Signed-Rank Z-Score	2,666
One-Sided P-Value	0,004
Two-Sided P-Value	0,008

Table 2: Wilcoxon Signed-Rank Statistic for Padberg-Rinaldi *GTSP* results

The observed value of $z = +2.666$ is significant on a .004 level for a directional test and on a .008 level for a two-tailed non-directional test.

6.2 Numerical Experiments for ATSP

The *Asymmetric TSP (ATSP)* is a variant of the $\mathcal{NP}$-hard problem TSP characterized by the fact that the cost of an edge is not the same with the cost of the inverse edge.

The proposed model for solving *ATSP* is implemented using sensitive stigmergic agents with initial randomly generated PSL values. Sensitive-explorer agents autonomously discover new promising regions of the solution space to sustain search diversification. Each generation the PSL values are updated according to the reinforcing learning mechanism described in Section 4. The learning rule used ensures a meaningful balance between search exploration and exploitation in the problem solving process.

The performance of the proposed model in solving *ATSP* is compared to the results of classical *ACS* technique and the *Max-Min Ant System (MMAS)* [21]. Several problem instances from TSP library [24] are considered for numerical experiments. Comparative results obtained are presented in Table 3. The parameters of the algorithm are similar to those of ACS: ten ants are used and the average of the best solutions is calculated for ten successively runs.

The proposed model detects a near-optimal or optimal solution for all the ATSP problems engaged in the numerical experiments. For one of the problem instances, all three methods compared find the optimal solution. For the other instances, solutions detected by the proposed model are very close to the optimal value and better than those supplied by the other methods considered.

Problem	Opt.val.	ACS	MMAS	Proposed Model
Ry48p	14422	**14422**	**14422**	**14422**
Ft70	38673	38781	38690	**38682**
Kro124	36230	36241	36416	**36238**
Ftv170	2755	2774	2787	**2755**

Table 3: Numerical results for solving *ATSP*

Problem	ACS	Proposed Model
pr76	**58782.40**	59087.40
pr107	28028.20	**27979.60**
pr124	37176.60	**37157.20**
pr136	45479.50	**44754.20**
pr144	37176.60	**37157.20**

Table 4: Numerical results for solving a dynamic variant of the Padberg-Rinaldi *GTSP* data set

6.3 Numerical Experiments for Dynamic GTSP

The dynamic version of GTSP engaged in this set of numerical experiments refers to changing the number of cities (nodes) that need to be covered dynamically during runtime. More specifically, one randomly chosen node is blocked at each iteration [17]. As opposed to standard ant-based models, the agents of the proposed model are able to directly communicate the change in the enironment (i.e. disappearance of a city) and react appropiately. The results of the proposed model are compared to that of the ACS technique (reported in [17]).

Paramater setting is the following: $\alpha = 1$, $\beta = 5$, $\rho = 0.1$, $q_0 = 0.9$, the initial value of the pheromone trail is considered to be 0.01 and the population size is ten.

Table 4 presents the average results over ten runs of the ACS and proposed algorithm for each problem instance considered.

The numerical results obtained for the dynamic version of GTSP indicate the importance of direct communication related to environment changes. The proposed model obtains overall better results compared to the ACS model indicating the potential of hybrid agent behaviour (stigmergic and direct interaction) in models addressing dynamic optimization problems.

7 Conclusions

An agent-based approach to combinatorial optimization has been proposed. The components of a multi-agent system are endowed with a supplementary capacity - the ability of communication by environmental changes. Agents adopt a stigmergic behaviour (being able to produce pheromone trails) to identify problem solutions and use direct communication to share knowledge about the environment.

The primary elements of the proposed model refer to communication, stigmergy, sensitivity and learning. The hybrid behaviour of the agent is modulated by the individual sensitivity level facilitating a balance between the strategies of direct and stigmergic communication used in making decisions. Different sensitivity levels for each agent and the learning mechanism able to adapt the sensitivity lead furthermore to a potential space exploration/exploitation equilibrium benefic to the search process. Numerical comparative results indicate the effectiveness and the potential of the proposed technique.

References

[1] E. Bonabeau, M. Dorigo, G. Tehraulaz, Swarm intelligence from natural to artificial systems, Oxford, UK: Oxford University Press, 1999.

[2] J.M. Bradshow, An Introduction to Software Agents, in Software Agents, J.M. Bradshow, MIT Press, 1997.

[3] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, *Self organization in biological systems*, Princeton Univ. Press, 2001.

[4] C. Chira, D. Dumitrescu, C-M. Pintea, Heterogeneous Sensitive Ant Model for Combinatorial Optimization, Genetic and Evolutionary Computation Conference GECCO'08, ACM, pp. 163-164, July 12-16, 2008, Atlanta, Georgia, USA

[5] C. Chira, C-M. Pintea, D. Dumitrescu, Sensitive Stigmergic Agent Systems, *Adaptive and Learning Agents and Multi-Agent Systems (ALAMAS)*, Maastricht, The Netherlands, 2 & 3 April 2007, MICC Technical Report Series 07-04 (Karl Tuyls, Steven de Jong, Marc Ponsen, Katja Verbeeck Eds.), pp. 51-57, 2007.

[6] O. Chira, C. Chira, D. Tormey, A. Brennan, T. Roche, An Agent-Based Approach to Knowledge Management in Distributed Design, Special issue on E-Manufacturing and web-based technology for intelligent manufacturing and networked enterprise interoperability, Journal of Intelligent Manufacturing, Vol. 17, No. 6, 2006.

[7] M. Dorigo, L.M. Gambardella, Ant Colony System: A cooperative learning approach to the traveling salesman problem, IEEE Trans. on Systems, Man, and Cybernetics. 26, 1996, pp. 29-41.

[8] M. Dorigo, G. Di Caro, L. M. Gambardella, Ant algorithms for discrete optimization, *Artificial Life*, 5, pp. 137-172, 1999.

[9] M. Dorigo, C. Blum, Ant Colony Optimization Theory: A Survey, *Theoretical Computer Science*, 344, 2-3, pp. 243-278, 2005.

[10] T. Finin, Y. Labrou, J. Mayfield, Kqml as an Agent Communication Language, in Software Agents, B.M. Jeffrey, Ed., MIT Press, 1997.

[11] P.-P. Grasse, La Reconstruction du Nid et Les Coordinations Interindividuelles Chez Bellicositermes Natalensis et Cubitermes sp. La Thorie de la Stigmergie: Essai d'interpretation du Comportement des Termites Constructeurs, *Insect Soc.*, 6, pp. 41-80, 1959.

[12] M. Guntsch, M. Middendorf, Pheromone modification strategies for ant algorithms applied to dynamic TSP. In Applications of Evolutionary Computing: Proceedings of EvoWorkshops 2001. Springer Verlag, 2001.

[13] N. R. Jennings, An agent-based approach for building complex software systems, *Comms. of the ACM*, 44, 4, pp. 35-41, 2001.

[14] H. Nwana, L. Lee, N. Jennings, Coordination in Software Agent Systems, BT Technology Journal, Vol. 14, No. 4, 1996, pp. 79-88.

[15] H. S. Nwana, Software Agents: An Overview, *Knowledge Engineering Review*, 11, pp. 1-40, 1996.

[16] C-M. Pintea, P.C. Pop, C. Chira, Reinforcing Ant Colony System for the Generalized Traveling Salesman Problem, Proc. BIC-TA, vol. Evolutionary Computing, 245-252, 2006.

[17] C-M. Pintea, P.C. Pop, D. Dumitrescu, An Ant-based Technique for the Dynamic Generalized Traveling Salesman Problem, Proc. of the 7-th WSEAS Int. Conf. on Systems Theory and Scientific Computation, pp. 257-261, 2007

[18] J. Renaud, F.F. Boctor, An efficient composite heuristic for the Symmetric Generalized Traveling Salesman Problem, *European Journal of Operational Research*, 108, pp. 571-584, 1998.

[19] S. Russel, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2nd edition, 2002.

[20] L. V. Snyder, M. S. Daskin, A Random-Key Genetic Algorithm for the Generalized Traveling Salesman Problem, *European Journal of Operational Research*, pp. 38-53, 2006.

[21] T. Stützle and H.H. Hoos, Max-Min Ant System, Future Generation Computer Systems, 16, 9, 2000, pp. 889-914.

[22] M. Wooldrige, An Introduction to Multiagent Systems, Wiley, 2002.

[23] M. Wooldridge, P. E. Dunne, The Complexity of Agent Design Problems: Determinism and History Dependence, *Annals of Mathematics and Artificial Intelligence*, 45, 3-4, pp. 343-371, 2005.

[24] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, TSPLIB - library of sample instances for the TSP (and related problems).

[25] http://www.fipa.org, Foundation for Intelligent Physical Agents.

[26] http://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test, Wilcoxon signed-rank test.

C. Chira, C-M. Pintea and D. Dumitrescu
Babes-Bolyai University
Department of Computer Science
M. Kogalniceanu 1, 400084 Cluj-Napoca, Romania
E-mail: {cchira, cmpintea, ddumitr}@cs.ubbcluj.ro

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 63-83

Conceptual Model of an Intelligent Supervising System for Evaluating Policies in the Operation of an Underground Mine

Felisa M. Cordova and Luis E. Quezada

Abstract: This work presents a conceptual model for the design of an Intelligent Supervising System (ISS) built to support the scheduling for an underground mine in order to supervise its operation. The system is composed by a Simulation Model (SM) linked to a Knowledge Based System (KBSM) designed by means of hierarchical, colored and temporal Petri Nets. Simulation Model allows simulating the operation of the production, reduction and transport levels in the mine. Knowledge Based System is activated by events produced in daily operations and yields the results of registered events and the actions taken to solve the problem, generating operation rules. The proposed model allows different types of mine operations and scenarios providing data for decision-making. The system helps to evaluate different policies for programming the activities in the mine thus seeking to improve the equipment productivity. The model also allows the feasibility assessment of the Daily Master Plan based on the input data of the simulation model.

Keywords: Intelligent System, Simulation Model, Knowledge Based System, Petri Nets, Underground mining.

1 Introduction

The main operations of underground mining industry over the last decade of the twentieth century have been designed to automate the extractive processes withdrawing operators from contaminated and unsafe places inside the mine. In order to achieve this objectives tele-operation and robotization of equipment were introduced in underground mine operations [1],[2]. The optimization of operation management has been obtained by moving from a push to a pull system approach or by using heuristics to produce an optimal schedule. In this way, pseudo intelligent simulation models can be used to generate policies to re-schedule the activities in real time [1],[11].

During daily operations of the mine there are continuous events taking place which alter the normal work cycle of the mine, affecting its activities in their diverse levels or resources [1, 6, 11]. Observation and analysis of their behavior and the effects of these events thereafter in daily mine operations is of the most importance, as shown in other applications [6]. In this context, it is necessary to simulate the behavior of the underground mine, whose continually functioning does not allow it to be available for its study, thus providing in this way new alternatives to improve its productivity, its efficiency and efficacy.

Petri Networks are being used for modelling dynamic operation of discrete systems in different domains [4, 5, 7] mainly in manufacturing [4, 8, 9]. They are also utilized like a very useful tool for modeling, to analyze, to simulate and to control production systems [10, 12]. Petri Nets are a good tool to model production systems, emulating parallel and concurrent systems, without the need of building a simulation model for each system [3, 4, 7]. It allows modeling the production process of an underground mine because it is possible to build various independent modules and then to produce a given configuration as a combination of those modules. The advantage of Petri Nets over a discrete-event simulation is that it is possible to build various models as independent modules and then to produce a given configuration as a combination of those modules [9, 10]. In this way, a number of basic networks representing different elements of the mine are built and then they are combined to represent the whole system.

A Petri Net is defined by a 7-tuple $\{L, T, V, I, O, M, m_0\}$, where:

L: $\{l_1, l_2, l_3, l_4, ..l_n\}$, finite set of places non-empty, $n \geq 0$.

T: $\{t_1, t_2, t_3, t_4, ..t_m\}$, finite set of transition not empty, $m \geq 0; V : Values\{0, 1\}$

I: Binary function used to determine connections from places to transitions.

So, $I : L \times T \rightarrow V$ and if $Il, t = 1$, the place l is connected to transition t, otherwise there is no connection.

S: Binary function used to determine which transitions are connected to which places.

So, $S : T \times L \rightarrow V$ and if it exists a connection from transition t to the place l, if and only if $St, l = 1$.

M: Set of Tokens: $\{0, 1, 2, 3, \ldots, m\}$

m_0: Initial token function, $m_0 : L \rightarrow N$

$L \bigcap T = 0;\ I, O :\ T \rightarrow L$

2 Mine processing cycle and the Production System

In the underground mine processing cycle, the rocks in higher levels are reduced in size. However, the transition from secondary rock to primary rock creates a problem of inefficiency causing a fall of productivity up to 10 % . In this context, the underground mine companies use a mechanized method of exploitation called “Block Caving” which is shown in Figure 1. The production system consists basically of three main levels:

Production level: in this level the chunks of material are taken 18 meters down the ground level until the collector shafts called ore-passes. Then, the material is loaded from ore-passes, transported and dumped into pits by Load-Haul-Dump (LHD) vehicles.

Reduction level: this level is located 35 meters under the production level, where the mineral falls into the rock breaker chambers. Here a rock breaker reduces its granulation to less than a cubic meter and then it follows its gravitational movement. The reduction is necessary because some chunks of mineral, which are too big, can cause a bog inside the transport chimneys, and they can also damage the train wagons of the inferior level.

Transport level: here, the mineral that comes down from the rock breaker chambers is loaded into mailboxes and finally into trains wagons and it's taken by trains (crossed) to the processing areas where copper is detached from barren material.

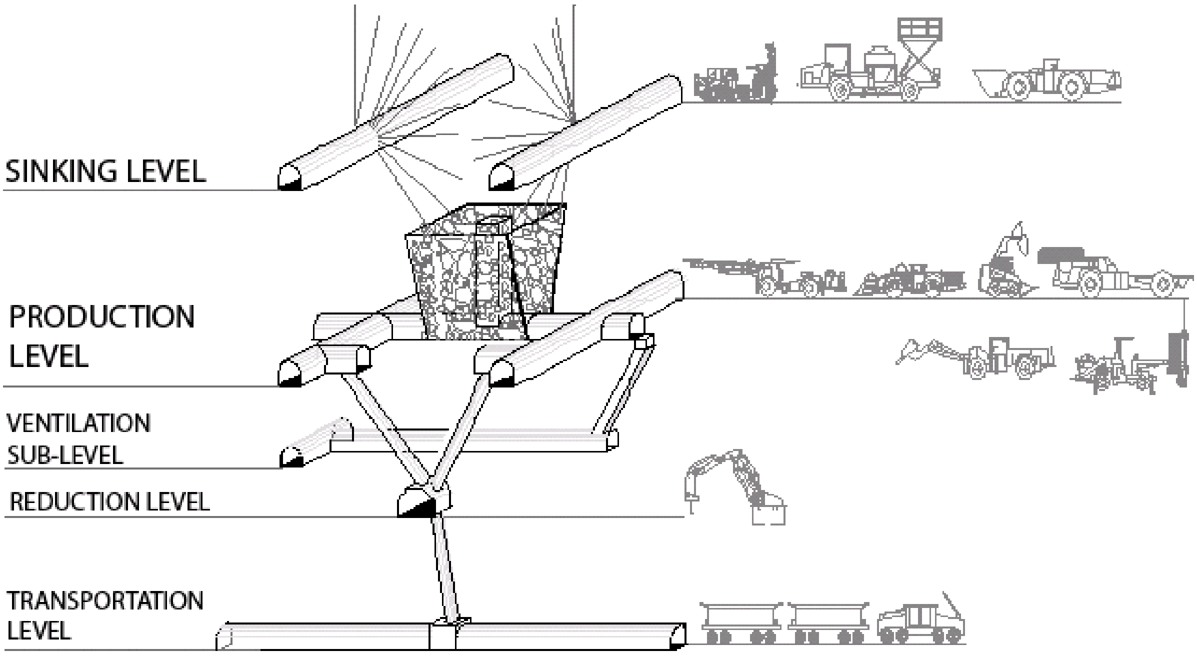


Figure 1: Block Caving Exploitation Method.

Initially, the Production Master Plan worked in a push mode, not taking into consideration the supervision and effective coordination of the levels previously described. It means that each level processed the material as soon as it cames from the level immediately above. The coordination of activities among levels was done from the highest to the lowest level (one by one), taking into account the availability of equipment in each level, without any visual information of them.

The actual Daily Master Plan defines the operations for each available resource according to the production goals, also indicates the extraction points, tonnage to be extracted and availability of pits and ore-passes. Simulated scenarios allow the simulation of a working shift in the mine under operational conditions without reprogramming. The system designed provides recommendations regarding actions to be taken, whenever some unexpected event happened. Reports reflect a summary of the outcomes and failures which occurred, also recommendations delivered by the system and the final actions taken in the process. Reprogramming orders will be executed whenever a major outcome is registered and the needs to reprogram activities to achieve the daily goals are detected. Generation of new operational rules is activated when some outcome is registered revealing no final solution. Events or outcomes that perturb the established working program can force to modify it to achieve the production goals.

The main variables involved in the generation of the Daily Master Plan are:

a) copper law searching attractive exploitation areas;

b) arsenic law;

c) extraction speed rate: is determined by the LHD capacities;

d) Monthly tons. to be extracted;

e) availability of infrastructure at any level;

f) failure of any equipment at the subsequent level will affect the actual level of operation;

g) time schedule of the daily program not including non scheduled events which could results in failing the expected extraction.

3 The Intelligent Supervising System

The Intelligent Supervising System (ISS) for the operation of the mine is composed by a Simulation Model (SM) and a Knowledge Based System Model (KBSM) integrated in a single model. The simulator, which is being fed by the Daily Master Plan, allows processing and simulation of the different scenarios that may take place during daily mine operations, utilizing as a basis the historical data of the various shifts. The Knowledge Based System provides the knowledge acquired by experts on the operation of all resources that participate in the productive process, as well as the possible failures of those resources. This allows generating events, solutions, and new operating rules for the system identifying critical failures for the production, reduction and transport levels. The architecture of the Intelligent Supervising System is presented in Figure 2.

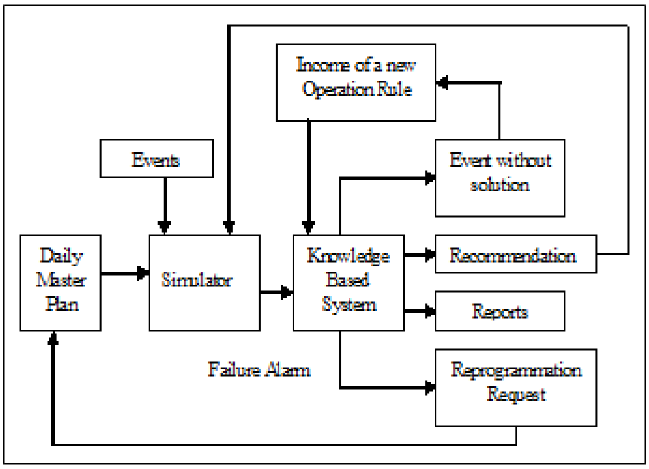


Figure 2: Architecture of Intelligent Supervising System.

The operation, inputs and results of the Intelligent Supervising System are detailed as follows:

Daily Master Plan: it is a set of rules which define the number of bucket mineral operations for each available resource (workers and teams), also indicates the extraction points, tons. to be extracted, availability of draining channels, emptying points and the Copper Law within the sector. These rules are established according to the annual production goals which are defined by the mine itself.

Events or outcomes: an outcome is defined as an event which perturbs the established working program which can force to modify it to achieve the production goals. There are a variety of events which can be associated to a working level within the mine or an event within a working team. Whatever the paralyzed resource will be, the expected operation of the mine will be affected demanding the intervention of some expert who then will define what action should be taken to carry on with the planned production program.

Simulated Scenarios: corresponds to the simulation of a working shift in the mine under normal operational conditions without reprogramming, step which is executed by introducing the Daily Master Plan information to the Simulation Module.

Outcome without solution: outcome which does not generate a recommendation or solution defined in the knowledge base of the KBSM consequently yielding the storage of a new outcome as a case, adding new knowledge or a solution which generates new operational rules.

Recommendation: advisability regarding actions to be taken, whenever some unexpected event happened within the mine. The knowledge stored in the KBSM yields the appropriate indication to be followed.

Reports: these reports will reflect a summary of the outcomes and flaws which occurred, recommendations delivered by the system and the final actions taken in the process. Reprogramming orders: these orders will be executed whenever a major outcome is registered and the needs to reprogram activities to achieve the daily goals are detected.

Generation of new operational rules: this step is activated when some outcome is registered revealing no final solution. It will be stored in the KBSM which by means of interactions with the simulation module will yield a new operational rule which can then resolve the event.

3.1 The Simulation Model(SM)

The proposed model describes the operation of load-haul-dump vehicles (LHDs) at the Production Level, also the operation of rock-breakers equipment at the Reduction Level from the beginning of a shift up to its end. LHDs vehicles moves through the tunnels inside the mine according to a given schedule, which will be affected by different events such as own breakdowns, rock-breakers breakdowns, etc. It is comprised of three submodules: production, reduction and transport. The Production Level Module shown in Figure 3 initiates itself with the place denominated Generator LHD, from which tokens are activated which represent the LHD resources that contain the start up attributes of the simulation, yielded by the Daily Master Plan. Arriving to the street module, the token must identify the action that it will realize (entering or leaving street) once the token has continued on the indicated route, it must enter the first physical resource fixed in the mine and of the level production denominated Street ((C_1R to $C_{15}R$, C_1L to $C_{15}L$).

The Street Module is composed by three modular and flexible structures, which are combinable between them and are representative of each street: Ore-pass Module, Two

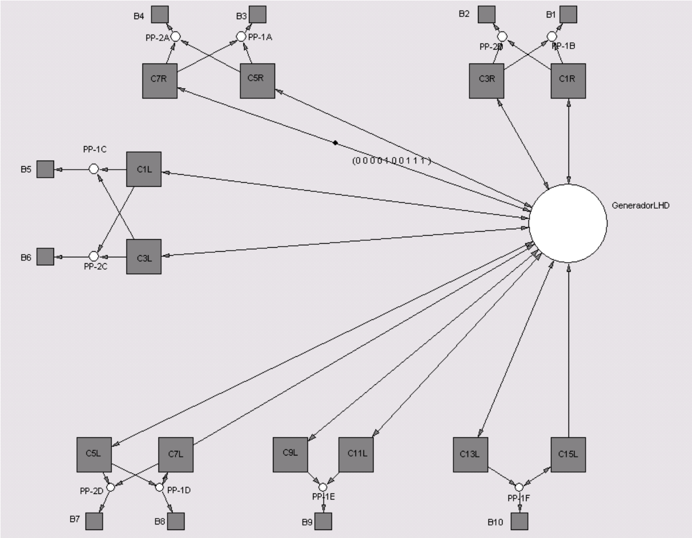


Figure 3: Production Level Module.

Ore-pass Module and Two Ore-pass-Pit Module, which communicate with the rest by means of a fourth, more complex structure denominated Routing Tree. Figure 4 shows in particular, the design of the Two Ore-pass Pit Module which adds new attributes to the initial module, which allow for the routing of the LHD through the totality of the module.

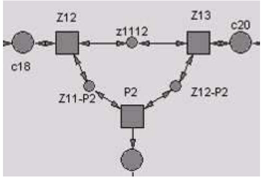


Figure 4: Two Ore-passes - Pit Module.

With the union of these structures it is possible to build the street resource. They have a different topology among them. Each module represented by Z_i presents a pair of parallel points of extraction, while each module Pi represents a pit. Next, the internal design of each one of the ore-passes is realized (Z_{12}, Z_{13}) and the pit (P_2) respectively. In Figure 5 it is possible to observe the configuration of physical fixed resource and Street Module. Each module represented by Z_i presents a pair of parallel points of extraction, while each module P_i represents an emptying Pit.

The internal design of each one of the Ore-pass is realized and the Pit respectively. Figure 6 shows the design of a Pit. Figure 7 shows the design of a Ore-pass.

The Routing Tree Module of the LHD has the function of directing the token or LHD through each one of the modules previously defined, by means of various alternative roads in order to comply with the tasks assignment of the Master Daily Plan. This one assigns to each token that enters and leaves the Master Daily Plan a unique vector of nine attributes $abcdsenbcal$ which contain a unique combination representing a unique movement to be followed by the LHD vehicle through those modules.

Ore-pass Module attributes:

a: Attribute which determines the events of loading or passing through the module.

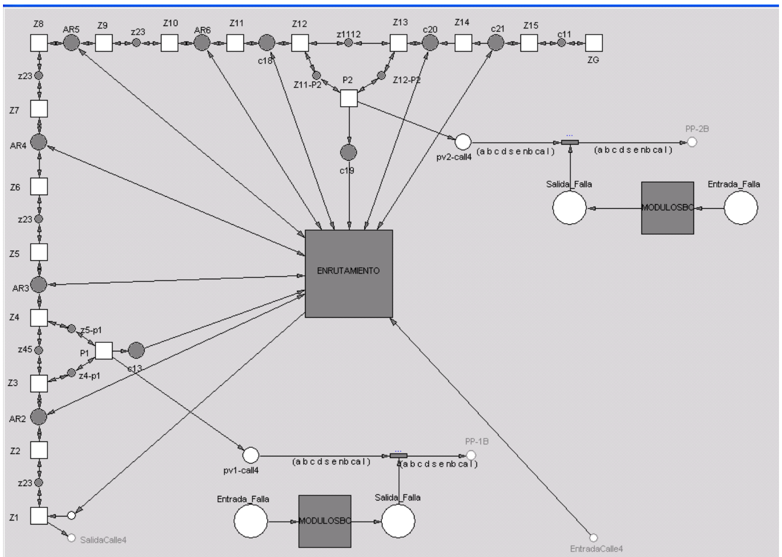


Figure 5: Physical Fixed Resource and Street Module.

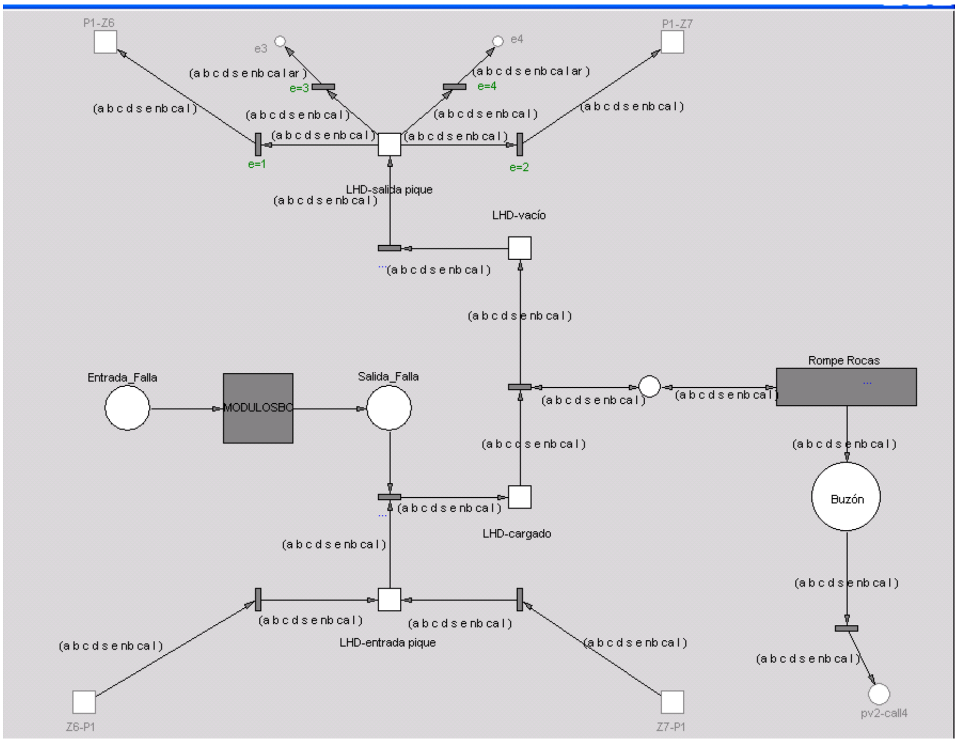


Figure 6: Pit Module.

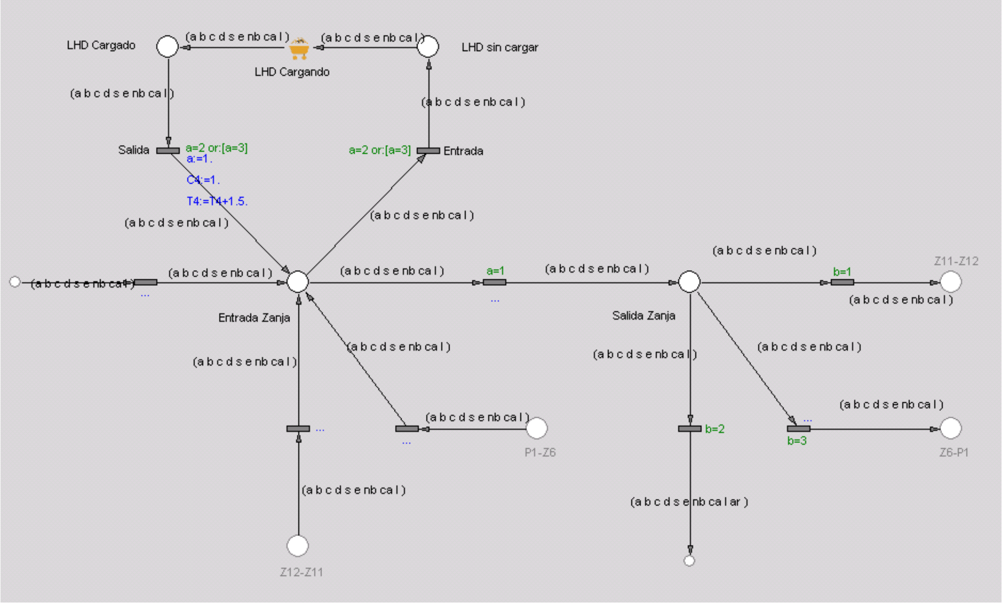


Figure 7: Ore-pass Module.

b: Attribute which determines the output sense of LHD.
Two Ore-pass Module attributes:
c: attribute which determines the events of loading or passing from Two Ore-pass Module.
d: attribute which determines the output sense of LHD from Two Ore-pass Module.
Two Ore-pass-Pit Module attributes:
a: attribute which determines the events of loading or passing from ore-pass 1 of ZPZ.
b: attribute which determines the output sense of LHD from ore-pass 1 of ZPZ.
c: attribute which determines the events of loading or passing from ore-pas 2 of ZPZ.
d: attribute which determines the output sense of LHD from ore-pass 2 of ZPZ.
e: attribute which determines the output sense of LHD from pit module.
nb: Number of bucket pending from the current instruction.
Movement of LHD attributes:
s: attribute giving the sense of movement for LHD vehicle.
ca: this attribute keeps the street number in where the LHD vehicle is running when it is passing through the tree.
l: is a binary attribute. When $b = 1$, the LDH vehicle must continue running by the street until the turning zone, then it must return to the ore-pass for loading.

The resources at the Reduction Level are the Rock-breaker and the Rock-breaker Operator. At this level the size reduction of material and minerals is realized which will be then transported to be processed at the plant. The resources rock-breaker and its operator are in charge of the reduction of the material extracted at the production level and which must be taken to the transport level.

The material coming from the Reduction Level is loaded and kept in Mailboxes, until it reaches its limit of storage capacity (280 tons.). Once this limit is reached, it opens its gates freeing the material to the Crossed resource, which in turn transports the material out of the mine for its later processing. In case the resource Mailbox presents any failure, the Mailbox Operators must aid and repair the Mailbox, so that it continues with its

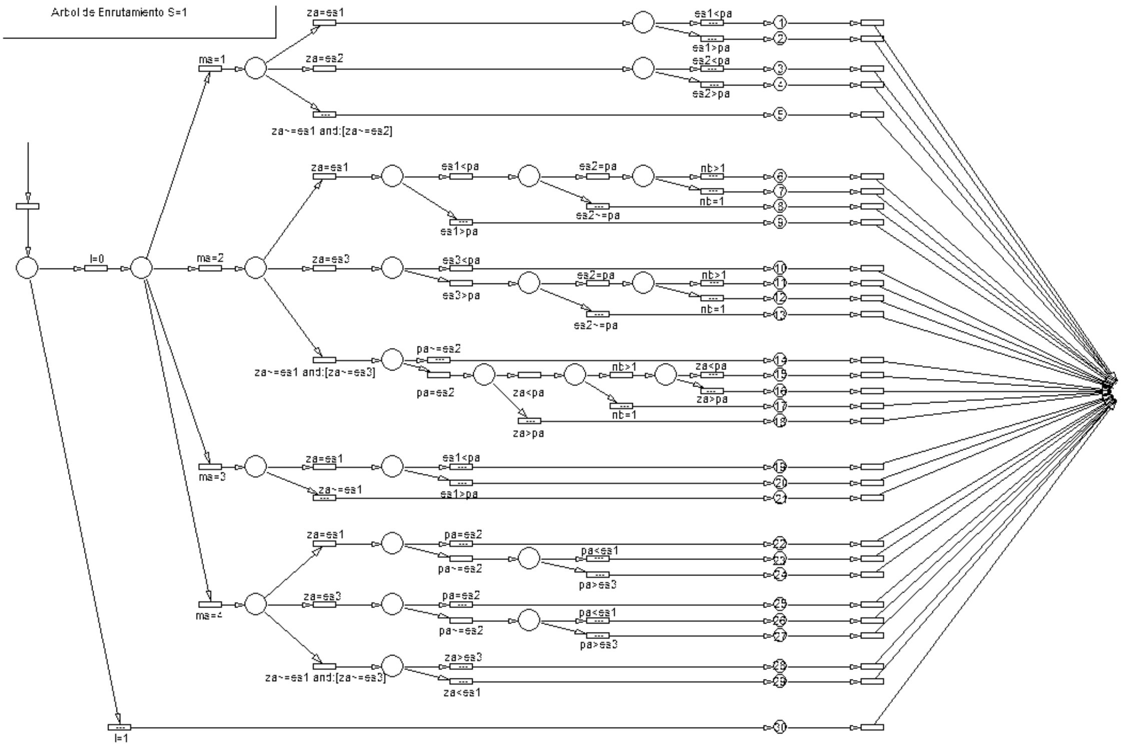


Figure 8: Routing Tree.

normal operation. In the model, each one of the LHD bucket represents a token therefore if a Mailbox concentrates 40 tokens, a new token will be generated in the Crossed. Figure 9 shows the link between Reduction and Transport Modules.

4 The Knowledge Based System Module

The Knowledge Based System (KBSM) is the engine of knowledge which allows the simulation module to include solutions to the failures that might have happened at each level. It consists of a modular structure which by means of a routing chart allows identification of the events or flaws which happen at each level. Its solution to the occurring problem at the expected production level has to represent the best solution to successfully continue any operation programmed by the system.

4.1 Knowledge Based System Input Module

The activation of the KBSM occurs when the initial transition KBS Input Module generates some token with numerical attributes allowing the association of a token to some particular failure. All token activated have some specific attribute “z” which describes the characteristics of the event occurred. This token generation was defined as a uniform probability which helps modeling the events or failures. Once the event has been identified together with its solution, it will leave the KBSM giving recommendations related to one of fifty six (56) failures that have been modeled. The events that happen at each level are associated to it frequency of occurrence per shift. According to this frequency

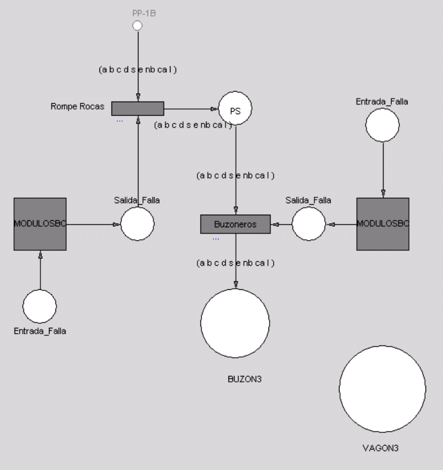


Figure 9: Transport Level Module linked to Reduction Level Module.

a numerical interval is associated to each level, resource or failure. Each token generated in the initial transition of the model is characterized by numerical values.

The KBS Input Module is connected with the Productivity Module and with the Human Resource Module by means of an entry called Failure Entry. Once it has looked over each branch of the knowledge system and once the event has been identified together with its solution, it will leaves the operational tree at the place called Fault Exit. This first structure can be observed in Figure 10. The modeling at different levels was achieved by means of hierarchical Petri networks. The colored ones allow representation and differentiation of events which call for a significant stopping of all operations.

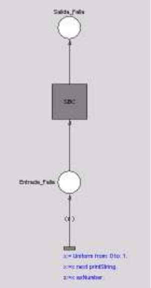


Figure 10: KBS Input Module.

The events or faults that happen at each level are associated to some frequency of occurrence per shift. The functioning of the Knowledge Based System Module is based on these frequencies. In fact the design of the event and fault generation was associated to a continuous probability interval, reason that explains why each token generated in the initial transition of the model, is characterized by numerical values. To determine if the failure which had been generated at the production level, reduction, transport or human resources, a series of connectors were built to allow identification of the characteristic numerical values of a token and associate it to a defined failure which originates at some level and some given resource. The values of these intervals correspond to the frequency of occurrence of each failure. The knowledge tree solving a failure step considers only one entry place which corresponds to the generated failure and only one exit which corresponds to the solution of the failure and its repercussion at the production level. Four probability intervals associated to the frequency occurrence of each failure at each level were defined. The first interval corresponds to failures at the Production Level; six modules have been built at this level: LHD, Street, Pit, Ore-pass, special equipment, and mini LHD. The second one corresponds to failures at the Reduction Level, the resources considered at this level are: rock-breaker equipment and rock-breaker operator. The third interval considers the failures occurring at the Transport Level involving mailbox, operator mailbox, and crossed (railways) resources. The fourth one keeps track of failures occurring at the Human Resource Level. If a failure token is activated and its numerical value falls within one of these intervals, the failure will be directed to only one of these modules. The connections of the internal KBSM design need to define the attributes which determine if the token taken this particular path has effectively the characteristics needed to enter a particular module and additionally allow activation of the system failure token. The defined attribute corresponds to the variable “z”.

4.2 Production Module

The Production Level modeling includes the categories of physical common and fixed resources such as LHD, street, pits, ore passes, and special equipment. This production level is internally connected with the design of the KBS Input Module by means of a point named Input Level Production. From there, 6 resource connections to the global system are established. Each branch has a transition which contains some specific condition describing a specific fault which is related to the resource defined for the particular branch activated by the token. The token which coincides with the generated condition will be able to continue its route through the branch of the tree diagram. Figure 11 shows the described Production Level and its corresponding failure histogram.

The conditions which were defined in each transition associated with a determined resource are associated to a controller defined for each resource. The connectors connect the transitions with the places called LHD Input, Street Input, Pit Input, equipment Input, and small LHD Input. These places define interconnection and input to the resource modules of the production level. Six modules have been built which are labeled according to their production level: LHD, Street, Pit, Ore passes and Equipment. These modules correspond to complex structures and are sub-levels of the Production Level and their modeling characterizes the hierarchical Petri Networks. These structural modules are connected to 6 places which correspond to the output interconnection points and their names are LHD Output, Street Output, Pit Output, Ore pass Output, and small LHD Output respectively which later on are connected to 6 branch output transitions. These transitions connect the branches of the Production Level with the interconnection point of the KBS Input Module, point or place that is coined as Output Level Production. Six connections start at the Event Input level and each one communicate with one of the modules assigned to the different resources. Six key transitions identifying the activated token are identified between the Event Input point and the place which is connected with the modular structure of the resource modules. These transitions filter the token which are trying to pass through due to the defined conditions. Only those tokens which possess the attributes which exactly correspond to those defined by conditions of a particular transition will be able to follow their route through the tree.

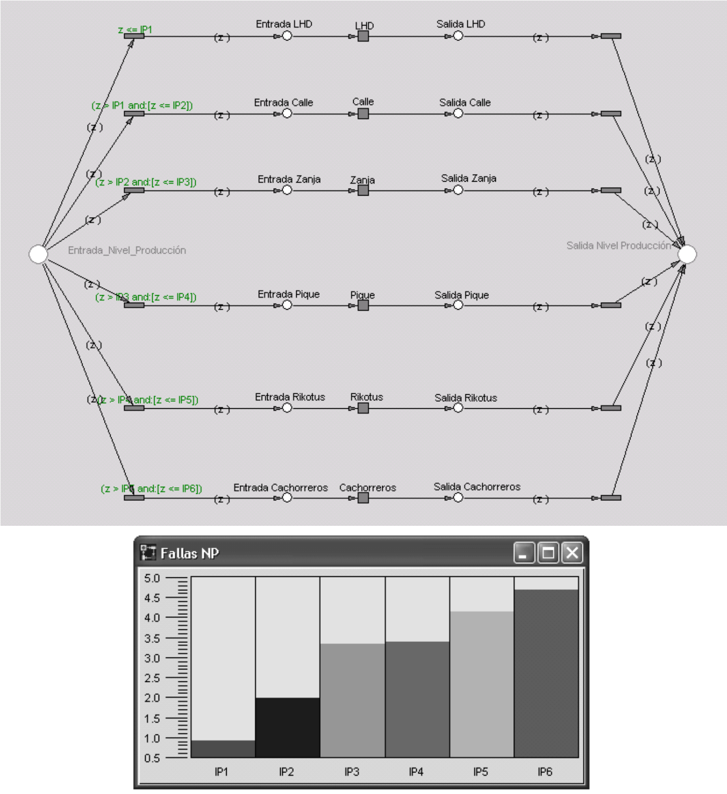


Figure 11: Production Module and corresponding failure histogram.

The LHD resource is a production level resource which is continuously exposed to faults. The potential events which have been defined for this resource are: repair, major failure, minor flaw and oil supply. Figure 12 shows LHD Module. The potential events which have been defined for this resource shown in Figure 12 are: repair, major failure, minor flaw and oil supply.

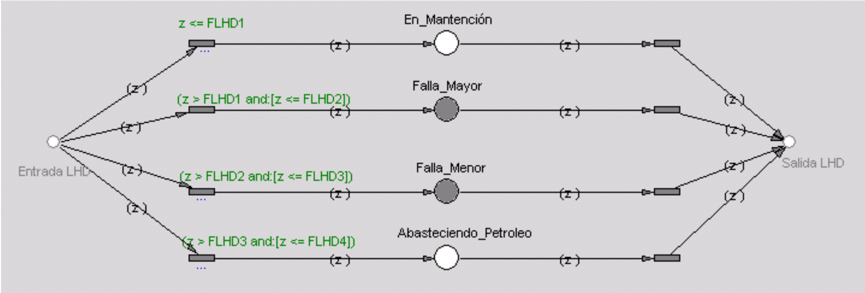


Figure 12: LHD Module.

If a token has arrived to the Street Module then it can be unequivocally identified with the event that has occurred and define the affected resource associated to one of the streets. The faulty states which have been defined for this resource are: isolation through powder, out of service, intervention by means of equipment, repair, no information and dirty, as shown in Figure 13. Ten transitions filtering tokens which try to cross the conditions defined for each transition assigning features to those who make it though with actions previously defined for each case for each transition, can be found between these places and the place called Input Ore pass.

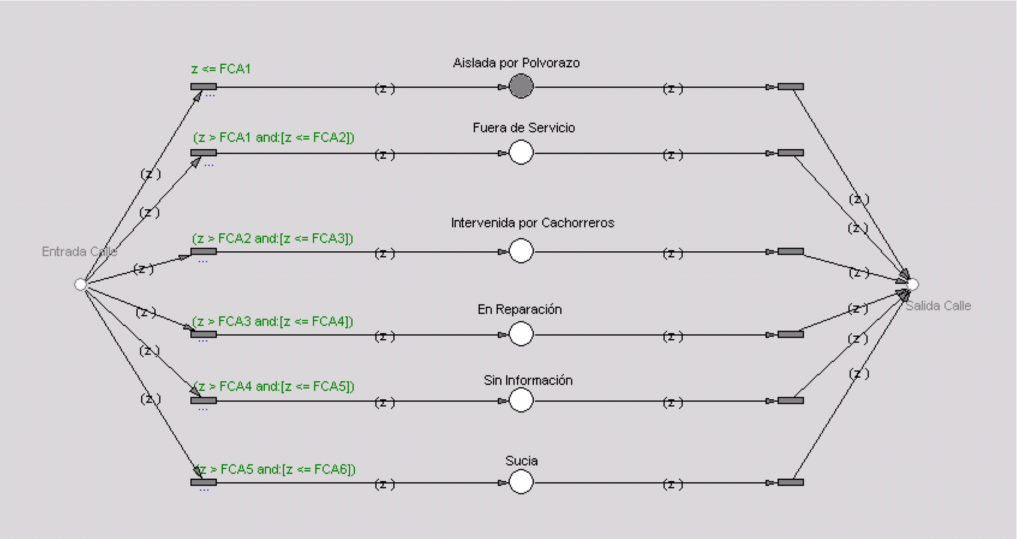

z <= FCA1
Aislada por Polvorazo
(z > FCA1 and:[z <= FCA2])
Fuera de Servicio
Intervenida por Cachorreros
(z > FCA2 and:[z <= FCA3])
Entrada Calle
En Reparación
(z > FCA3 and:[z <= FCA4])
Salida Calle
Sin Información
(z > FCA4 and:[z <= FCA5])
Sucia
(z > FCA5 and:[z <= FCA6])
(z)

Figure 13: Street Module.

The actions of each transition are the key elements within the system since they yield a particular solution and the effects as a function of time as well as the effects in the production levels at some breakdown. At a later stage 10 transitions corresponding to the output transitions are built. They communicate with the branches of the Ore-pass Module and the interconnection point of the Production Level Module which is labeled as Ore pass Exit. The variable “z” which coincides with the attributes of the tokens circulating through the module is assigned to those connectors joining places with transactions and transitions with places. Once the token which has passed through its coincident route leaves the module, it allows to communicate with the production level tree. The faulty conditions identified and modeled for the ore-pass resource are the following: UIT Barrier, With Mud, Closed, Hung, and Hung with Height, Cut off, Sunk, Limited, Operational Problems, and Repaired.

In the same way the identified faulty status modeled by means of the Pit resource is: Full, in Repair, No information, Without Information and Barred. Each one of these four transitions determines if a token can or cannot follow a particular route. A token will follow its determined route only if its features fulfil the determined conditions in each transition. The transitions have a condition which allows finding out if the features of a token are associated to the resource state which had been defined.

4.3 Reduction Level Module

The Reduction Level was modeled according to a similar logic defined at the Production Level. A flexible and hierarchical design of resources which are exposed to events and failures was utilized. This design includes a module which considers all possible faults which might occur within a given resource of the Reduction Level which affects the system operations. The potential faulty states defined for the design of the considered resources are: rock-breaker equipment and rock-breaker operator. The design of the reduction level starts with the place called Reduction Level whose goal is to connect the internal design of the KBS Module with the internal design of this level. Two connectors with attributes "z" appear at the place Reduction Level. It is important to mention the fact that if a token does not coincide with the attribute of a connector, then it will not be able to continue its route through this branch. Each one of these connectors is linked to a new transition whose finality is to filter all token which intend to pass through this tree. The tokens which are activated during the initial transition of the KBS Input Module have specific characteristics associated to a given fault which in turn affects a specific resource of the system. Therefore if a token makes its way through some branch of the Reduction Module, it must have some common definition with some transition of the module. Figure 14 shows the Reduction Level Module and its corresponding failure histogram.

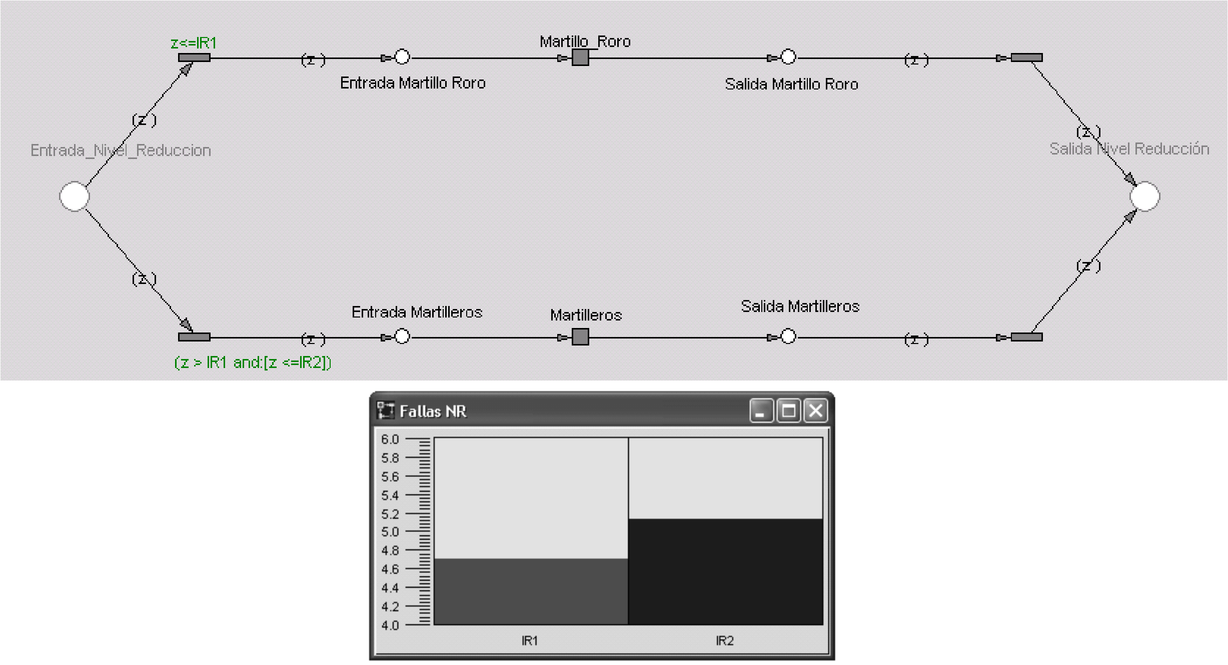


Figure 14: Reduction Level Module and corresponding failure histogram.

The transitions which determine the characteristics of each branch of the Reduction Module possess two characteristics for the modeling: conditions and actions. The conditions are meant to identify those tokens which have characteristics that coincide with the definition of a given transition. Each condition was built according to the frequency of fault occurrence at the corresponding level. If a token which has been activated has attributes that coincide with the condition defined for a transition, it will be allowed to continue its journey through the branch. The conditions are the means of a filter which determines which fault is or is not associated to some resource. If a token coincides exactly with the condition defined in the transition, it will be able to follow its route and in addition it will take up the characteristics that have been assigned to it by the action of the transition. The action hands over to the token new attributes, e.g. assigns fault time to the system and in addition will yields information concerning the best solution which has to be applied to continue the operation. The actions assign the optimal solution defined by experts. The places which allow connecting the modules with the reduction level are called Rock-breaker and Rock-breaker operator Input. The exit of each one of these modules is connected to the exit transitions of the module. Each one of these transitions is connected to the place called Exit Reduction Level. The faulty states of the defined and modeled resources are: energy loss, Major fault, Minor fault, and Quality Control. Figure 15 shows the Rock-breaker Equipment Module.

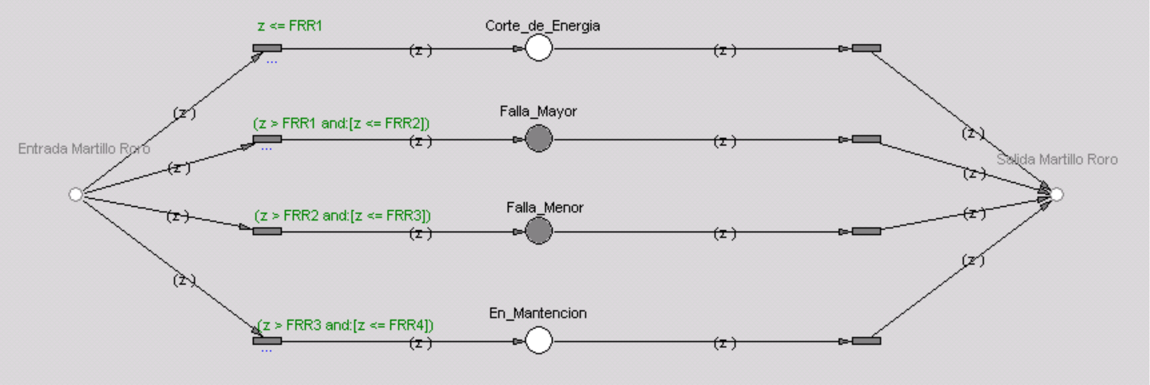


Figure 15: Rock-breaker Equipment Module.

4.4 Transport Level Module

This module represents the resources associated with the Transport Level. The connectors joining the modules and places of this level have a common attribute labeled "z" which allows tokens whose attributes are as well defined as "z" to take the routes through the different branches of the module. The tokens which get to this module are appearing by the initial transition generating fault tokens of the KBS module. These tokens pass through the place Fault Input and step into the internal KBS module moving around within the inner sections of this module until they arrive to a place called Input Events. At this point, the entrance transition verifies if the characteristics of the arriving token matches with the condition which had been defined. If this is the case the token will be allowed to pass through to get to the Transport Level Module as shown in Figure 16. These mentioned conditions correspond to the characteristics that filter and identify the faults that have been defined and which can be generated for the transport module. The resources which participate in the transport level are: Mailbox, Mailbox Operators and Crossed. The faulty states considered for the design of the Mailbox resource are: Ore-pass pumper, Embanked Mailbox, Major Fault, Minor Fault and Tanked Mailbox. The identified faulty states which were modeled within the Crossed module are: Major problem, Minor fault, and Crossed Disraeli and Mill-hopper problems. This module as shown in Figure 16 and 17 was built using Colored Petri Networks which allow identification of those critical failures in this level affecting the fixed physical resource Crossed. The failures that can be considered as critical for this resource are Major Failure, Minor fault, and derail. The Module Mailbox Operators is the last Common Physical Resource which is modeled in the transport level. Its design originates from the construction of the place called Input Mailbox Operators.

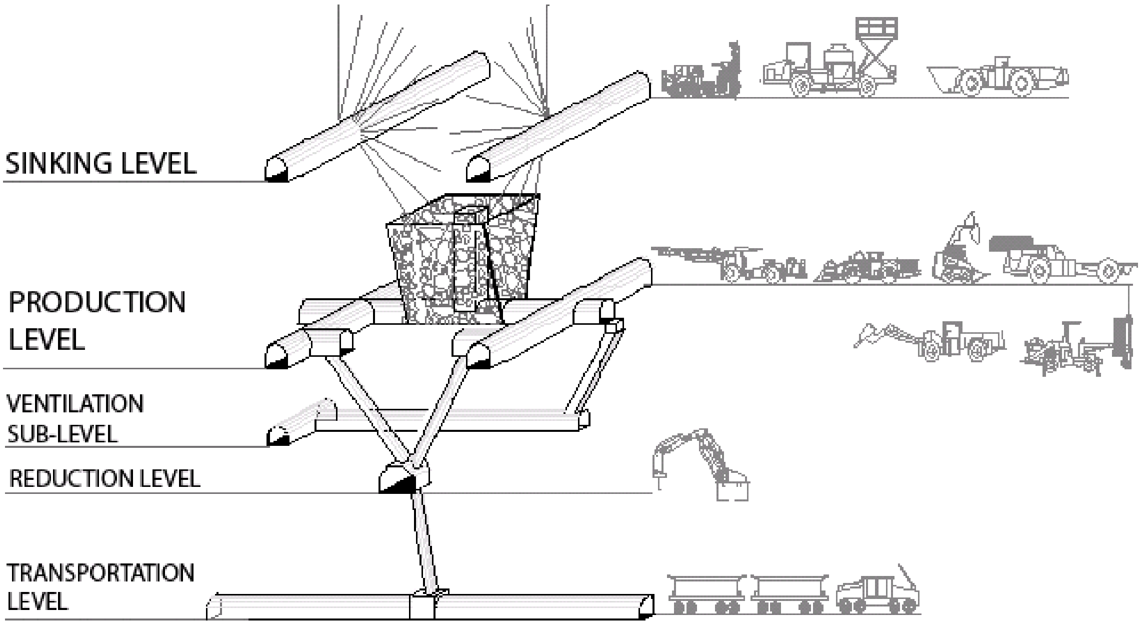


Figure 16: Transport Level Module and corresponding failure histogram

4.5 Human Resource Level Module

This module is one of the most important in the modeling process since its events are those who generate the main delays or interruptions during operations in the mine. In fact, the personnel of the mine are rotating between the different levels according to the requirements and the state of each level. This was the reason to model the human resource as another module of equal importance than the production, reduction and transport levels. On the other hand as human resources becomes one single unique module one can integrate the availability of all personnel that is participating in operational activities and optimally assign these resources for the operation.

The modeling of this module starts with the place labeled HH.RR. Input which connects these resources with the KBS module which has three levels of operation. Twelve connectors appear at this point, each one generating a given branch all associated with some failure affecting human resources. Each connector must connect the entry point with the entry transitions of the module. These transitions have determined conditions that identify if the tokens which intend to pass through the transition fulfil or not the attributes defined by the condition. Only those token will be allowed to pass when they coincide with the conditions defined by each branch. The transitions include actions that allow the assignment of new features to the tokens which fulfil the condition of the transition. The corresponding histogram is evaluated for assisting the modus operandi of the

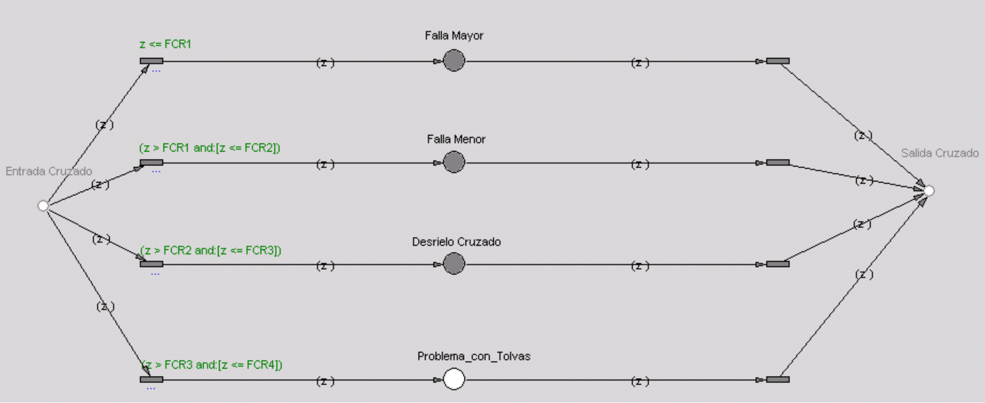


Figure 17: Crossed Module.

KBS model yielding the information required by the conditions of the transitions with the intention to identify the tokens which are defined by a numerical value associated to an event at the HH RR level.

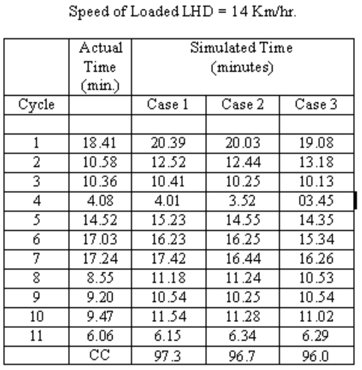
The events considered are the following: Move LHD, Crossed Disraeli, Mill-hopper problems, Ventilators detention, Electrical Energy cut, different kinds of Human Resources time lacks, Labor Accidents, isolation areas, and Dispatch Fall. Besides the fact that he complete module HH.RR. was designed as a modular structure, the events which are generated directly affect these resources and hence one cannot refine the modular features of its construction. However one has performed differentiation with respect to the critical character of certain events after identifying those which paralyzed the entire working shift. In this case the events were designed using Colored Petri networks. Following the places whose names correspond to defined events for this level, twelve transitions were built. They correspond to exit transitions of the module and a connector appears at each one connecting the exit place of the module to the event branch. This place is labeled as HH.RR. Output and is in charge of the connection of the HH.RR. module with the rest of the KBS system.

5 Results

The Intelligent Supervising System (SM-KBS) was implemented using Pace TM Petri Nets tool. In order to validate the Simulation Model data from the real production plan for a vehicle was analyzed. The time to perform 11 cycles was recorded, where one cycle corresponds to the number of trips (from one ore-pass to one pit and from one pit to one ore-pass) required for completing the amount of tons. of material to be extracted from the extraction point as specified in the production plan. The real time to perform each cycle is shown in the second column of the table in Figure 18. The three following columns show the simulated time for different cases of speed of the vehicle considered in the simulation model.

Case 1: Speed of Unloaded LHD = 10 Km/hr; Speed of Loaded LHD = 12 Km/hr.
Case 2: Speed of Unloaded LHD = 12 Km/hr; Speed of Loaded LHD = 12 Km/hr.
Case 3: Speed of Unloaded LHD = 12 Km/hr; Speed of Loaded LHD = 14 Km/hr.

The last row shows the correlation coefficient (in %) between the data series of the



Speed of Loaded LHD = 14 Km/hr.

	Actual Time (min.)	Simulated Time (minutes)		
Cycle		Case 1	Case 2	Case 3
1	18.41	20.39	20.03	19.08
2	10.58	12.52	12.44	13.18
3	10.36	10.41	10.25	10.13
4	4.08	4.01	3.52	03.45
5	14.52	15.23	14.55	14.35
6	17.03	16.23	16.25	15.34
7	17.24	17.42	16.44	16.26
8	8.55	11.18	11.24	10.53
9	9.20	10.54	10.25	10.54
10	9.47	11.54	11.28	11.02
11	6.06	6.15	6.34	6.29
	CC	97.3	96.7	96.0

Figure 18: Real time to perform each LHD cycle

column and the column of the real production. As it can be seen the model generates similar results in relation to the real situation. However, the best results are obtained in the second case.

A similar analysis was carrying out by taking a production plan and computing the simulated time to complete the production plan for the whole street. The results show that the simulated time for the same 3 cases is:
Real time: 127,32 min.
Case 1: 138,36 min.
Case 2: 134,96 min.
Case 3: 132,47 min.

As it can be seen, the maximum difference with the real system is only 8% , corresponding to the case 1. In the third case the difference is 3,9% . According to the results, it can be concluded that the model is a good representation of the actual mine.

Also the impact of different types of undesired events is studied. Two types of criteria are used:

Production Criterion: The objective of this criterion is to find the minimum number of occurrences of a given type of event having the consequence that the level of production assigned to a work shift start to diminish.

Time Criterion: The objective of this criterion is to establish the minimum number of occurrences of a given type of event having the consequence that the work shift finishes without the production plan having being completed.

For carrying out this analysis, two actual production plans were used (Plan 1 and Plan 2). The sensibility analysis was undertaken using an iterative process, isolating in each case the probability of occurrence of the event studied, as a way to obtaining its effect on the production level. To carry out the iterative process an initial probability of occurrence was used (based on historical records) and it was increased in intervals of 1 to 2% , depending on the type of event. The number of runs was selected so that the level of confidence of the results is 95% . As an illustration the analysis of the "Major Failure of a LHD vehicle" in the case of two streets (S1 and S2) is presented in Figure 19. As can be seen, the frequency of failures o LHD vehicles has a significant effect on the Production Level.

Different scenarios have been proposed in an 8 hours shift in order to evaluate the

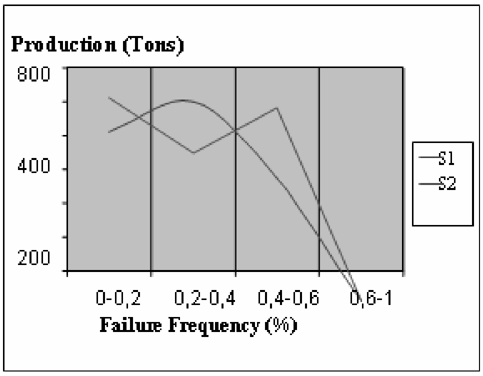


Figure 19: Analysis of Major Failure LHD

hole operation of the mine. In initial scenario for testing, in 6 streets modeled without stochastic events, the programmed production and the real one agree. Next, several iterations were done generating between 8 and 13 random failures which affect the different levels and resources of the system, altering the programmed initial operation. Real time versus the programmed time and tonnage is shown in Figure 20.

The failures detected in this scenario were: a) at the Production Level, 5 dirty street failures affecting streets 1, 4 and 5 with a failure probability of 18 %; b) at the Reduction Level, 6 failures affecting rock-breakers, for example: chamber with disconnection of chains affecting street 6 with a failure probability of 1.2 %; c) at the Transport Level, one failure was detected in the Mail Box with pit pump; d) at the Human Resource Level, one failure related to delay in the "Operator assignment" with a delay of 60 minutes of no availability of the resource and a failure probability of 18 %.

The first validity step was done at the moment of checking that the solutions given by the system for each failure was correct. When checking this information, it was confirmed that the system gave the scheduled recommendations as outcome variable; therefore, in this first point the right working of the system was established. In the case of dirty street the system recommends calling to LHD equipment to work in cleaning the street with its bucket or using a small LHD vehicle which make the cleaning of the place. In the case of closed ore-pass the system recommends to stop the extraction in ore-pass because of low copper law or by fulfilling the programs and to reprogram. In the case of rock breakers with disconnection of chains the system suggest that the chains must be connected after waiting that the pit is full and put signals in the Transport level. In the case a minor failure happened in the rock breakers, the system recommends that mechanical must be called to repair in situ, where they can take the bad spare out to a special place for its reparation or be repaired in situ. It must be continued loading if there is any space. In the case of the system recommends that the train lines must be cleaned, iron wagons, box and repair any damage in the facilities caused by the fallen material fallen violently from the pit, then it is necessary to reprogram the activities.

It was observed that the increase in the number of failures leads to the decrease in the efficiency and at the same time to the decrease in the real tonnage of the streets. In spite of that the box resource is not something that affect directly to the programming production of a street, the global effect of this stop affects in a cross way the use of the resources operation that could be used.

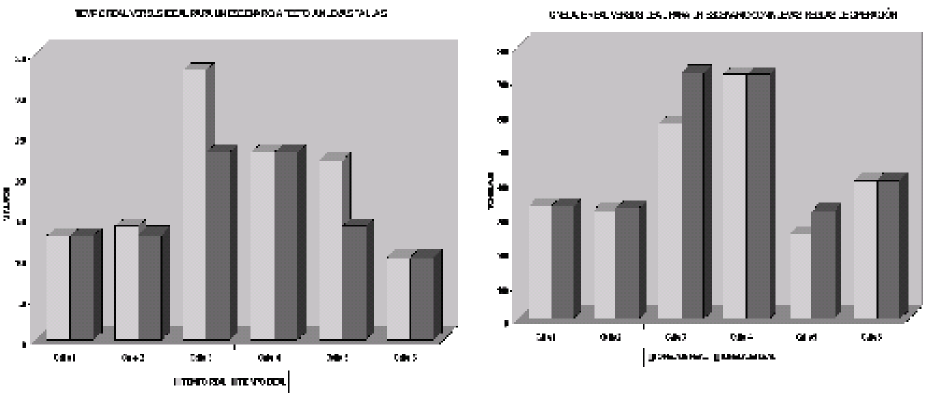


Figure 20: Real time versus the programmed time and tonnage.

In this simulated scenario composed by 13 random failures, only 11 of them were recognized by the system. When an undetermined failure is found, identifying the level or levels affected, its frequency, the resources involved, and the estimated time for its solution, as a result of this, a new operation rule is added to the KBS.

6 Summary and Conclusions

The Intelligent Supervising System designed, composed by a Simulator linked to Knowledge Based System allows the study of the real operation of an underground mine, including the generation of events affecting it. The design covers the three operational levels of the mine, including human resource events. The use of hierarchical, colored and temporized Petri Nets in the design gave more flexibility to the model and allowed a hierarchy of the levels with their resources. In order to identify the failures or critical events that take place in the operation of the mine, the colored Petri Nets were used, tool that helped to give the aspect of critical that they have in the real operation.

The programming considers 56 types of failures and it is possible to modify the probability of occurrence of one of them. Regarding the results obtained, the model was utilized to investigate the impact of the different types of failures. It was found that, in order of importance, that the events that affect the production most are the blocking of a secondary ore-pass, a minor breakdown of a rock breaker and a major breakdown of a LHD vehicle.

The effect of human resource failure is global and it affects directly the amount of production for the shift. When failures occurred, the KBS System gave the correct solution that was programmed. The delay that was originated for each failure was scheduled according to the information given by experts in terms of the time consumed in each case. The incorporation of critical failures in the modeling process is another innovative characteristic which represents in a realistic way the operation of the mine. For outcomes which do not generate a recommendation or solution defined in the KBS, a new outcome as a case is generated, adding new knowledge or a solution which generates new operational rules. These results are useful to the managers, because they know now in which resources to pay attention and improve the maintenance of those resources.

References

[1] L. R. Atero, F. M. Cordova, L. E. Quezada, V. Olivares, J. Sepulveda, "Conceptual Model of Virtual Supervising Operation System VOSC," *Proc. of the 17 International Conference on Production Research, Virginia, USA*, 2003.

[2] F. Cordova, L. E. Quezada, R. Salinas,"R3: A mineral reduction system for underground mining,"*Proc. of the 5th International Symposium on Mine and Mechanization and Automation, Sudbury, Canada*, 1999.

[3] M. Ghaeli, A. B. Bahri, P. Lee and T. Gu, "Petri-net based formulation and algorithm for short-term scheduling of batch plants," *Computers and Chemical Engineering*, Vol. 29, Issue 2, pp. 99-102, 2005.

[4] T. Gu, A. B. Bahri, "A survey of Petri Nets applications in batch manufacturing," *Computers in Industry*, Vol. 47, Issue 1 pp. 99-102, 2002.

[5] G. Gutuelac, "Descriptive Timed Membrane Petri Nets for Modelling of Parallel Computing,"*International Journal of Computers, Communications and Control*, Vol. I, N 3, 2006.

[6] M. Medjoudj, P. Yim, "Extraction of Critical Scenarios in a Railway Level Crossing Control System,"*International Journal of Computers, Communications and Control*, Vol. II, N 3, 2007.

[7] T. Murata, "Petri Nets; Properties, Analysis and Applications," *Proceedings of the IEEE*, Vol, 77, Num. 4, pp. 541-580.

[8] F. Ounnar, P. Ladet, "Consideration of machine breakdown in the control of flexible production systems," *International Journal of Computer Integrated Manufacturing*, Vol. 17, Issue 1, pp. 62-82, 2004.

[9] J. M. Proth, X. Xie, *Petri Nets: A Tool for Design and Management of Manufacturing System,* John Wiley & Son Inc, 1997.

[10] J. Rosell, "Assembly and Task Planning using Petri Nets: A Survey," *Proceedings of the Institution of Mechanical Engineers-Part B- Engineering Manufacturing*, Vol. 18, Issue 10, pp. 987-994, 2004.

[11] USACH-FONDEF Group, "Virtual Supervising Control of Underground Mining Operations,"*Proposal FONDEF Project D01I1091*, CONICYT, Chile, 2001.

[12] W. Zhang, Th. Freiheilt, H. Yang, "Dynamic Scheduling in Flexible Assembly System based on timed Petri Nets Model," *Robotics and Computer Integrated Manufacturing*, Vol. 21, Issue 6, pp. 550-558, 2005.

Felisa M. Cordova, Luis E. Quezada,
University of Santiago of Chile, USACH
Department of Industrial Engineering
Ecuador 3769, Santiago, Chile
E-mail: felisa.cordova@usach.cl

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 84-99

Neuro-Fuzzy Modeling of Event Prediction using Time Intervals as Independent Variables

Simona Dziţac, Ioan Felea, Ioan Dziţac and Tiberiu Vesselenyi

Abstract: Estimation of possibility of future events is one of the main issues of predictive maintenance in technical systems. The paper presents an application of neuro-fuzzy modeling in reliability analysis. For this application we first consider some aspects of neuro-fuzzy modeling using the MATLAB programming environment and the mathematical model of event prediction. Than we succeed with prediction of time intervals at which events can occur using the registered data from a group of power energy distribution center maintenance databases. Conclusions are drawn regarding method applicability and future improvements.

Keywords: neuro-fuzzy modeling, prediction, membership function.

1 General considerations

Prediction of parameter values related to halts, events or failures of power distribution nets is one of the major simulation methods in the field of reliability. In this paper, the authors had searched o method which can generate a prediction model based on measurements of parameter values of the system. Such a model can be developed using neuro-fuzzy simulation methods [2, 3, 4, 5, 7].

Fuzzy reasoning has the ability to deal with uncertain information, while neural nets can be developed based on input-output data sets. Neuro-fuzzy algorithms combines the benefits of both methods.

The goal of this article is to present how neuro-fuzzy models can be used in the prediction of some events in the area of power distribution systems based on time intervals between two events. The time interval analysis can be then used to study how often events will occur in the future. The case study had been developed using the “events” tables from “Electric Energy Quality Indicators” database [1, 6, 8]. Prediction of time intervals at which events can occur had been made for the Center of Electric Energy Distribution of the City of Oradea, using the MATLAB environment as software development tool.

2 Neuro-fuzzy modeling

The base idea of neuro-fuzzy modeling is that the fuzzy system parameters (inference rule set, fuzzy membership functions) are established by learning (training) known input-output data sets. The learning methods are similar with those used in the case of neural nets.

The way in which this method is used is similar with the way in which the majority of identification techniques are used:

- in the first step a hypothesis is generated on the way that the system should operate and an initial model is defined (by associating membership functions and a set of rules).

- in the second step, input-output data sets are selected for training.
- in the third step using the selected data the adjustment of fuzzy membership functions is made using the neuro-fuzzy and accounting for corresponding minimal error criterion.
- in the final step, a testing data set is chosen and the developed system is tested with this set.

If the obtained results for the testing data set is satisfying then a series of methods exist in order to optimize the model:

a. increasing the amount of data in he training set;

b. increasing the training epochs number;

c. decreasing the increment of membership function adjustment algorithm.

In general, this type of model is working well if the training data can reflect representative characteristics of the modeled system. In practice the measured data set are not always reflecting the whole number of situations, which can occur and a certain value of uncertainty must be accepted.

System testing is made using a set of data different from the training data and the mean squared error obtained during the training epochs and the testing algorithm. If this error converges than the training is supposed to be correct.

Computing of adjustment parameters of membership functions is made with the help of a gradient vector, which optimizes the correlation between the neuro-fuzzy model and the real system specified by the input-output data. Once the gradient vector found, an optimization algorithm defined by minimizing the mean square error of real and modeled data is applied. The membership function parameter adjustment then is made using a hybrid algorithm based on the least square method and error back-propagation [2, 4, 5, 7].

In order to develop neuro-fuzzy models in MATLAB environment, the ANFIS module (Adaptive Neuro-Fuzzy Inference System) is used. ANFIS generates a self-adjustable fuzzy inference system by training with input-output data sets. The ANFIS module is called from MATLAB programs by the “anfis” command, which bears the following set of parameters [3]:

- number of training epochs;
- training error limit;
- initial value of optimization step;
- optimization step decrease rate;
- optimization step increase rate;

The training process stops when the number of prescribed epochs or the training error limit is reached.

The “anfis” command returns an informational structure, which is the neuro-fuzzy model of the process as it was given by the training data set.

3 Event prediction using neuro-fuzzy models

In the case that the goal is the prediction of future values based on a set of measured values reflecting a certain anterior evolution of a system, the use of a predictive model based on input-output data sets is somehow unusual, because only one set of data is at disposal.

That is the case of the values contained in time series, covering a domain of $[0, t]$, which values are to be used to predict the systems behavior at time $t + P$.

A convenient method for such prediction is to consider a series of N values from the known interval with the step $\triangle$:

$$(x(t-(N-1)\ \triangle), \ldots, x(t-\ \triangle), x(t)) \tag{1}$$

If, for example $\triangle = P$, is chosen, for each value of the known series, at time t the $w(t)$ vector can be defined as:

$$w(t) = [x(t-\ \triangle)x(t-\ \triangle)x(t-\ \triangle)x(t)] \tag{2}$$

The $w(t)$ vector can be used as a set of training input data for the neuro-fuzzy model. The output data set will be predicted (or estimated) as $s(t)$:

$$s(t) = x(t+\ \triangle) \tag{3}$$

As usual, the data set is divided in two intervals: the first interval is the training data and the second interval will be used as testing data.

In the training data, for each value the $w(t)$ value will be computed and the $w(t)$ vector will be considered as the input set and the second interval will be considered as the testing data set.

4 Application regarding time interval prediction at which events can occur in an electric power supply system

In the "Events" table of "Electric Energy Quality Indicators" database [1, 6, 8] there are recorded the dates of events at every consumer.

In the neuro-fuzzy predictive method, presented in the preceding paragraph, the algorithm had generated 16 rules with 104 parameters of the membership functions (considering "Gaussian" type membership functions). In order to reach satisfactory results is important that the number of training data be twice larger than the number of parameters of the membership functions. The case of the database used, a minimal set of 200 training data could be used, only in the case of "Electrica Nord Transilvania Oradea" company for a time interval of 6 years. Considering a predictive analysis on subdivisions (centers) of three distinct studies had been made for: Oradea, Stei and Alesd cities. From the reliability point of view it had been considered efficient to analyze the time between two events. So future consecutive event occurrence intervals can be estimated (or how frequently such events will occur in the future).

In order to fulfill the prediction task under the MATLAB environment, database information (initially worked out in EXCEL), had been decrypted with a special module [1]. Then the main interest fields were sorted (primary index is the date and time of the event and secondary index is the center name [1]).

From the sorted data, the number of days between two incidents had been counted and the neuro-fuzzy prediction was made for every center with the help of the software realized by the authors [1].

The parameters of the ANFIS command were as follows:

- number of training epochs = 150;
- training error limit = 0;
- initial value of the optimization step = 0,0001;
- the decrease rate of the optimization step = 1,9;
- the increase rate of the optimization step = 2,1;

For Alesd center the training data set and the testing data set was 200 values each and for Stei and Oradea centers was 300 values each.

Results of program running are presented in diagrams of figures 1-18. If the event occurrence diagram is required then it can be extracted from the predicted event period data using a specially designed program module, which can rebuild the time scale and represent the eventless periods as zeros and the event as values of ones. Using this function the diagrams in figures 19-21 were obtained.

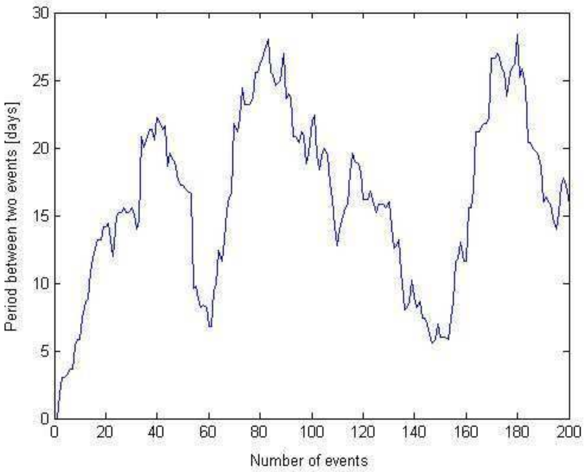


Figure 1: Training data for center Aleşd.

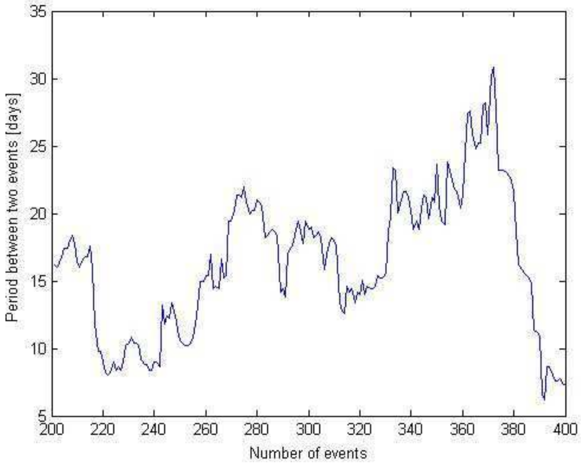


Figure 2: Testing data for center Aleşd.

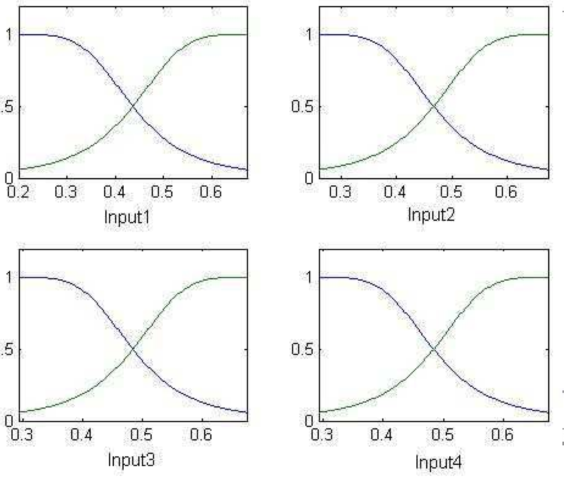


Figure 3: Initial membership functions for centre Aleşd.

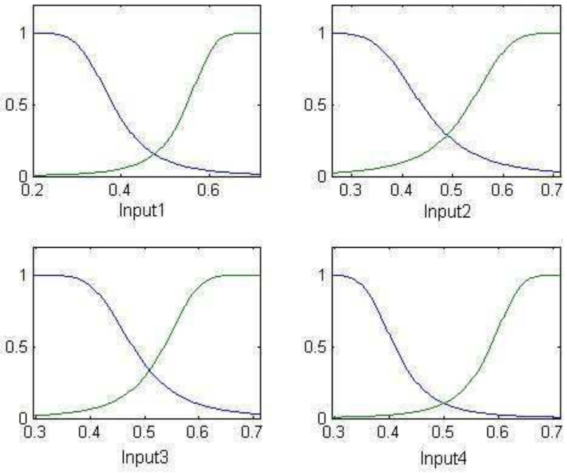


Figure 4: Adjusted membership functions for center Aleşd.

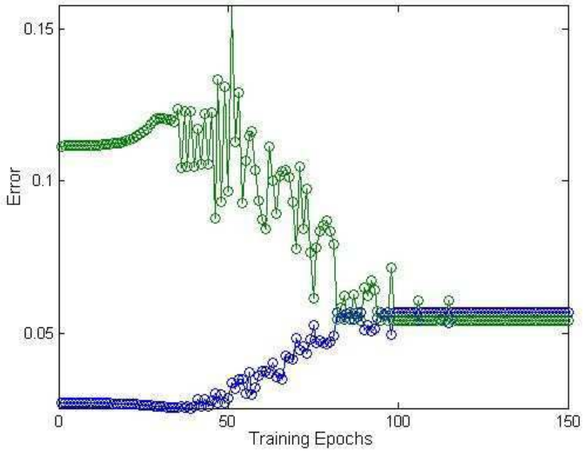


Figure 5: Comparison of training (blue) and testing (green) errors for center Aleşd.

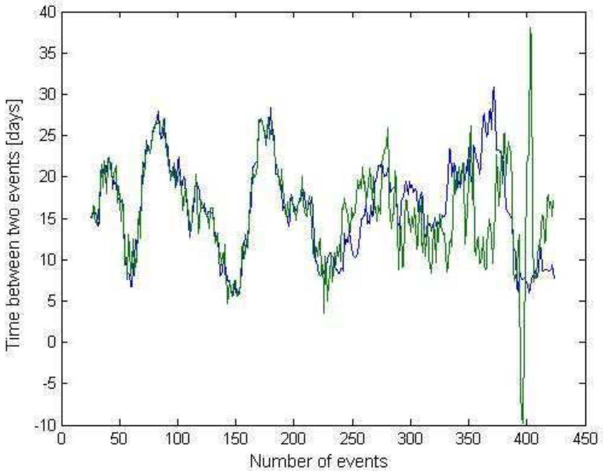


Figure 6: Event period prediction diagram real values (blue); simulated values (green) center Aleşd.

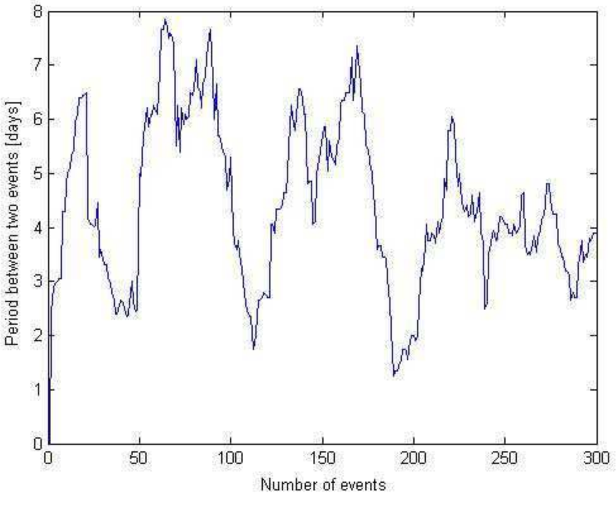


Figure 7: Training data for center Stei.

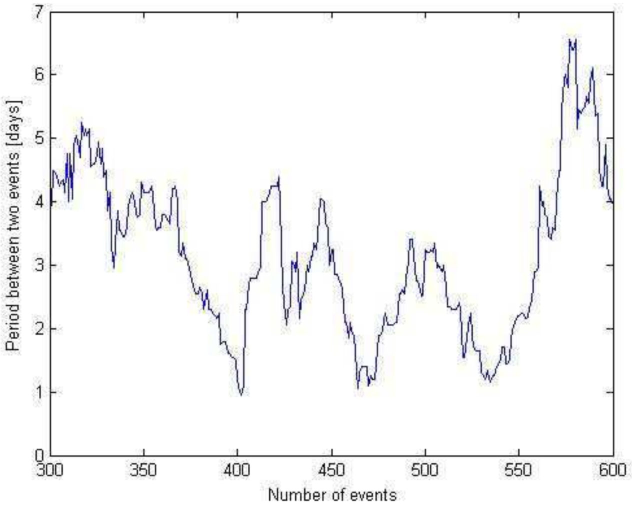


Figure 8: Testing data for center Stei

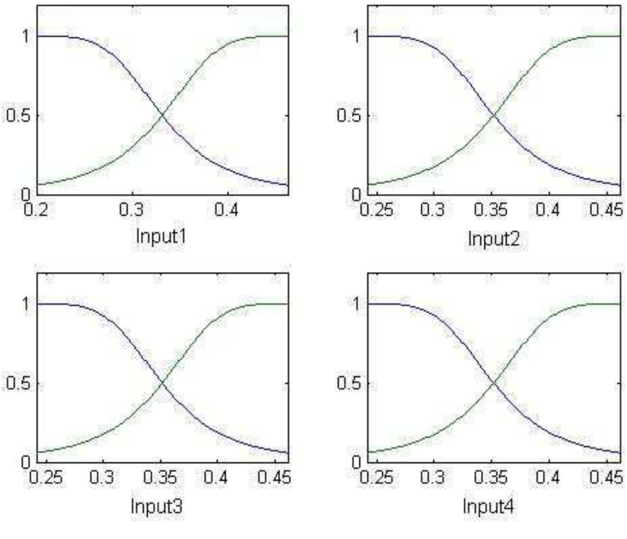


Figure 9: Initial membership functions for center Stei.

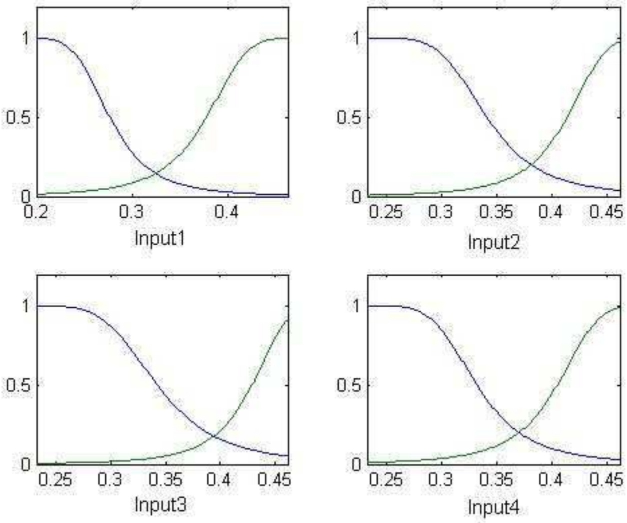


Figure 10: Adjusted membership functions for center Stei.

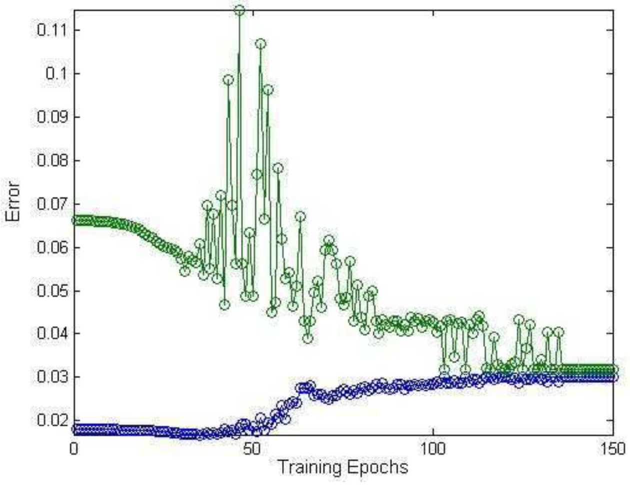


Figure 11: Comparison of training (blue) and testing (green) errors for center Stei.

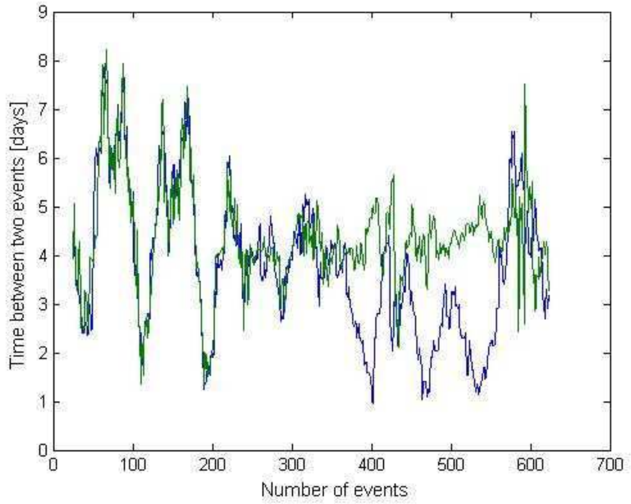


Figure 12: Event period prediction diagram real values (blue); simulated values (green) center Stei.

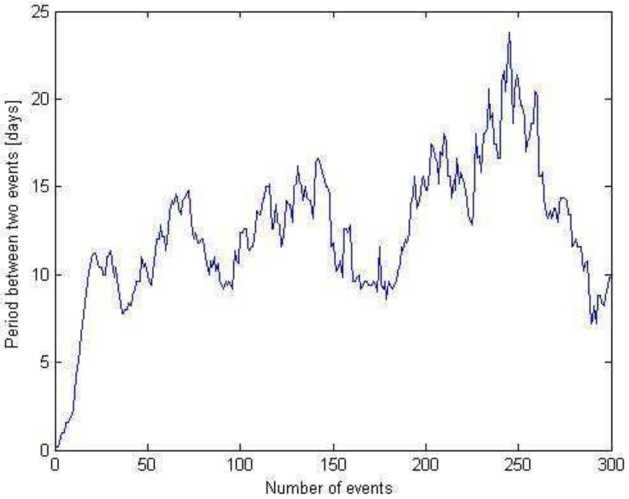


Figure 13: Training data for centre Oradea

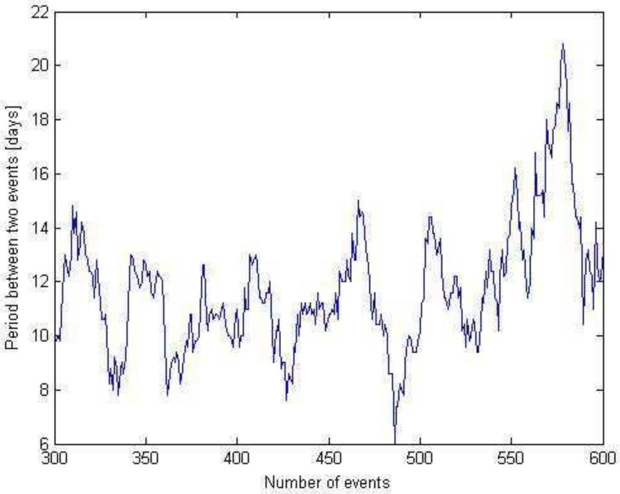


Figure 14: Testing data for centre Oradea

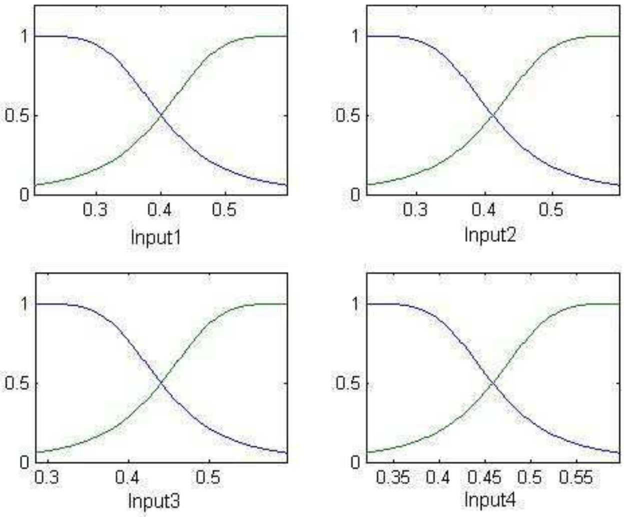


Figure 15: Initial membership functions for centre Oradea

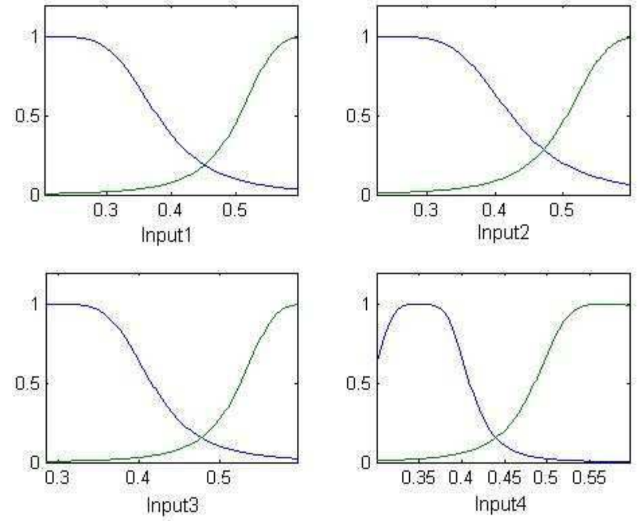


Figure 16: Adjusted membership functions for centre Oradea

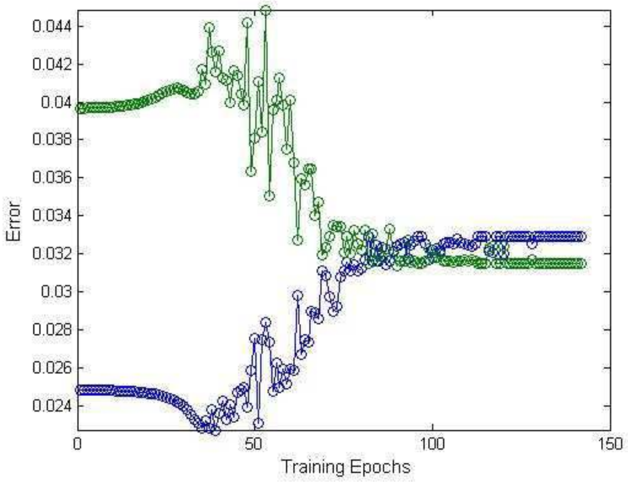


Figure 17: Comparison of training (blue) and testing (green) errors for centre Oradea

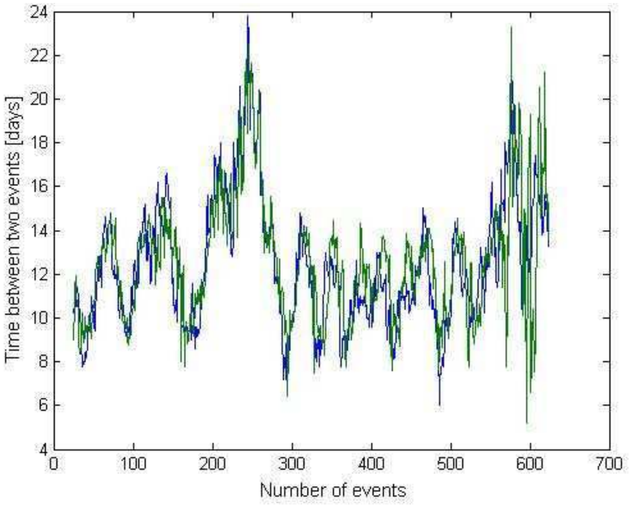


Figure 18: Event period prediction diagram real values (blue); simulated values (green) center Oradea.

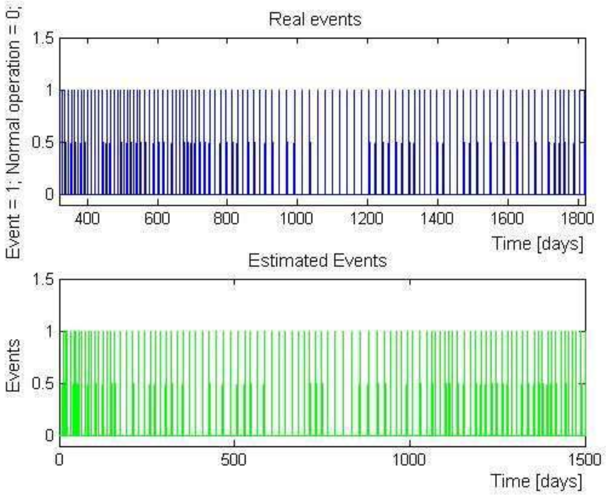


Figure 19: Event occurrence diagram for centre Aleşd

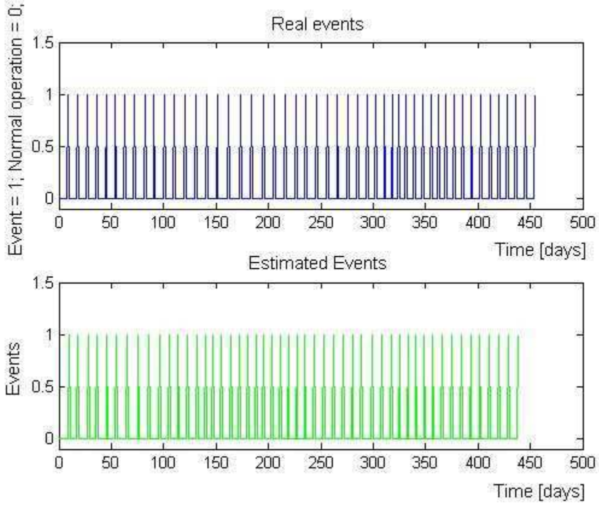


Figure 20: Event occurrence diagram for centre Ştei

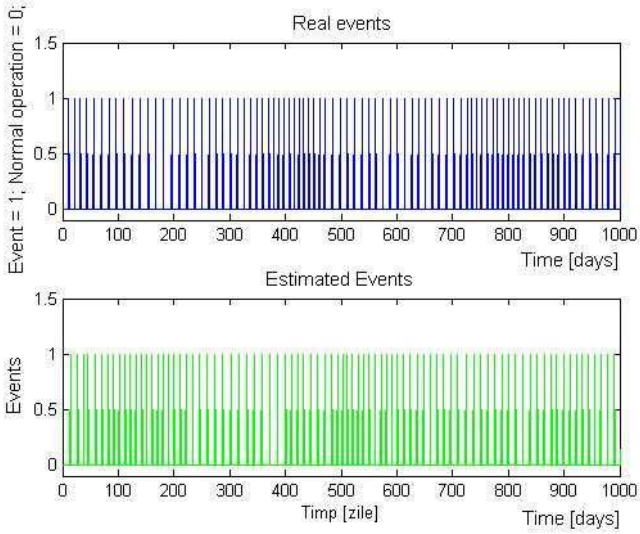


Figure 21: Event occurrence diagram for centre Oradea

5 Conclusions

Analyzing the error comparison diagrams (figures 5, 11 and 17) it can be observed that in all cases we have a convergence of values after 150 epochs. In diagram in figure 6 (centre Alesd), the predicted values are in the range of 200-400 events. The predicted values, which can be considered as satisfactory are in the range of 200-350 events.

In diagrams at figures 12 (centre Stei) and 18 (centre Oradea), the predicted values are in the interval of 300-600 events. for centre Stei, prediction values are considered satisfactory in the interval of 300-350 events and for centre Oradea, the satisfactory prediction is considered to be in the range of 300-500 events. In this range the prediction error is about 1-2 days.

The best prediction results are for centre Oradea and the worst are for centre Stei.

The event period and event occurrence diagrams presented can be considered as new tools for reliability studies in the field of electric energy systems.

The researches lead the authors to the conclusion that a good result can be obtained only if a large amount of data is available. In order to have good predictions the energy delivering centers must rethink their data recording systems to be more accurate and meaningful. The data should be also selectable in function of the event's nature. Consideration of event nature could not be made in this study because of missing information. Future studies should also consider additional information as machine and device condition and equipment age, which would improve prediction accuracy. For this approach a multi-criteria fuzzy inference system can be used.

There had been acknowledged that neuro-fuzzy prediction is a useful tool for this kind of systems reliability analysis.

References

[1] S. Dziţac, *Contribution to Modeling And Simulation of Electric Power Distribution Systems Reliability and Availability Performances*, PhD Thesis, University of Oradea, 2008.

[2] J.-S. R. Jang, "*Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm,*" Proc. of the Ninth National Conf. on Artificial Intelligence (AAAI-91), pp. 762-767, July 1991.

[3] J.-S. R. Jang, "*ANFIS: Adaptive-Network-based Fuzzy Inference Systems,*" IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665-685, May 1993.

[4] J.-S. R. Jang, and C.-T. Sun, *Neuro-fuzzy modeling and control*, Proceedings of the IEEE, March 1995.

[5] J.-S. R. Jang, and C.-T. Sun, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, 1997.

[6] C. Secui, I. Felea, S. Dziţac, E. Dale, I. Boja, *Database and software system for evaluation of quality indexes of electric energy supplying service*, Scientific Bulletin of the "PO-

LITEHNICA", Timioara, Seria Energetica, Proceedings of the 7 th International Power Systems Conference, november 22-23, PSC 2007, pag. 589-596, ISSN 1582-7194.

[7] L. A. Zadeh,, "Fuzzy Logic," Computer, Vol. 1, No. 4, pp. 83-93, 1988.

[8] *Elaborarea unui pachet software si constituirea bazei de date pentru evaluarea indicatorilor de calitate a serviciului de furnizare a EE la nivelul consumatorilor industriali si edilitari din judetul Bihor*, Raport la contractul de cercetare nr. 3050/2006, Universitatea din Oradea, Facultatea de Energetica, 2007.

Simona Dziţac
University of Oradea
Universitatii St. 1, 410087, Oradea, Romania
E-mail: sdzitac@rdslink.ro

Ioan Felea
University of Oradea
Universitatii St. 1, 410087, Oradea, Romania
E-mail: ifelea@uoradea.ro

Ioan Dziţac
Department of Economic Informatics
Agora University of Oradea
Piata Tineretului 8, Oradea 410526, Romania
E-mail: idzitac@univagora.ro

Tiberiu Vesselenyi
University of Oradea
Universitatii St. 1, 410087, Oradea, Romania
E-mail: tvesselenyi@yahoo.co.uk

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 100-128

Towards to Colony of Robots

Gastón Lefranc

1 Introduction

A Nomad robot was used to do all mission in Mars, being a nice application of a mobile robot, obtaining new knowledge in the space. However, the inversion is very high and the design is complex. It requires many capabilities to operate in several different environments. The nature of these work environments requires the robotic systems be fully autonomously in achieving human supplied goals. One approach to designing these autonomous systems is to develop a single robot that can accomplish particular goals in a given environment. The complexity of many environments or works may require a mixture of robotic capabilities that is too extensive to design into a single robot. The main problem is when the robot fails, in this case all the work will stop. An alternative way is to use a groups of mobile robots working together to accomplish an aim that no simple robot can do alone.

Groups of heterogeneous robots working together, like a society of insect, can accomplish the same mission that one robot. Using simpler mobile robots doing specific task, is less expensive, more reliable and it can reach the same aims of one robots. Others applications of these groups are potentially good in manufacturing, medicine, space exploration and home.

Additionally, time constraints may require the use of multiple robots working simultaneously on different aspects of the mission in order to successfully accomplish the objective. In cases, it may be easier and cheaper to design cooperative and collaborative teams of robots to perform the same tasks than it would be to use a single robot. Then, it is possible to build groups of heterogeneous robots that can work together to accomplish a mission, where each robot has different architecture performing different task in a cooperative and collaborative manner.

If this group of robots is organised, it is called a Colony of Robots. This Colony needs reliable communication among them, in such way that the robots will be able to accomplish their mission even when no robot failures occur. The multi robot system required some knowledge of capabilities of its team-mates, before the start of the mission.

The Colony of Robots can be modeled observing the natural behaviour of insects. They form colonies with individuals that perform different roles in function of the needing of the community. Using this model, it is possible to have colony of robots with some robots in charge of some responsibilities to work with others in a cooperative way to do same tasks, in a collaborative way, to communicate each other to be more efficient or to take decisions in a collective way, etc. in the same form as natural insects. This colony has to have "nest" where the some robot assigns to others what to do, and other robot that receive orders from human and communicate him the results.

It is necessary to formulate, describe, decompose, and allocate problems among a group of intelligent agents; to communicate and interact; to act coherently in actions; and to recognize and reconcile conflicts. This Chapter is focusing in Colony of Robots. This implies to merge several disciplines such as mobile robotics, intelligent agents, ontology, semantics, as well as automatic control, models of communities, communication, and others ones, to have control of a society of robots working together in a collaborative and cooperative way in a non structured environments.

2 Colony of robots

Having a colony of robots has many advantages over a single robot in cost, complexity and capabilities. The group of robots has to be reliable and to act, in some cases, simultaneously, each doing different tasks in a cooperative and collaborative work.

Those groups of cooperating robots have proven to be successful at many tasks, including those that would be too complex for a single robot to complete the objectives. Furthermore, whereas a single complex robot can be easily crippled by damage to a critical component, the abilities of a colony can degrade gradually as individual agents are disabled. In nature, some of the most successful organisms survive by working in groups.

A colony of robots could have the following components and actors (Fig. 1): Center of Colonies, controlled by persons placed far away of the Colony; a Nest of the Colony, a Home base of the robots and the following mobile robots: Colony Leader, Agency Leader, different types of Robots Workers (Working Robots) and Robots Helpers.

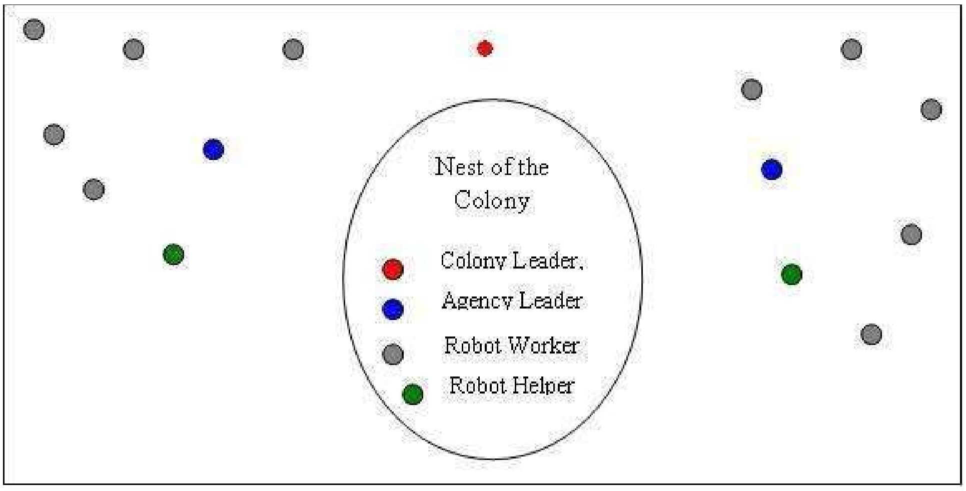


Figure 1: Composition of Colony of Robots

The Colony of Robots has to have Robustness, Fault tolerance, Reliability, Flexibility, Adaptively and Coherence. Robustness refers to the ability of a system to gracefully degrade in the presence of partial system failure. Fault tolerance refers to the ability of a system to detect and compensate for partial system failures.

The group of robot requires robustness and fault tolerance in a cooperative architecture emphasizes the need to build cooperative teams that minimize their vulnerability to individual robot outages. The design of the cooperative team has to ensure that critical control behaviours are distributed across as many robots as possible rather than being centralized in one or a few robots. The failure of one robot does not jeopardize the entire mission. It must ensure that each robot should be able to perform some meaningful task, even when all other robots have failed, on orders from a higher level robot to determine the appropriate actions it should employ. The design has to ensure that robots have some means for redistributing tasks among themselves when robots fail. This characteristic of task reallocation is essential for a team to accomplish its mission in a dynamic environment.

Reliability refers to the dependability of a system, and whether it functions properly each time it is utilized. One measure of the reliability of the cooperative robot architecture is its ability to guarantee that the mission will be solved, within certain operating constraints, when applied to any given cooperative robot team.

Flexibility and adaptively refer to the ability of team members to modify their actions as the environment or robot team changes. Ideally, the cooperative team should be responsive to changes in individual robot skills and performance as well as dynamic environmental changes. In addition, the team should not rely on a pre-specified group composition in order to achieve its mission.

The robot team should therefore be flexible in its action selections, opportunistically adapting to environmental changes that eliminate the need for certain tasks, or activating other tasks that a new environmental state requires. The aim is to have these teams perform acceptably the very first time they are grouped together, without requiring any robot to have prior knowledge of the abilities of other team members. [1]

Coherence refers to how well the team performs as a whole, and whether the actions of individual agents combine toward some unifying goal. The coherence is measured along some dimension of evaluation, such as the quality of the solution or the efficiency of the solution [2]. Efficiency considerations are particularly important in teams of heterogeneous robots whose capabilities overlap, since different robots are often able to perform the same task, but with quite different performance characteristics. To obtain a highly efficient team, the control architecture should ensure that robots select tasks such that the overall mission performance is as close to optimal as possible.

Key Issues of Colony of Robots

Robustness,
Fault tolerance,
Reliability,
Flexibility,
Adaptively,
Coherence,
Distribution,
Cooperation,
Collaboration.

A colony of robots must have fault tolerant; cooperative control; Distributed control; Adaptive action selection, the importance of robot awareness, Inter robot communication, and improving efficiency through learning, Action recognition and Local versus global control of the individual robot.

Previous research in fully distributed heterogeneous mobile robot cooperation includes: one proposes a three layered control architecture that includes a planner level, a control level [3], and a functional level; [4], who describes an architecture that includes a task planner, a task locator, a motion planner, and an execution monitor; [5] who describes an architecture that utilizes a negotiation framework to allow robots to recruit help when needed, [6], who uses a hierarchical division of authority to perform cooperative fire-fighting. However, these approaches deal primarily with the task selection problem and largely ignore the issues so difficult for physical robot teams, such as robot failure, communication noise, and dynamic environments. Since these earlier approaches achieve efficiency at the expense of robustness and adaptively, [1] emphasizes the need for fault tolerant and adaptive cooperative control as a principal characteristic of the cooperative control architecture.

3 Models for Colony of robots

It is necessary have a model of the community, normally taken from the low level intelligence, such as ants, bees or other insects. With one of those models, the robots used are limited in capabilities, such as limitations in computation, actuation and sensing capabilities.

Many colonies of robots have successfully demonstrated cooperative actions, most notably in [7] and [8]. But while much of the existing research has centred on highly-specialized, expensive robots, others create a colony with simple, small and inexpensive robots. It proposes the developing a robot colony, with intelligent distributed behaviours, with the goals: low-cost robots, homogeneous architecture and distributed algorithms. [9]

Multiple agents can often accomplish tasks faster, more efficiently, and more robustly than a single agent could. Groups of cooperating robots have proven to be successful at many tasks, including those that would be too complex for a single robot to complete the aims. Multi-agent systems have been applied to such diverse tasks as complex structure assembly [10], large-object manipulation [11, 12], distributed localization and mapping [13], multi-robot coverage [14], and target tracking [15]. Furthermore, whereas a single complex robot can be easily crippled by damage to a critical component, the abilities of a colony can degrade gradually as individual agents are disabled. In nature, some of the most successful organisms survive by working in groups.

Models of Colony of Robots can divide in two classes: microscopic and macroscopic. Microscopic descriptions treat the robot as the fundamental unit of the model. These models describe the robot's interactions with other robots and the environment. Solving or simulating a system composed of many such agents gives researchers an understanding of the global behaviour of the system. [16]

Microscopic models are reported in [17, 18]; they have been used to study collective behaviour of a swarm of robots engaged in object aggregation and collaborative pulling. Rather than compute the exact trajectories and sensory information of individual robots, the robot's interactions with other robots and the environment are modeled as a series of stochastic events, with probabilities determined by simple geometric considerations and systematic experiments with one or two real robots. Running several series of stochastic events in parallel, one for each robot allows studying the collective behaviour of the swarm.

A macroscopic model, on the other hand, directly describes the collective behaviour of the robotic swarm. It is computationally efficient because it uses fewer variables. Macroscopic models have been successfully applied to a wide variety of problems in physics, chemistry, biology and the social sciences. In these applications, the microscopic behaviour of an individual is complex, often stochastic and only partially predictable, and certainly analytically intractable. Rather than account for the inherent variability of individuals, scientists model the behaviour of some average quantity that represents the system they are studying. Such macroscopic descriptions often have a very simple form and are analytically tractable. It is important to remember that such models do not reproduce the results of a single experiment rather, the behaviour of some observable averaged over many experiments or observations.

The two description levels are related: one can start from the Stochastic Master Equation that describes the evolution of a robot's probability density and get the Rate Equation, a macroscopic model, by averaging it [20]. In most cases, however, Rate Equations are phenomenological in nature, i.e., not derived from first principles. Below we show how to formulate the Rate Equations describing dynamics of a homogeneous robot swarm by examining the details of individual robot controller.

The Rate Equation is not the only approach to modeling collective behaviour. Anderson [19], for example, shows how geometric analysis can be used to predict distribution of individuals playing spatial participative games from the microscopic rules each individual is following.

3.1 Stochastic Approach to Modeling Robotic Swarms

The behaviour of individual robots in a swarm has many complex influences by external forces, many of which may not be anticipated, such as friction, battery power, sound or light signals, etc. Even if all the forces are known in advance, the robots are still subject to random events: fluctuations in the environment, as well as noise in the robot's sensors and actuators. A robot will interact with other robots whose exact trajectories are equally complex, making it impossible to know which robots will come in contact with one another. The designer can take advantage of the unpredictability and incorporate it directly into the robot's behaviour: the simplest effective policy for obstacle avoidance is for the robot to turn a random angle and move forward. So, the behaviour of robots in a swarm is so complex, it is best described probabilistically, as a stochastic process.

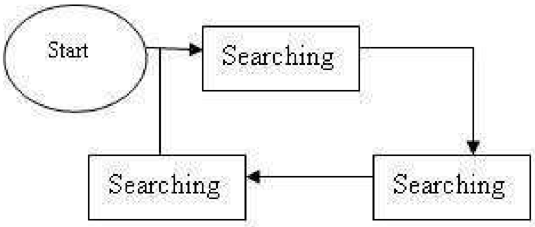


Figure 2: Robot Controller for a simplified foraging scenario

Consider Figure 1, it depicts a controller for a simplified foraging scenario (Collect objects scattered in the arena and assemble them at a "home" location). Each box represents a robot's state the action it is executing. In the course of accomplishing the task, the robot will transition from searching to puck pick-up to homing. Transitions between states are triggered by external stimuli, such as encountering a puck. This robot can be described as a stochastic Markov process, and the diagram in Figure 2 is, therefore, the Finite State Automaton (FSA) of the controller.

The stochastic process approach allows us to mathematically study the behaviour of robot swarms. Let p(n, t)b e the probability robot is in state n at time t. The Markov property allows us to write change in probability density as [16]

$$\begin{aligned} \Delta p(n,t) &= p(n,t+\Delta t) - p(n,t) & (1) \\ &= \sum_{n'} p(n,t+\Delta t|n',t)p(n',t) - \sum_{n'} p(n',t+\Delta t|n,t)p(n,t) & (2) \end{aligned}$$

The conditional probabilities define the transition rates for a Markov process

$$W(n|n',t) = \lim_{\Delta t \to 0} \frac{p(n,t+\Delta t|n',t)}{\Delta t} \tag{3}$$

The quantity $p(n,t)$ also describes a macroscopic variable - the fraction of robots in state n, with Equation 1 describing how this variable changes in time. Averaging both sides of the equation over the number of robots (and assuming only individual transitions between states are allowed), we obtain in the continuous limit ($\lim \Delta t \to 0$)

$$\frac{dN_n(t)}{dt} = \sum_{n'} W(n|n',t)N'_n(t) - \sum_{n'} W(n'|n,t)N_n(t) \tag{4}$$

where $N_n(t)$ is the average number of robots in state n at time t. This is called Rate Equation, sometimes written in a discrete form, as a finite difference equation that describes the behaviour of $N(kT)$, k being an integer and T the discretization interval: $(N(t+T) - N(t))/T$. Equation 3 can be the interpretation: the number of robots in state n will increase in time due to transitions to state n from other states, and it will decrease in time due to the transitions from state n to other states. Rate Equations are deterministic. In stochastic systems, however, they describe the dynamics of average quantities. How closely the average quantities track the behaviour of the actual dynamic variables depends on the magnitude of fluctuations. Usually the larger the system, the smaller are the (relative) fluctuations. [16]

Mathematical models have been applied to study collective collection experiments (aggregation and foraging). In the aggregation experiments, the task was to gather small objects in a single cluster starting from a situation where they were all randomly scattered in an arena. [17, 18]

4 Behaviour of Colony of robots

The behaviour approach to autonomous robot control is based on the observations of animal behaviour, particularly the lower animals, obtaining results without the need for a complex, human-level architecture. Since there are so many varieties of social behaviour in the animal kingdom, the classification proposed by Tinbergen [22] is of particular interest for current robotics research in cooperative systems, as it parallels two possible approaches to cooperating mobile robot development. According to Tinbergen, animal societies can be grouped into two broad categories: those that differentiate, and those that integrate. Societies that differentiate are realized in a dramatic way in the social insect colonies [23]. These colonies arise due to an innate differentiation of blood relatives that creates a strict division of work and a system of social interactions among the members. Members are formed within the group according to the needs of the society. In this case, the individual exists for the good of the society, and is totally dependent upon the society for its existence. As a group, accomplishments are made that are impossible to achieve except as a whole. On the other hand, societies that integrate depend upon the attraction of individual, independent animals to each other. Such groups consist of individuals of the same species that "come together" by integrating ways of behaviour [24]. These individuals are driven by a selfish motivation which leads them to seek group life because it is in their own best interests. Interesting examples of this type of society are wolves and the breeding colonies of many species of birds, in which hundreds or even thousands of birds congregate to find nesting partners.

Such birds do not come together due to any blood relationship; instead, the individuals forming this type of society thrive on the support provided by the group. Rather than the individual existing for the good of the society, we find that the society exists for the good of the individual.

There are two approaches to model cooperative autonomous mobile robots, with behaviour similar to classifications of animal societies discussed above. The first approach involves the study of emergent cooperation in colonies, or swarms, of robots, an approach comparable to differentiating animal societies. This approach emphasizes the use of large numbers of identical robots that individually have very little capability, but when combined with others can generate seemingly intelligent cooperative behaviour. Cooperation is achieved as a side-effect of the individual robot behaviours. A second approach parallels the integrative societies in the animal kingdom, aims to achieve higher-level, "intentional" cooperation among robots. In this case, individual robots that have a higher degree of "intelligence" and capabilities are combined to achieve purposeful cooperation. The goal is to use robots that can accomplish meaningful tasks individually, and yet can be combined with other robots with additional skills to complement one another in solving tasks that no single robot can perform alone. To be purely analogous to the integrative animal societies, robots in this type of cooperation would have individual, selfish, motivations which lead them to seek cooperation [36]. Such cooperation would be sought because it is in the best interests of each robot to do so to achieve its mission. Of course, the possession of a selfish motivation to cooperate does not necessarily imply consciousness on the part of the robot. It is doubtful that we would attribute consciousness to all the integrative societies in the animal kingdom; thus, some mechanism must exist for achieving this cooperation without the need for higher-level cognition. The type of approach one should use for the cooperative robot solution is dependent upon the applications envisioned for the robot team. The differentiating cooperation approach is useful for tasks requiring numerous repetitions of the same activity over a relatively large area, relative to the robot size, such as waxing a floor, agricultural harvesting, cleaning barnacles off of ships, collecting rock samples on a distant planet, and so forth. Such applications would require the availability of an appropriate number of robots to effectively cover the work area while continuing to maintain the critical distance separation. On the other hand, the intentional cooperation approach would be required in applications requiring several distinct tasks to be performed, perhaps in synchrony with one another. Throwing more robots at such problems would be useless, since the individual tasks to be performed cannot be broken into smaller, independent subtasks. Examples of this type of application include automated manufacturing, industrial-household maintenance, search and rescue, and security, surveillance, or reconnaissance tasks.

A Colony of Robots is a behaviour-based system that uses information from the input of the robot's sensors to model the environment. This behaviour is reactive to the changes in its environment. Behaviour-based robots (BBR) usually show more biological-appearing actions than their computing-intensive counterparts, which are very deliberate in their actions. A BBR often makes mistakes, repeats actions, and appears confused, but can also show the anthropomorphic quality of tenacity.

Comparisons between BBRs and insects are frequent because of these actions. BBRs are sometimes considered examples of Weak artificial intelligence, although some have claimed they are models of all intelligence [22]. The term reactive planning starts using 1988, the term has now become a pejorative used as an antonym for proactive. All agents using reactive planning are proactive; some researchers have begun referring to reactive planning as dynamic planning.

In a Colony of Robots, the behaviours of each robot acts independently using simple local rules to form a complex colony-wide behaviour. The simplest of such behaviours that can be demonstrated is swarming. In this behaviour, each robot uses its rangefinders to avoid obstacles (Figure 3).

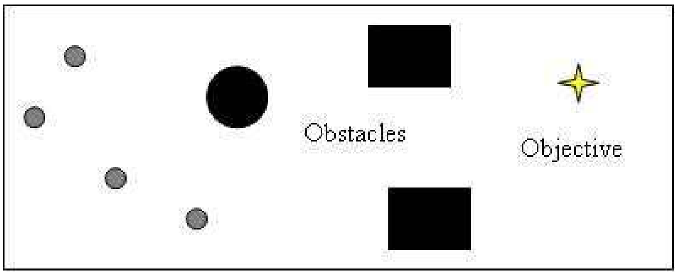


Figure 3: Simple obstacle avoidance behaviour

This simple obstacle avoidance behaviour requires no inter-robot communication and can easily scale without degrading performance as more robots are added.

Adding wireless communication and bearing-only localization allows Colony of robots to perform a lemming like behaviour in which each robot will follow another robot in a chain. One robot assigns itself as the leader and all subsequent robots follow the robot in front of them. This allows the behaviour to scale to arbitrarily long chains of robots.

Colony of Robot has other capabilities to perform more intelligent, collaborative and cooperative behaviours. Using localization data provided in the bearing matrix, Colony of robots can effectively seek a target independently and then converge on the target as a group once it has been found. Figure 4 shows one such scenario in which robots are placed in an unknown environment and work collectively to seek a goal. Once a robot has found the goal, each robot is capable of finding the shortest line-of-sight path to that robot by running a simple graph search on its bearing matrix

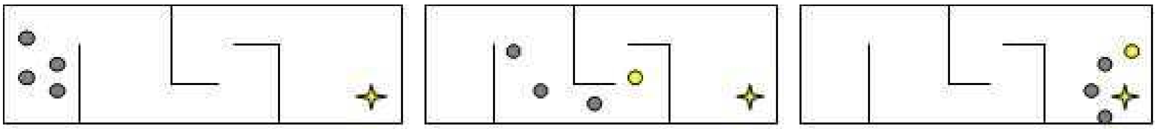


Figure 4: Collective behaviour: a. - Robots begin in random seek, b. - One robot find the goal and communicate to others, c. - Colony arrives to goal.

The Colonies of Robots have to operate without human intervention for a long time, so the Colony must be able to recharge their batteries autonomously. This behaviour needs sensors and infrastructure with three custom pieces of hardware that comprise: a charging station, a medium-range homing sensor, and a battery charging board. The system has to have a task scheduler and bay allocation. Recharging is a complex task involving many actions that the robots can execute. When a low battery signal is detected, the robot will switch from its default task to the recharging task, and it will remain in this state until charging has completed. In the charging task, the robot sends a charging request over wireless to the charging station. If the charging station is full or if another robot is currently attempting to dock, the station denies the request and the requesting robot waits before requesting again. Otherwise, the station assigns the robot a bay using its bay allocation algorithm. The bay allocation algorithm ensures that multiple bays are not assigned to a single robot. If possible, it also attempts to minimize the likelihood of collisions during homing by maximizing the distance between allocated bays. By intelligently managing its docking bays, the charging station can effectively manage a limited number of resources (charging bays) that must be shared between a large numbers of robots. [51]

An important behaviour of Colony of Robots is the interface between the Colony and the Centre of Colonies across the Internet. The core functionally of this interface allows a user to send commands and requests from a client terminal via the Internet to robots and returns responses from the robots back to the client. The system interface (Fig. 5) allows users to remotely monitor and control a large number of robots from anywhere in the world. It consists of three primary modules: the robot library, the server, and the client. The robot library maps high level server requests to basic robot functions and provides an interface for robots communicating over system interface by handling wireless communication to and from the server. The wireless interface between the robots and the server is implemented using a Colony robot programmed to relay wireless data to and from a computer via a USB port and with a server-side wireless library.

It possible to measure of robot's behaviour as components of a theory of robot-environment interaction. Using dynamical systems theory and methods of analysis derived from chaos theory, the quantitative measurement the behaviour of a mobile robot changes if the robot's environment is modified, and the robot's control code is modified.

The behaviour of a mobile robot is the result of properties of the robot itself, the environment, and the control program (the "task") the robot is executing (see figure 6). This triangle of robot, task and environment constitutes a non-linear system, whose analysis is the purpose of any theory of robot-environment interaction.

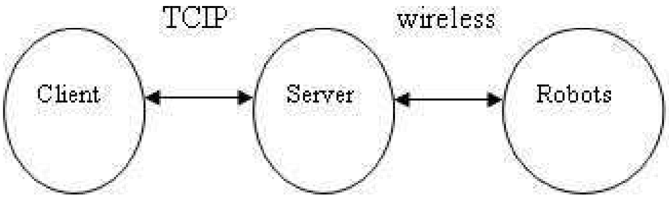


Figure 5: Interface behaviour.

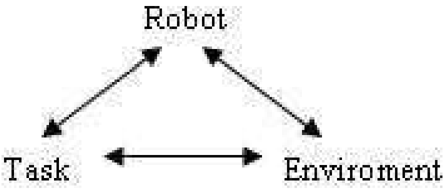


Figure 6: Mobile Robot Interface.

Mobile robotics research to date is still largely dependent upon trial-and-error development, and results often are existence proofs, rather than general, fundamental results. However, independent replication and verification of experimental results are unknown in mobile robotics research practice. One reason for this is the dearth of quantitative measures of behaviour. The research considered the application of dynamical systems theory, specifically deterministic chaos theory to the analysis of robot environment interaction. Using Lyapunov exponent and correlation dimension of the attractor the result of robot's behaviour exhibited deterministic chaos. [52]

5 Multi-robots Systems

In Multi-robots systems have been identified seven primary research topics, when the colony has distributed mobile robot systems. These issues are the following: biological inspirations, communication, architectures, localization/mapping/exploration, object transport and manipulation, motion coordination, and reconfigurable robots.

5.1 Biological Inspirations

Most of the work in cooperative mobile robotics has biological inspirations, imitating social characteristics of insects and animals after the introduction of the new robotics paradigm of behaviour-based control. This behaviour-based paradigm has had a strong influence in the design of the cooperative mobile robotics research. The most common application uses of the simple local control rules of various biological societies, particularly ants, bees, and birds, to the development of similar behaviours in cooperative robot systems. Work has demonstrated the ability for multi-robot teams to flock, disperse, aggregate, forage, and follow trails. The application of the dynamics of ecosystems has also been applied to the development of multi-robot teams that demonstrate emergent cooperation as a result of acting on selfish interests. To some extent, cooperation in higher animals, such as wolf packs, has generated advances in cooperative control. Significant study in predator-prey systems has occurred, although primarily in simulation. In other work implements a pursuit-evasion task on a physical team of aerial and ground vehicles. They evaluate various pursuit policies relaying expected capture times to the speed and intelligence of the evaders and the sensing capabilities of the pursuers.

5.2 Communication, Architectures

The communication in multi-robot teams has been extensively studied. There are implicit and explicit communication, in which implicit communication occurs as a side-effect of other actions, whereas explicit communication is a specific act designed solely to convey information to other robots on the team. Several researchers have studied the effect of communication on the performance of multi-robot teams in a variety of tasks, and have concluded that communication provides certain benefit for particular types of tasks. Additionally, these researchers have found that, in many cases, communication of even a small amount of information can lead to great benefit.

Other work in multi-robot communication has focused on representations of languages and the grounding of these representations in the physical world. This work has extended to achieving fault tolerance in multi-robot communication, such as setting up and maintaining distributed communications networks and ensuring reliability in multi-robot communications.

In [53] communication enabling multi-robot teams to operate reliably in a faulty communication environment. In [54], explores communications in teams of miniature robots that must use very low capacity RF communications due to their small size. They approach this issue through the use of process scheduling to share the available communications resources. [23]

The distributed robotics has focused on the development of architectures, task planning capabilities, and control. This research area addresses the issues of action selection, delegation of authority and control, the communication structure, heterogeneity versus homogeneity of robots, achieving coherence amidst local actions, resolution of conflicts, and other related issues. The architecture, developed for multi-robot teams, tends to focus on providing a specific type of capability to the distributed robot team. Capabilities that have been of particular emphasis include task planning, fault tolerance, swarm control, human design of mission plans, role assignment, and so forth. [25]

5.3 Localization/Mapping/Exploration

Research has been carried out in the area of localization, mapping, and exploration for multi-robot teams. Almost all of the work has been aimed at 2D environments. Initially, most of this research took an existing algorithm developed for single robot mapping, localization, or exploration, and extended it to multiple robots. Some algorithms have developed that are fundamentally distributed. One example of this work is given in [13], which takes advantage of multiple robots to improve positioning accuracy beyond what is possible with single robots.

Another example is a decentralized Kalman-filter based approach to enable a group of mobile robots to simultaneously localize by sensing their team-mates and combining positioning information from all the team members. They illustrate the effectiveness of their approach through application on a team of three physical robots. [56, 57]

Others works develops and analyzes a probabilistic, vision-based state estimation method that enables robot team members to estimate their joint positions in a known environment. Their approach also enables robot team members to track positions of autonomously moving objects. They illustrate their approach on physical robots in the multi-robot soccer domain.

Research approaches to localization, mapping, and exploration into the multi-robot version can be described using the familiar categories based on the use of landmarks, scan-matching, and/or graphs, and which use either range sensors (such as sonar or laser) or vision sensors. [25]

5.4 Object Transport And Manipulation

Enabling multiple robots to cooperatively carry, push, or manipulate common objects has been a long-standing, yet difficult, goal of multi-robot systems. Only a few projects have been demonstrated on physical robot systems. This research area has a number of practical applications that make it of particular interest for study. Numerous variations on this task area have been studied, including constrained and unconstrained motions, two-robot teams versus "swarm"-type teams, compliant versus non-compliant grasping mechanisms, cluttered versus uncluttered environments, global system models versus distributed models, and so forth. Perhaps the most demonstrated task involving cooperative transport is the pushing of objects by multi-robot teams. This task seems inherently easier than the carry task, in which multiple robots must grip common objects and navigate to a destination in a coordinated fashion. A novel form of multi-robot transportation that has been demonstrated is the use of ropes wrapped around objects to move them along desired trajectories. A research explores the cooperative transport task by multiple mobile robots in an unknown static environment. Their approach enables robot team members to displace objects that are interfering with the transport task, and to cooperatively push objects to a destination. In other presents a novel approach for cooperative manipulation that is based on formation control. Their approach enables robot teams to cooperatively manipulate obstacles by trapping them inside the multi-robot formation. They demonstrate their results on a team of three physical robots.

5.5 Motion Coordination

Another topic in multi-robot teams is that of motion coordination. Research themes in this domain that have been particularly well studied include multi-robot path planning, traffic control, formation generation, and formation keeping. Most of these issues are now fairly well understood, although demonstration of these techniques in physical multi-robot teams (rather than in simulation) has been limited.

The motion coordination problem in the form of path planning for multiple robots is presented that performs path planning via checkpoint and dynamic priority assignment using statistical estimates of the environment's motion structure. Additionally, they explore the issue of vision-based surveillance to track multiple moving objects in a cluttered scene. The results of their approaches are illustrated using a variety of experiments. [25]

The contributions to the Robot Motion Planning (RMP) field throughout a 35-year period, from classic approaches to heuristic algorithms, surveying around 1400 papers in the field, the amount of existing works for each method is identified and classified, concluding that only about 3% of papers dealt with heuristic algorithms. [45]

There exits different approaches for coordination of multiple robots, considering integration of communication constraints in the coordination of robots. In Yamauchi approach, uses a technique for multi-robot exploration which is decentralized, cooperative and robust to individual failures. He demonstrated a frontier-based exploration which can be used to explore office buildings. He used evidence grids which are Cartesian grids, which store the probability of occupancy of the space (prior probability equal 0, 5). The robots create their evidence grids by using their sensors, classifying each cell as Open, occupancy probability < prior probability; Unknown, occupancy probability = prior probability; and Occupied: occupancy probability > prior probability. In this manner, any open cell which is near to an unknown cell is labeled as frontier edge cell. Frontier regions are formed by adjacent frontier edge cells. The robots have to move to boundary between open space and unexplored part of the space to gain much new information. After a robot detected frontiers in the evidence grid, it tries to go to the nearest frontier. Besides, robots use path planner to find the nearest unvisited frontier and reactive obstacle avoidance behaviours to hinder collisions with unseen obstacles on the evidence grid.

After reaching and performing a 360 degree sensor sweep in the frontier, it adds the new information to the evidence grid of its local map. In multi-robot exploration, there is a local evidence grid which is available for all the robots. Besides that, every robot is creating its own global evidence grid. This global evidence grid shows its knowledge about the environment for the robot. Using two separate evidence grid gives the advantage of being decentralized and cooperative. For instance, when a robot detects a new frontier, it starts to travel this point. After reaching this point, it performs a sensor sweep. By this way, it updates its local evidence grid with the new information. Moreover, it transmits updated local evidence grid information to the other robots. Besides that, global evidence grid is integrated with local evidence grid in a straightforward way. Using this cooperative approach, all new information is available for the other robots. Thus, each robot can update its own global evidence grid. There are two advantages of sharing a global map. Firstly, robots can make decision about which frontiers are unexplored yet by using updated maps. This improves the efficiency of exploration. Additionally, if a robot is disabled in the area, this won't affect the other robots. In his study, he developed a coordination approach based on frontiers. His frontier based approach is became a milestone for the following researches. [26]

The approach of R. Simmons, the coordination among robots is done to explore and create a map. This multi-robot exploration and mapping which is based on cost of exploration and estimation of expected information gain. They decreased the completion time of creating map task by keeping the robots well separated, resulting of the minimizing the overlap in information gain between the robots. Besides, they distributed most of the computation, which takes place in exploring and creating the map. Global map is constructed by the distinctive sensor information of the robots same as Yamauchi's approach. As a result, creating a consistent global map and assigning task to each robot which maximizes the overall utility are efficient examples of coordinated mapping and coordinated exploration, respectively. Besides, there are three important achievements with their approach: they used same software for both the local and global mapping; the robots update the global map with new information, even if they cannot communicate each other directly. This minimizes the alignment in local maps. Lastly, after each robot creates its local map, it sends local map to central mapper module. In the process of combining the maps to create one global map, central mapper module combines the data iteratively. Thus the localization error is minimizing. Their approach to distribute most of the computation among the robots is remarkable. However, they have two assumptions in this situation. Firstly, robots know their position relative to another.

More sophisticated methods must be found for mapping and localization where the initial positions are not known by robots. Secondly, the researchers assumed that robots have an access to high-bandwidth communication, [27].

Another approach proposed, is based on separating the environments into stripes. These stripes show the successively explored environment by the multi-robots. However, in this approach, if one robot moves to a point, the rest of the robots wait on their position and watch for the moving robot.

This approach significantly decreases odometry error, however it is not designed for distribution of the robots and the robots tend to stay close. [28]

The coordination of multi robots combines the global map; central mapper distributes the new map. Additionally, robots produce new bids from the updated map information. By using these bids, robots can mark the map cells as obstacle, clear or unknown.

They used the frontier-based algorithm for exploration which is found by Yamauchi. However, there are two modifications to frontier-based approach. Firstly, they determined the estimation of the cost of traveling a frontier cell by calculating the optimal paths from the robots initial positions. These computations are made simultaneously by using a flood-fill algorithm. By determining the cost of traveling to a frontier cell for each robot, they assign this exploring task to the robot which has the optimal path. Secondly, they estimated information gain from the frontier cell by creating a rectangle which approximates the information gain region. Thus, executive uses these rectangles to estimate potential overlaps on coverage. By finding the cost of traveling and estimating the information gain in the frontier cells, executive assigns tasks to the robots. The idea behind the assigning tasks is discounting. For example, executive finds the bid location with the highest utility for a robot. After that, it discounts the bids of the remaining robots, selects another robot which has highest utility among the other robots. This task assignment continues till no robot or no task remains.

A new approach for coordination uses market-based approach, by minimizing the costs and maximizing the benefits. Like the previous approaches, robots communicate with each other continuously to receive new information about the environment. Thus, robots can improve their current plans. Even though there is a central agent, they are not dependent the central agent. However, in exploration process, if the central agent is reachable, the robots are communicating with central agent to learn if there are new goal points. [29]

Auction methods have been investigated as effective, decentralized methods for multi-robot coordination. Theoretical analysis and experimental of the performance of auction methods for multi-robot routing, has shown great potential. These methods are shown to offer theoretical guarantees for a variety of bidding rules and team objectives.

The problem of routing in multi-robot is specified by a set of robots, $R = r1, r2, \ldots, rn$, a set of targets, $T = t1, t2, \ldots, tm$, their locations, and a non-negative cost function $c(i, j), i, j \in R \cup T$, which denotes the cost of moving between locations i and j. Assuming that these costs are symmetric, $c(i, j) = c(j, i)$, are the same for all robots, and satisfy the triangle inequality. Travel distances and travel times between locations satisfy these assumptions in any typical environment. The objective of multi-robot routing is to find an allocation of targets to robots and a path for each robot that visits all targets allocated to it so that a team objective is optimized. In Auction methods the team objectives could be: MINISUM: Minimize the sum of the robot path costs over all robots, MINIMAX: Minimize the maximum robot path cost over all robots. MINIAVE: Minimize the average target path cost over all targets. The robot path cost of a robot r is the sum of the costs along its entire path, from its initial location to the last target on its path. The target path cost of a target t is the total cost of the path traversed by robot r from its initial location up to target t, where r is the unique robot visiting t. Optimizing performance for any of the three team objectives is NP-hard, considering that there is no polynomial time algorithm for solving multi-robot routing optimally with the MINISUM, the MINIMAX, or the MINIAVE objective, unless $P = NP$.

The main advantage of this multi-round auction mechanism is its simplicity and the fact that it allows for a decentralized implementation on real robots. Initially, each robot needs to know its own location, the location of all targets, and the number of robots (the number of bids in each round), but not the locations of the other robots. In each round, each robot computes its single bid locally and in parallel with the other robots, broadcasts the bid to the other robots, receives the bids of the other robots, and then locally determines the winning bid. This procedure is repeated in every round of the auction. Broadcasting can be achieved by means of relaying messages from robot to robot. Clearly, there is no need for a central auctioneer, and therefore, there is no single point of global failure in the system. Notice also the low communication complexity; each robot needs to receive n numbers (bids) in each of the m rounds, therefore O(nm) numbers need to be communicated over any single link. [30]

5.6 Reconfigurable systems

The motivation for reconfigurable distributed systems of this work is to achieve function from shape, allowing individual modules, or robots, to connect and re-connect in various ways to generate a desired shape to serve a needed function. These systems have the theoretical capability of showing great robustness, versatility, and even self-repair.

Most of the work in this area involves identical modules with interconnection mechanisms that allow either manual or automatic reconfiguration. These systems have been demonstrated to form into various navigation configurations, including a rolling track motion, an earthworm or snake motion, and a spider or hexapod motion. Some systems employ a cube-type arrangement, with modules able to connect in various ways to form matrices or lattices for specific functions.

An important example of this research presents a biologically inspired approach for adaptive communication in self-reconfigurable and dynamic networks, as well as physical module reconfiguration for accomplishing global effects such as locomotion. [61]

6 Colony of Ant robots

Social insects that live in colonies, such as ants, termites, wasps, and bees, develop specific tasks according to their role in the colony. One of the main tasks is the search for food. Real ants search food without visual feedback (they are practically blind), and they can adapt to changes in the environment, optimizing the path between the nest and the food source. This fact is the result of stigmergy, which involves positive feedback, given by the continuous deposit of a chemical substance, known as pheromone. Ants are social insects capable of short-range interactions, yet communities of ants are able to solve complex problems efficiently and reliably. Ants have, therefore, become a source of algorithmic ideas for distributed systems where a robot (or a computer) is the "individual" and a swarm of robots (or the network) plays the role of the "colony". Ant robots are simple and cheap robots with limited sensing and computational capabilities. This makes it feasible to deploy teams of ant robots and take advantage of the resulting fault tolerance and parallelism.

They cannot use conventional planning networks due to their limitation and their behaviour is driven by local interactions. Ant Robots almost never know exactly where they are in the environment A common way is to use probabilistic planning, provides to robots, the best possible location estimate, to achieve their goals without ever worrying about where they are in the terrain. Other approach of Ant robots can communicate via markings that they leave in the terrain, similar to ants that lay and follow pheromone trails, and solving robot-navigation tasks in a robust way. Using Pheromone Traces of alcohol, heat, odor [33], and virtual traces [56, 57] no location is estimates, no planning is need, no direct communication with a simpler hardware and software. The result is a very robust navigation. It has been developed a theoretical foundation for ant robotics, based on ideas from real-time heuristic search, stochastic analysis, and graph theory. [31, 32]

Teams of robots can do mine sweeping, surveillance, search-and-rescue, guarding, surface inspection and many others work. For example, a team of robots that cover terrain repeatedly can guard a museum at night. [33] The main areas, involved to have group of robots are: Agent coordination (swarms), Robotics (robot architectures, ant robots, sensor networks), Search (real-time search), complexity analysis of graph algorithms, Communication.

6.1 Ant colony optimization

The ants construct a pheromone trail in the search for a shorter path from nest to the food. When an obstacle is inserted in the path, ants spread to both sides of the obstacle, since there is no clear trail to follow. As the ants go around the obstacle and find the previous pheromone trail again, a new pheromone trail will be formed around the obstacle. This trail will be stronger in the shortest path than in the longest path. As shown in [58], there are many differences between real ants and artificial ants, mainly: artificial ants have memory, they are completely blind and time is discrete. On the other hand, an ant colony system allows simulation of the behaviour of real-world ant colonies, such as: artificial ants have preference for trails with larger amounts of pheromone, shorter paths have a stronger increment in pheromone, and there is an indirect communication system between ants, the pheromone trail, to find the best path.

The term collective behaviour, in the robotics literature, means: joint collaborative behaviour that is directed toward some goal in which there is a common interest; a form of interaction, usually based on communication; and joining together for doing something that creates a progressive result such as increasing performance or saving time. Cooperative behaviour is to associate with another or others for mutual, often economic, benefit.

A collaborative robot is a robot designed, for example, to assist human beings as a guide or assistor in a specific task. A cooperative robots are a group of robots that can work together to move large objects, sharing the load.

Having coordination in actions, sharing sensors and computing power, multi robots can perform tasks such as drill holes and pitch tents in tight coordination. They can carry out the tasks in an unstructured outdoor environment.

7 Multi-agents Systems applied to Colony of Robots

Using decentralized approaches, as Multi-Agent Systems, gives a new way for modeling a colony of robots. This approach is able to generate a self-organized system with some robust and efficient ways of solving problematic like some insect societies, as ants. [34]

This focus provides three multi-level advances: (i) the development of the concept of Complex Systems give a new way of modeling, based on decentralized representation composed of interaction network of entities from where emergent properties appear ; (ii) Object-Oriented programming had proposed a first step in the decomposition of computing and so in the following, agent oriented programming adds to objects some autonomous properties; (iii) the development of huge computer networks promote the distributed computing which finally allowed implementations of these previous concepts. [35, 48]. Complex systems are usually presented as some systems of interacting entities which can be represented as a kind of networks. Agent-based can represent the interactions between entities as communication processes.

7.1 Robot Architecture

Robot architecture has been proposed, based on a multi-agent system (MAS). The agents have the goal: to control the robot and to do it intelligent, while competing for resources. This approach produce a more robust, flexible, reusable, generic and reliable architecture that can be easily modified and completed to permit social behaviour among robots; it is also holonic multi-agent systems. The agents that make up the proposed architecture may also be Multi-agent systems themselves. The Task Planning Agent is a multi-agent system formed by planner agents and a coordination agent. [60]

7.2 Multi-agent plan coordination

The Multi-agent plan coordination problem arises whenever multiple agents plan to achieve their individual goals independently, but might mutually benefit by coordinating their plans to avoid working at cross purposes or duplicating effort. Although variations of this problem have been studied, there is no agreement over a general characterization of the problem. A general framework that extends the partial order, causal-link plan representation to the Multi-agent case, and that treats coordination as a form of iterative repair of plan flaws that cross agents. Multi-agent planning has acquired a variety of meanings over the years. In part, this may be due to the ambiguity of exactly what someone considers to be "multi-agent" about the planning. In some work, it is the planning process that is multi-agent; for example, multiple agents, each with specialized expertise in certain aspects of planning, might collaborate to formulate a complex plan that none of them could have generated alone. In other work, it is the product of planning, the plan itself, that is multi-agent, in the sense that it specifies the activities of multiple actors in the environment such that they collectively achieve their individual and/or common goals. And sometimes, it is both where multiple agents interact to arrive at plans that will be carried out by multiple (and often, it is implicitly assumed, the same) agents. In the third class of problems, a Multi-agent Plan Coordination Problems (MPCPs), in which multiple agents, has each plan the individual activities, but might mutually benefit by coordinating their plans to avoid interfering with each other unnecessarily duplicating effort. Multi-agent plan coordination differs from "team planning", in which agents must work together more tightly as a team in order to achieve their joint goals.

Instead, multi-agent plan coordination is suited to agents that are loosely-coupled (nearly independent), where each agent can achieve its own goals by itself, but the presence of other agents who are also asynchronously operating in the same environment leads to potential conflicts and cooperative opportunities. [36] Other similar approaches is described, where the problem of finding plans with minimal make span is considered. In both, the degree of coupling measures, the degree of interaction between different plans (threads]) and thus affects the inherent difficulty of the planning problem. In general, the multi-agent plan coordination problem is known to be NP-Hard. It has developed a rigorous computational theory of single-agent and multi-agent plan coordination, and implemented an efficient and optimal algorithm that, under assumed characteristics, is polynomial with respect to the size of the plan coordination problem. [37, 38]

The autonomy of colony of robots depends on the behaviour of each agent associated to each robot in terms of actions and in terms of flexible group decision making.

To achieve this objective, it is necessary that agent architectures can help in designing the architecture of the software of each robots; a multi-agent approach can address the problem of interactions between robots; and automated planning can provide the basis of robots intelligence.

A vehicle can be adapted to act as robots. It has proposed architecture and distributed planning method for multi-vehicle missions contribute to the increase of vehicle intelligence and autonomy. The integration of online planning, disruptive events in absence of human intervention do not lead necessarily to aborting the mission. However, it is important to note that the architecture addresses a specific class of multi-vehicle missions. For this class the plan exists at the beginning of the mission and provides actions up to the end of the mission. In a context where there is a large uncertainty about the ending conditions of the mission or where there are systematically a large difference between the situation expected at planning time and the actual situation, other architectures based on a more systematic activation of the planning module are more suited.

7.3 Cooperative Planning

The problem of the cooperative planning is very complex. To specify a planning task for the system of entities more precisely, it has to know about how the system □ shares the knowledge; how precise is the map of the working environment; when the environment is static or changing; and the kind of a task is being solved. When system has multiple entities, it can be organized in a centralized way, or a decentralized. Centralized systems have a central control unit managing tasks for other entities, the knowledge about the solved tasks and working environment, system configuration, actual state of the system (positions of particular entities) is stored and maintained in the central unit. It also distributes local tasks according to priorities to other entities. On the other hand, knowledge in decentralized systems is shared among all entities where each entity plans and performs its own activities autonomously with respect to needs and request of other entities. Advantages of centralized systems are in terms of traffic control, resource management, and task optimization. On the other hand decentralized systems are superior in terms of robustness and scalability of the systems. Robustness can be defined as the ability of a system to gracefully degrade when some entity in the system fails. Robust systems are able to work properly even in the case that any entity is malfunctioning, as long as there are some functioning robots. Centralized systems do not have this characteristic because failure of the central unit disables the entire system. It can take advantage of both systems, using hybrid solutions. [39]

7.4 Planning task

Planning task is the amount of information about working environment. If the map of working area is available and accurate, the planning problem leads to geometrical optimization, computed off-line. When robots are operating in unknown environments, the task cannot be divided optimally without complete information. This problem is solved on-line because new information about environment can be obtained during fulfilling the plan. Another situation sets in when a map is available but the working environment is rapidly changing or a map is not precise. In this case, primary activities can be planned off-line based on the available map. Concrete actions can be specified on-line in more details with respect to acquired information about the environment in which the system operates.

If the colony of robots has only three kinds of tasks such as coverage problem: exploration, and coordinated planning. Coverage planning for a team of robots deals with the problem ensuring that every point in the working environment is visited by at least one robot.

Coverage planning can be used for a number of different tasks, for example floor cleaning, grass cutting, foraging and mine detection and removal. Exploration of a working area is a similar task; the goal is not to "reach" all places in the environment, but to "see" all places. The main aim during exploration is how to move particular robots in order to minimize the time needed to completely explore the environment. When mobile robots have the ability to explore a surrounding environment efficiently, they are able to build a model (map) of their environment and to solve more complex tasks that ensue from mapping like the detection of specified objects. The key question of coordinated planning is how to plan paths for particular robots in order to avoid collisions. In other words, two robots cannot be at the same place in the same time.

7.5 Planning multiple robots

Planning for multiple robots is one of the main research topics within multi-robots systems [40]. Differences in the approaches are mainly in methods of knowledge sharing. Multi agent approaches and techniques are applicable (mainly in task of the distribution for multiple entities) but it is necessary to consider a low preciseness of localization and mapping strategies. An example of the application of the behaviour-based multi agent approaches with consideration to physical multi-robot systems is presented in [41].

The common attribute of all planning problems is the effort to find the optimal solution. A typical criterion to be optimized is the overall time spent by all team members during the task execution or the sum of lengths of particular robot's paths. The type of criterion leads to applicability of different strategy planning for multiple robots. Basic problem of the planning is the path planning. The research for robot's path planning has centred on the problem of finding a path from a start location to a goal location, while minimizing one or more parameters such as length of path, energy consumption or journey time. The optimal path planning is the essential problem of the exploration and the coverage task. [42]

7.6 Multi-agent Learning

Multi-Robot Systems MRS can often be used to fulfil the tasks with uncertainties, incomplete information, distributed control, and asynchronous computation, etc. The performance of MRS in redundancy and co-operation contributes to task solutions with a more reliable, faster, or cheaper way. Multi-agent reinforcement learning can be useful for multi-robot systems The challenges in MRSs involve basic behaviours, such as trajectory tracking, formation-keeping control, and collision avoidance, or allocating tasks, communication, coordinating actions, team reasoning, etc. For a practical multi-robot system, firstly basic behaviours or lower functions must be feasible or available. In upper modules, for task allocation and planning, have to be designed carefully. Robots in MRSs have to learn from, and adapt to their operating environment and their counterparts. Thus control and learning become two important and challenging problems in MRSs. [43]

Multi-agent reinforcement learning RL allows participating robots to learn mapping from their states to their actions by rewards or payoffs obtained through interacting with their environment. Robots in MRSs are expected to coordinate their behaviours to achieve their goals. These robots can either obtain cooperative behaviours or accelerate their learning speed through learning. [44]

Among RL algorithms, Q-learning has attracted a great deal of attention. Explicit presentation of an emergent idea of cooperative behaviours through an individual Q-learning algorithm can be found. Improving learning efficiency through co-learning was shown. The study indicates that K co-operative robots learned faster than they did individually. It has also demonstrated that sharing perception and learning experience can accelerate the learning processes within robot group. Recently there has been growing interests in scaling Multi-agent RL to MRSs. Although RL seems to be a good option for learning in Multi-agent systems, the continuous state and action spaces often hamper its applicability in MRSs. Fuzzy logic methodology seems to be a candidate for dealing with the approximation and generalization issues in the RL of Multi-agent systems. However, this scaling approach still remains open. [45]

7.7 Ontology and Semantic

Several methodologies exist for building multi-agent systems; few of them address the information domain of the system. Just as important as the system's representation of the information domain is the various agents' information domain view.

Heterogeneous systems can contain agents with differing data models, a case that can occur when reusing previously built agents or integrating legacy components into the system. Most existing methodologies lack specific guidance on the development of the information domain specification for a multi-agent system and for the agents in the system.

An appropriate methodology for developing ontology's must be defined for designers to use for specifying domain representations in multi-agent systems. The existing methodologies for designing domain ontologies are built to describe everything about a specific domain; however, this is not appropriate for multi-agent systems because the system ontology should only specify the information required for proper system execution. The system ontology acts as a prerequisite for future reuse of the system, as the ontology specifies the view of the information domain used by the multi-agent system.

Any system that reuses the developed multi-agent system must ensure that the previously developed system ontology does not conflict with the ontology being used in the new system.

Once the system ontology is constructed, a multi-agent system design methodology should allow the analyst to specify objects from the data model as parameters in the conversations between the agents. To ensure the proper functionality of the multi-agent system, the designer must be able to verify that the agents have the necessary information required for system execution. Since the information is represented in the classes of the data model, the design of the methodology must show the classes passed between agents. [46]

7.8 Multi-agent Systems Engineering (MaSE)

Multi-agent Systems Engineering is an attempt to answer the sixth challenge, how to engineer practical multi-agent systems, and to provide a framework for solving the first five challenges. It uses multi-agent systems for developing intelligent, distributed software systems. MaSE uses two languages to describe agents and multi-agent systems: the Agent Modeling Language (AgML) and the Agent.

Definition Language (AgDL), a specific methodology for formal agent system synthesis. Both AgML and AgDL will be defined with a precise, formal semantics. The methodology can also be successfully applied with traditional software implementation techniques as well.

There are a lists six challenges of multi-agent systems: to decompose problems and allocate tasks to individual agents. To coordinate agent control and communications.

To make multiple agents act in a coherent manner. To make individual agents reason about other agents and the state of coordination. To reconcile conflicting goals between coordinating agents. How to engineer practical multi-agent systems. AgML and AgDL semantics are based of multi-sorted algebras. Algebraic approaches have the advantage that there has been a great deal of work in automatically synthesizing code from algebraic specifications. The work is similar in many respects to the agent methodologies based on object-oriented concepts. However, few of these have a formal basis. Some work in formalization of agent systems has been performed in but has focused on formal modeling and not automated code synthesis. MaSE and AgML together provide many advantages over traditional software engineering techniques. Because of this abstraction, MaSE can capture traditional object-oriented systems as well as agent-based systems for which traditional techniques are inappropriate. Then, it has a more concise representation than object-oriented techniques. It has a formal syntax and semantics. [45, 47]

8 Summarizing Colony of Robots

In this paragraph is presented a synthesis what would be a Colony of Robots, its characteristics, its components, its applications, its basis and the way to build the Colony as an Engineering Project. [50]

8.1 Parts of a Colony of robots

A colony could have the following components and actors: Centre of Colonies, Nest of the Colony, Colony Leader, Agency Leader, and different types of Working Robots.

Centre of Colonies is a place controlled by human beings. It is located far from zone of work of the colony. Persons determine the works that the Colony must realize, when initial and final time, in what place, which it is necessary to do and when the Colony must report its work. The orders of work are sent to the nest of the Colony in a remote way.

Nest of the Colony is where the Colony is placed, composed of several heterogeneous autonomous robots, which has a Colony Leader and several Agency Leaders who assign works.

Colony Leader is a robot that has communication with the Centre of Colonies. This Leader receives the orders of work from Centre of Colonies; it sends reports of realized work, problems happened in the work or in the colony; it decomposes orders into tasks and assigns a task to the different Agency Leaders Leading. It receives the state of works and report s to Agency Leaders.

Agency Leader, are robots that have communication with the Colony Leader and with Working Robots or Robot Agents. It receives orders, from Colony Leader, of the task that it is necessary to realize, determines that the needs of Working Robots for the task and sends orders to go to the place of work and to do the task.

Working Robots are a group of heterogeneous mobile robots that receives orders Agency Leader, to move to a workplace to carry out a given task.

The Working Robots communicate themselves to determine the best path to arrive to the workplace, to warn if they need help to do a task, to report the work realized, to report when it has faults or it has little energy.

The Working Robots also communicate with an Agency Leader, reporting the works done, the problems of the group of robots and the need of help, etc. It could be some types of these robots are specialist. One of them can have computer vision systems; others it have manipulation systems of objects; another ones with derricks systems to transport robot; robots repairers, etc. [50]

8.2 The Characteristics of the Colony

A Colony of Robots would have to have defined its characteristics. The main characteristics can selected for the persons in the Centre of Colonies. Normally, the Colony characteristics must be: Model of the Colony; Operational Environment; Communication in the Colony; Colony Coordination; Cooperative and Collaborative Work, among the Working Robots to carry out works; Robustness, Fault tolerance, Reliability, Flexibility, Adaptively and Coherence; Reconfigurability, Localization, Mapping, Exploration, Object transport and manipulation. [50]

Model of the Colony. The colony must have a biological inspiration model of how operating. It has to have a model to imitate, such as animal colony or of insects colony. There are several models developed, usually it is an Ant Colony Model.

Operational Environment. The colony of robots must be designed for thinking in the environment that it will settle. According to this way, different types of robots are needed. If an average normal environments, mobile robots need to be designed to realize works in that environment. If they are extreme environments, the robots must be designed for those conditions. An alternative is "to automate" a vehicle that already exists, in the way of which it could be autonomous and apt to do the tasks that it are wanted.

Communication in the Colony. The communication is important in a Colony. These communications have to be reliable, precise and fault tolerant. The communications are among the Centre of Colonies and the Colony; among Colony Leader and Agency Leaders; among these Agency Leaders and the Working Robots; and among Working Robots.

Colony Coordination. Coordination must exist between the different Working Robots of the Colony for the movement to the work place; in the cooperation and collaboration among robots. In general, it is desired that the colony be capable of coordinating path planning, traffic control, formation generation, and formation keeping.

Cooperative and Collaborative Work, among the Working Robots to carry out works. Robustness, Fault tolerance, Reliability, Flexibility, Adaptively and Coherence in all the different robots in the Colony.

Reconfigurability. The colony must be reconfigurable distributed systems to achieve function from shape, allowing robots to connect and re-connect in various ways to generate a desired shape to serve a needed function. These systems have to have capability of showing great robustness, versatility, and even self-repair.

Localization, Mapping, Exploration. The colony of robots must be capable of knowing the coordinates where it is and where that it must go, using GPS systems (Global Positioning System). It has to be capable of creating a map of the tour and of being able to explore. This information must be shared for groups of robots involved in a work.

Object transport and manipulation the colony must be capable of manipulating, transporting or pushing simple objects, in individual way or in collaboration and / or cooperation way, with other robots.

8.3 Colony wanted and for what

What is expected from a colony is that it realizes the works assigned in an efficient, reliable form and it reports the works done, achievements and problems. The works can be very diverse depending what the persons want in the Centre of Colonies. The Colony can be utilized in Exploration, in Industries, at Home, in Military Forces, in Education, and many others uses.

In Exploration. When a Colony of Robots has assigned works to know certain area, the exploration has to know, for example, temperature, pressure, ozone level, radioactivity, etc. In general, it is needed to measure physical or chemical variables, to inspect the place, searching path, to make images of the environment, animals, vegetables, to see the composition of the area, water, etc.

This exploration can realize in different locations to measure the environment, especially if it is hostile (volcanoes, deserts, Antarctic, Arctic, etc.). Other kind of exploration is the spatial exploration to know the Moon and Mars.

In Industries. There can have groups of robots to do maintenance of pipelines, for painting, for making measurements and to do works in dangerous zones, etc. Also, the group can be useful in flexible Manufacturing Systems FMS; in the agricultural industry to detect plagues of insects, to determine when it is necessary to harvest, etc.; in industries of production of raw materials and production of energy.

At homes. To improve quality of live of the persons. For example groups of robots to sweep, to clean, to paint etc.

In Education; in Military Forces and many Others.

8.4 How the Colony of Robots is built

The colony of robots is constructed from the specifications given by the users. Though the principles are the same, the design of the Centre of the Colony, the Nest of the Colony, the different robots, they depend on the works to be realized, on the environment in which they are going to work and the desired precision. According these parameters and others specifications, it is necessary to do an Engineering Project to have the Colony of Robots needed. This Project includes the support of providing companies of Robotics, Automation, Communications, Electronic, etc.

8.5 Colony of Robots Background

The Colonies of Robots base on physical principles, in general on scientific principles, and researches on Robotics, Computer Science, Mechanical, Automatic, Communications, Optimization and Planning, and many other fields related, shown in this chapter.

8.6 Research in Colonies of Robots

Though, there exist several groups doing research on colonies of robots, the motivations can be very different. Some Universities and Research Centres do works applicable to a

Colony of Robots. It is better to have a multidisciplinary teams to creating a Colony of Robots.

9 Conclusions

In this chapter has been presented an overview in colony of robots, considering models of communities of robots, the behaviour of groups of robots, ant robots colony, communication inside de colony, multi-robots characteristics and multi-agent systems applied to multi-robots.

To have control of a colony of robots that working together in a collaborative and cooperative way in a non structured environments, is important considerer the communication among the robots, the way of planning and coordination of the robots, how the object has transport and manipulation, and reconfiguration. The colony has to have a cooperative architecture with robustness, a good fault tolerance, to be reliability, to present flexibility, adaptively and coherence. Communication among robots of the colony permits to have a distributed control, planning and coordination to perform the tasks assign for achieving the aims of the colony.

It is expected that the colony of robots has a behaviour as insects, with intelligent distributed behaviours that can do well the tasks, in a cooperative and collaborative way. This colony could have low-cost robots, heterogeneous architecture and distributed algorithms.

The kinds of tasks of the colony of robots normally are exploration, and coordinated planning. To exhibit intelligence, the robot architecture in the colony, is based on a multi-agent system. In this manner, is possible to have control of each robot in an intelligent, way.

This approach produce a more robust, flexible, reusable, generic and reliable architecture that can be easily modified and completed to permit social behaviour among robots. Each robot is a Multi-agent system to be more efficient. Multi-agent planning and multi-agent coordination inside the colony are good solutions. The applications of colonies of robots several fields in industries, scientific exploration and at home. A guideline what considering in create a Colony of Robots has been presented.

References

[1] Lynne E. Parker, Heterogeneous Multi-Robot Cooperation. MIT 1994.

[2] Alan Bond and Less Gasser, Readings in Distributed Artificial Intelligence. Morgan Kaufmann, 1988.

[3] Fabrice R. Noreils. Toward a robot architecture integrating cooperation between mobile robots. Application to indoor environment. The International Journal of Robotics Research, 12(1), 79-98, February 1993.

[4] Philippe Caloud, Wonyun Choi, Jean-Claude Latombe, Claude Le Pape and Mark Yim. Indoor automation with many mobile robots. In Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, pages 67-72, Tsuchiura_ Japan, 1990.

[5] H. Asama, K. Ozaki. A. Matsumoto, Y. Ishida and I. Endo, Development of task assignment system using communication for multiple autonomous robots. Journal of Robotics and Mechatronics. 4(2):12-127, 1992.

[6] Paul Cohen, Michael Greenberg, David Hart and Adele Howe. Real-time problem solving in the Phoenix environment. COINS Tech.l Report, University of Massachusetts at Amherst, 1990.

[7] J. McLurkin, Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots. M.S. diss., Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass, 2004.

[8] A. Howard, L. Parker and G. Sukhatme, Experiments with a Large Heterogeneous Mobile, 2006.

[9] J. Cortes, S. Martnez, T. Karatas and F. Bullo, Coverage Control for Mobile Sensing Networks. In Proceedings of the IEEE Conference on Robotics and Automation, 1327-1332. Arlington, VA, 2002.

[10] Felix Duvallet, James Kong, Eugene Marinelli, Kevin Woo, Austin Buchan, Brian Coltin, Christopher Mar and Bradford Neuman, Developing a Low-Cost Robot Colony, AAAI Fall Symposium 2007 on Distributed Intelligent Systems, 2007.

[11] F. Heger and S. Singh, Sliding Autonomy for Complex Coordinated Multi-Robot Tasks: Analysis & Experiments. In Proceedings, Robotics: Systems and Science, Philadelphia, 2006.

[12] T. G. Sugar and V. C. Kumar, Control of Cooperating Mobile Manipulators. In IEEE Transactions on Robotics and Automation, Vol.18, No.1, 94-103, 2002.

[13] A. Trebi-Ollennu, H. D. Nayar, H. Aghazarian, A. Ganino, P. Pirjanian, B. Kennedy, T. Huntsberger and P. Schenker, Mars Rover Pair Cooperatively Transporting a Long Payload. In Proceedings of the IEEE International Conference on Robotics and Automation, 2002.

[14] D. Fox, W. Burgard, H. Kruppa and S. Thrun, A probabilistic approach to collaborative multi-robot localization. Autonomous Robots 8(3), 2000.

[15] J. Cortes, S. Martnez, T. Karatas and F. Bullo, Coverage Control for Mobile Sensing Networks. In Proceedings of the IEEE Conference on Robotics and Automation, 1327-1332. Arlington, VA, 2002.

[16] M. Hundwork, A. Goradia, X. Ning, C. Haffner, C. Klochko and M. Mutka, 2006. Pervasive surveillance using a cooperative mobile sensor network. In Proceedings of the IEEE International Conference on Robotics and Automation.

[17] K. Lerman, A. Martinoli and A Galstyan, A Review of Probabilistic Macroscopic Models for Swarm Robotic Systems. 2004.

[18] K. Lerman and A. Galstyan, Two paradigms for the design of artificial collectives. In Proc. of the First Annual workshop on Collectives and Design of Complex Systems, NASA-Ames, CA, 2002.

[19] A. J. Ijspeert, A. Martinoli, A. Billard and L. M. Gambardella, Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The Stick Pulling Experiment. Autonomous Robots 11(2):149-171, 2001.

[20] C. Anderson, Linking Micro- to Macro-level Behavior in the Aggressor-Defender-Stalker Game, in Workshop on the Mathematics and Algorithms of Social Insects (MASI-2003), December, Atlanta, GA, 2003.

[21] A. Martinoli, A. J. Ijspeert and L. M. Gambardella, A probabilistic model for understanding and comparing collective aggregation mechanisms. pp. 575-584. In D. Floreano, J.-D. Nicoud, and F. Mondada, editors, LNAI:1674, Springer, New York, NY, 1999.

[22] W. Agassounon, A. Martinoli and K. Easton, Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes. Special issue on Swarm Robotics, M. Dorigo, and E. Sahin, editors, Autonomous Robots, 17(2-3):163-191, 2004.

[23] N. Tinbergen. Social Behavior in Animal. Chapman and Hall Ltd. Great Britain, 1965.

[24] E. Wilson. The Insect Societies. The Belknap Press. Cambridge, 1971.

[25] A. Portmann. Animals as Social Beings. The Viking Press, New York. 1961.

[26] Tamio Arai, Enrico Pagello, Lynne E. Parker. Advances in Multi-Robot Systems. IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 655-661. October 2002.

[27] B. Yamauchi, "Frontier-based exploration using multiple robots," in Proc. of the second International Conference on Autonomous Agents, Minneapolis, MN, USA, pp. 47-53, 1998.

[28] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors and S. Thrun, and H., Y. (2000), "Coordination for multi-robot exploration and mapping", In Proc. f the National Conference on Artificial Intelligence (AAAI).

[29] I. Rekleitis, G. Dudek and E. Milios, Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. IJCAI Intert. Conference in AI, vol. 2, 1997.

[30] R.M. Zlot, A. Stentz, M.B. Dias and S. Thayer, "Multi-Robot Exploration Controlled By A Market Economy", IEEE International Conference on Robotics and Automation, May, 2002.

[31] Michail G. Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton Kleywegt, Sven Koenig, Craig Tovey, Adam Meyerson and Sonal Jain. Auction-Based Multi-Robot Routing. 2006.

[32] J. Svennebring and S. Koenig. Trail-Laying Robots for Robust Terrain Coverage. In Proceedings of the International Conference on Robotics and Automation, 2003.

[33] S. Koenig, B. Szymanski and Y. Liu. Efficient and Inefficient Ant Coverage Methods. Annals of Mathematics and Artificial Intelligence, 31, 41-76, 2001.

[34] R. Simmons and S. Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In Proceedings of the International Joint Conference on Artificial Intelligence, 1080-1087, 1995.

[35] S. Russel and P. Norvig, "Artificial Intelligence, a modern approach", Prentice Hall, 2nd ed., 2003.

[36] M. Wooldridge, "An introduction to MultiAgent Systems", John Wiley and Sons, LTD, 2002

[37] Jeffrey S. Cox and Edmund H. Durfee, An Efficient Algorithm for Multiagent Plan Coordination. *AAMAS'05*, July 2529, 2005, Utrecht, Netherlands.

[38] Q. Yang, Intelligent Planning. Springer-Verlag, Berlin, 1997.

[39] F. Pecora, R. Rasconi and A. Cesta, Assessing the bias of classical planning strategies on makespan-optimizing scheduling. In Proceedings of the 16th European Conference on Artificial Intelligence, pp. 677-681, 2004.

[40] X. Zheng and S. Koenig. Reaction Functions for Task Allocation to Cooperative Agents. In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2008.

[41] L. E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics", in Distributed Autonomous Robotic Systems., L. E. Parker, G. Bekey and J. Barhen eds., Springer-Verlag Tokyo, pp. 3-12, 2000.

[42] M. S. Fontan and M. J. Mataric. Territorial multi-robot task division. IEEE Transactions of Robotics and Automation, 14(5), 1998.

[43] Building Presence through Localization for Hybrid Telematic Systems, Research and task analysis of telematic planning. Report of Project funded by the European Community under the IST programme: Future and Emerging Technologies, PELOTE, IST-2001-38873, 2003.

[44] J. Melvin, P. Keskinocak, S. Koenig, C. Tovey and B. Yuksel Ozkaya. Multi-Robot Routing with Rewards and Disjoint Time Windows. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), 2332-2337, 2007.

[45] Erfu Yang and Dongbing Gu Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey. Engineering and Physical. 2006.

[46] Ellips Masehian and Davoud Sedighizade, Classic and Heuristic Approaches in Robot Motion Planning - A Chronological Review. Proceedings of World Academy of Science, Engineering And Technology Volume 23 August 2007.

[47] D.L. McGuinness and J. Wright, Conceptual Modeling for Configuration: A Description Logic-based Approach. Artificial Intelligence for Engineering Design, Analysis,and Manufacturing - special issue on Configuration. 1998.

[48] K. P. Sycara, Multiagent Systems. AI Magazine 19(2): 79-92, 1998.

[49] M. Wooldridge, and N. Jennings, IntelligentAgents: Theory and Practice. Knowledge Engineering Review, 10(2): 115-152, 1995.

[50] Gastn Lefranc, Colony of robots: New Challenge, Int. J. of Computers, Communications and Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. III (2008), Suppl. Issue ICCCC 2008, pp. 92-107, 2008.

[51] F. Duvallet, J. Kong, E. Marinelli, K. Woo, A. Buchan, B. Coltin, Ch. Mar and B. Neuman. Developing a Low-Cost Robot Colony. AAAI Fall Symposium: reagrding "Intelligence" in Distributed Intel?ligent Systems, 2007.

[52] Ulrich Nehmzow and Keith Walkery, The Behaviour of a Mobile Robot Is Chaotic, AISB Journal 1(4), c SSAISB, 2003.

[53] W. Sheng, Q. Yang, S. Ci and N. Xi, "Distributed Multi-robot Coordination Algorithm for Area Exploration and Coverage", IEEE International Conference on Robotics and Automation (ICRA) Workshop on The State-of-the-Art of Mobile Robot Area Coverage, 2004.

[54] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen and N. Papanikolopoulos: "Performance of a Distributed Robotic System Using Shared Communications Channels". IEEE Trans. on Robotics and Automation 22(5), 713(727), 2002.

[55] S. Roumeliotis and G. Bekey, Synergetic localization for groups of mobile robots, In Proceedings IEEE Conference on Decision and Control, pp. 3477-3482, Australia, 2000.

[56] Roumeliotis, I. Stergios and Ioannis M. Rekleitis, Analysis of Multirobot Localization Uncertainty Propagation, In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.

[57] R.Vaughan, K. Stoy, G. Sukhatme and M. Mataric, Lost: Localization-space trails for robot teams, IEEE Transactions on Robotics and Automation, 18(5), 796(812), 2002.

[58] W. David, Payton Intelligent real time control of robotic vehicles Communications of the ACM 1991.

[59] R. S. Parpinelli, H. S. Lopes and A. A. Freitas,. An ant colony algorithm for classification rule discovery. In: H. Abbass, R. Sarker and C. Newton, (eds.). Data Mining: a Heuristic Approach, London: Idea Group Publishing, 191-208, 2002.

[60] Ali Umut Irturk, Distributed Multi-robot Coordination For Area Exploration and Mapping. University of California Santa Barbara, 2006.

[61] D. C. McFarlane, S. Bussmann: Developments in Holonic Production Planning and Control. Int. Journal of Production Planning and Control, vol 11, N 6, pp 5522-536, 2000.

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 129-149

Constructing Multi-dimensional Utility Function under Different Types of Preferences over the Fundamental Vector Attributes

Natalia Nikolova, Daniela Toneva, Sevda Ahmed and Kiril Tenekedjiev

Abstract: Consequences in decision problems are often described as multi-dimensional vectors. The direct construction of a utility function over such consequences is difficult and in most cases impossible. The paper proposes an alternative solution introducing a generic seven-step procedure for construction of multi-dimensional utility in the case of mutual utility independence of fundamental vector attributes, and fuzzy rationality (i.e. when preferences of the decision maker only partially obey the rationality axioms). Detailed discussion of each step and the techniques used is proposed. A numerical example of a problem where nine-dimensional consequences are grouped into six fundamental vector attributes is used to illustrate the application of the algorithm.

Keywords: multi-dimensional consequences, utility function, fuzzy rationality, attributes

1 Introduction

The rational choice between risky alternatives is postulated by the paradigms of utility theory [29]. It argues that the rational decision maker (DM) should choose the alternative that best meets her preferences among all other possible alternatives in the problem. Risky alternatives are presented as lotteries, i.e. a full set of disjoint events, associated with consequences (prizes) from a set X according a known probability. Since preferences are measured by the utility function $u(.)$, then rational choice is brought down to calculating the expected utility of each lottery (i.e. weighing utilities of prizes by their probabilities) and choosing the lottery with the highest expected utility.

Authors have arrived at different sets of axioms of rational choice between lotteries [11], which allow to elaborate utility elicitation procedures. They envisage solving a preferential equation between a lottery and/or a prize during a dialog with the DM. The ideal DM obeys the rationality axioms and elicits unique point estimates. The real DM has finite discriminating abilities and elicits an uncertainty interval of estimates. As a result she partially disobeys some of the rationality assumptions and is referred to as fuzzy rational DM (FRDM) [18].

Any choice of action within a problem situation is expected to meet the objectives the DM has imposed, which also predefine the structure of the consequences. In the simplest case, there is only one objective (e.g. in economic decisions that would be maximization of profit) measured by a single attribute (e.g. in economic decisions that would be the net profit), and the preferences of the DM are strictly increasing (monetary prizes are a typical example of that). If the attribute is continuous then X is a one-dimensional bounded interval of prizes. The utility function may be constructed by interpolation/approximation on several elicited nodes. Classical elicitation methods are the probability equivalence

(PE) [3], certainty equivalence (CE) [1], and lottery equivalence (LE) [15]. Modern techniques are the trade-off (TO) [30], and the uncertain equivalence (UE) [27] methods. The approximation is applied if there are only few elicited nodes and/or if their uncertainty intervals are too wide. Here it is important to choose an analytical form depending on the risk attitude of the DM and the type of prizes [13]. In the case of non-monotonic preferences, the elicitation techniques can only be applied once the extrema are identified, and this process is also affected by fuzzy rationality [19].

In more complicated cases, the DM identifies a set of objectives each measured by at least one attribute. Then prizes are presented as multi-dimensional vectors, whose coordinates (attributes) measure the degree of achievement of the objectives. It is rarely possible to directly elicit the utility of multi-dimensional prizes. Therefore it is recommended to combine the attributes into several fundamental vector attributes each containing at least one attribute (and each attribute should belong to exactly one fundamental vector attribute) [13]. Then the multi-dimensional utility function is represented as a combination of the fundamental utility functions and their scaling constants (which measure the significance of each fundamental vector attribute for the preferences of the DM). Independence conditions ensure that such a representation corresponds to the opinion of the DM. The recommended approach is to establish mutual utility independence and depending on whether the sum of the constants equal to one or not construct either an additive or a multiplicative multi-dimensional utility function. The scaling constants are subjectively elicited, thus affected by fuzzy rationality. Their interval form requires special techniques to test whether they sum to one or not. One possible approach is the uniform method, whose essence and (numerical and simulation) realizations are presented in [16], [25].

This paper describes in detail the steps in the construction of a multi-dimensional utility function of a FRDM over prizes represented as sets of fundamental vector attributes. These procedures are structured in a generic algorithm. Initial discussion on that problem is presented in [21]. The latter applies if mutual utility independence is established between fundamental vector attributes. It takes into account the interconnection between the preferences of the attributes within a given fundamental vector attribute, and also the possibility to have different type of preferences over the fundamental vector attributes (monotonic or non-monotonic). All the steps of the algorithm are illustrated in a numerical example, where the utility function over prizes with nine attributes, grouped into six mutually utility independent fundamental vector attributes is constructed.

In what follows, section 2 presents the procedures for direct construction of a multi-dimensional utility function and its disadvantages. Opposed to it is another algorithm of seven steps that decomposes the multi-dimensional utility function to a set of functions over fundamental vector attributes of lower dimension. Description of each step of that algorithm is presented in section 3. Section 4 contains a numerical example that illustrates the algorithm from section 3.

2 Direct construction of multi-dimensional utility function

As a result of the multiple objectives that a DM has in a decision problem, consequences are usually defined as multi-dimensional vectors, whose attributes measure the degree to which objectives are met. In other words, multi-dimensional consequences are represented as d-dimensional vectors with attributes. If the i-th attribute is a random variable X_i with an arbitrary realization x_i, then prizes take the form of d-dimensional vectors $\overrightarrow{\mathbf{x}} = (x_1, x_2, \ldots, x_d)$ in the set of prizes, which is a subset of the d-dimensional Euclidean space. The set of all attributes $X_1, X_2, \ldots, X_d$ may be divided into $n \in 2, 3, \ldots, d$ non-empty non-overlapping subsets $Y_1, Y_2, \ldots, Y_n$. Each Y_j is a system of random variables with an arbitrary realization $\overrightarrow{\mathbf{y}}_j$ and then prizes in X may be presented as $\overrightarrow{\mathbf{x}} = (\overrightarrow{\mathbf{y}}_1, \overrightarrow{\mathbf{y}}_2, \ldots, \overrightarrow{\mathbf{y}}_n)$ [16]. Theoretically speaking, it is possible to construct the multi-dimensional utility function $u(.)$, whose domain are all possible multi-dimensional prizes $\overrightarrow{\mathbf{x}} \in \mathbb{R}^d$:

$$u = u(\overrightarrow{\mathbf{x}}) = u(x_1, x_2, \ldots, x_d). \tag{1}$$

An algorithm for that purpose is proposed in [13].

Algorithm 1. Direct construction of a utility function over d-dimensional prizes

1^0.Choose $\overrightarrow{\mathbf{x}}_{best}$ as $\overrightarrow{\mathbf{x}}_{best} \succsim \overrightarrow{\mathbf{x}}, \overrightarrow{\mathbf{x}} \in \mathbb{R}^d$.

2^0. Choose $\overrightarrow{\mathbf{x}}_{worst}$ as $\overrightarrow{\mathbf{x}} \succsim \overrightarrow{\mathbf{x}}_{worst}, \overrightarrow{\mathbf{x}} \in \mathbb{R}^d$.

3^0. Put $u(\overrightarrow{\mathbf{x}}_{best}) = 1$ and $u(\overrightarrow{\mathbf{x}}_{worst}) = 0$.

4^0. Choose a set M of t number of d-dimensional vectors, scattered in X, containing $\overrightarrow{\mathbf{x}}_{best}$ and $\overrightarrow{\mathbf{x}}_{worst}$: $M = \{\overrightarrow{\mathbf{x}}_1, \overrightarrow{\mathbf{x}}_2, \ldots, \overrightarrow{\mathbf{x}}_n\}$.

5^0. Solve $t-2$ number of preferential equations $< \overrightarrow{\mathbf{x}}_{best}(u_r)\overrightarrow{\mathbf{x}}_{worst} >\sim \overrightarrow{\mathbf{x}}_r$, $r = 1, 2, \ldots, t$; $\overrightarrow{\mathbf{x}}_r \neq \overrightarrow{\mathbf{x}}_{best}$, $\overrightarrow{\mathbf{x}}_r \neq \overrightarrow{\mathbf{x}}_{worst}$.

6^0. Put $u(\overrightarrow{\mathbf{x}}_r) = u_r$, $r = 1, 2, \ldots, t$; $\overrightarrow{\mathbf{x}}_r \neq \overrightarrow{\mathbf{x}}_{best}$, $\overrightarrow{\mathbf{x}}_r \neq \overrightarrow{\mathbf{x}}_{worst}$.

7^0. Construct $u(\overrightarrow{\mathbf{x}})$ by d-dimensional interpolation/approximation on the elicited utilities of the elements in the set M.

Here, $\overrightarrow{\mathbf{x}}_{best}$ is a consequence, whose attributes are set to their most preferred level, $\overrightarrow{\mathbf{x}}_{worst}$ is a consequence, whose attributes are set to their least preferred level, whereas $\overrightarrow{\mathbf{x}}_r$ is a consequence whose r-th attribute is set to its most preferred level all other being set to their worst ones. The sign $\succsim$ stands for the binary relation “at least as preferred as”, whereas $\sim$ stands for the binary relation “equally preferred to”, both of which defined over objects. The notation $< \overrightarrow{\mathbf{x}}_{best}(u_r)\overrightarrow{\mathbf{x}}_{worst} >\sim \overrightarrow{\mathbf{x}}_r$ stands for a preferential equation between the lottery $< \overrightarrow{\mathbf{x}}_{best}(u_r)\overrightarrow{\mathbf{x}}_{worst} >$ between $\overrightarrow{\mathbf{x}}_{best}$ and $\overrightarrow{\mathbf{x}}_{worst}$ with probabilities respectively u_r and (1-u_r), and the prize $\overrightarrow{\mathbf{x}}_r$. The DM identifies the value u_r in an iterative procedure, and at the moment of indifference, $u(\overrightarrow{\mathbf{x}}_r) = u_r$. A possible alternative solution of the task in step 7 is to use multi-dimensional spline [4]. Another possibility is to formulate an analytical form of the utility function, which includes several unknown parameters, but also uses prior information for the preferences of the DM. The unknown parameters may be estimated using the least square method with singular value decomposition [23].

The authors of the algorithm argue in [13] that it should be used in the last resort, since it is practically impossible to realize when $d > 3$. A much better approach is to decompose (1) to fundamental utility functions over the fundamental vector attributes $Y_1, Y_2, \ldots, Y_n$:

$$u(\overrightarrow{\mathbf{x}}) = u(\overrightarrow{\mathbf{y}}_1, \overrightarrow{\mathbf{y}}_2, \ldots, \overrightarrow{\mathbf{y}}_n) = f[u_{y,1}(\overrightarrow{\mathbf{y}}_1), u_{y,2}(\overrightarrow{\mathbf{y}}_2), \ldots, u_{y,n}(\overrightarrow{\mathbf{y}}_n)]. \tag{2}$$

Here, $f(.)$ is a real-valued function of n real variables. The adequacy of (2) requires that certain independence conditions of preferences over the attributes hold − preferential, utility and additive independence. Preferential independence is most common and the weakest independence condition. The strongest condition is additive independence, but also very hard to establish, and thus difficult to use in practice. Most important from a practical point of view is utility independence.

Let's divide the set of all fundamental vector attributes $\{Y_1, Y_2, \dots, Y_n\}$ into two non-empty non-overlapping subsets Z and $\overline{Z}$, where $\overline{Z}$ is complementary to Z. Since Z and $\overline{Z}$ are random vectors (with dimensionality coinciding with the sum of the dimensions of the consisting fundamental vector attributes), then these along with their realizations may be analyzed as vector attributes of a given prize: $\overrightarrow{\mathbf{x}} = (\overrightarrow{\mathbf{z}}, \overrightarrow{\overline{\mathbf{z}}})$. Let $l_{i,\overrightarrow{\overline{\mathbf{z}}}}$ be a lottery with prizes, for which $\overline{Z} = \overrightarrow{\overline{\mathbf{z}}}$. The vector attribute Z is utility independent of $\overline{Z}$, if for all $\overrightarrow{\overline{\mathbf{z}}} \in \overline{Z}$ the preference order of lotteries, whose prizes involve only changes in the levels in Z does not depend on the levels at which $\overline{Z}$ is held fixed:

$$l_{i,\overrightarrow{\overline{\mathbf{z}}}_k} \succsim l_{j,\overrightarrow{\overline{\mathbf{z}}}_k} \Rightarrow l_{i,\overrightarrow{\overline{\mathbf{z}}}} \succsim l_{j,\overrightarrow{\overline{\mathbf{z}}}} \tag{3}$$

If this holds, then Z is called UI (*utility independent*). The fundamental vector attributes $Y_1, Y_2, \dots, Y_n$ are mutually utility independent if each vector attribute Z defined on the basis of the fundamental vector attributes $Y_1, Y_2, \dots, Y_n$ is UI [2]. The most preferred value of the fundamental vector attribute Y_j for the DM is $(\overrightarrow{\mathbf{y}}_j)_{best}$, whereas the least preferred value is $(\overrightarrow{\mathbf{y}}_j)_{worst}$.

The mutual utility independence allows to represent the d-dimensional utility function $u(.)$ as a polynomial of n utility functions $u_{y,j}(.)$ over the fundamental vector attributes [6]:

$$\begin{aligned}
u &= u(\overrightarrow{\mathbf{x}}) = u(\overrightarrow{\mathbf{y}}_1,\ \overrightarrow{\mathbf{y}}_2,\ \dots,\ \overrightarrow{\mathbf{y}}_n) = \\
&= \sum_{j=1}^{n} k_{y,j} u_{y,j}(\overrightarrow{\mathbf{y}}_j) + K_y \sum_{j=1}^{n-1} \sum_{s=j+1}^{n} k_{y,j} k_{y,s} u_{y,j}(\overrightarrow{\mathbf{y}}_j) u_{y,s}(\overrightarrow{\mathbf{y}}_s) + \\
&+ K_y^2 \sum_{j=1}^{n-2} \sum_{s=j+1}^{n-1} \sum_{r=s+1}^{n} k_{y,j} k_{y,s} k_{y,r} u_{y,j}(\overrightarrow{\mathbf{y}}_j) u_{y,s}(\overrightarrow{\mathbf{y}}_s) u_{y,r}(\overrightarrow{\mathbf{y}}_r) + \\
&+ \dots + K_y^{n-1} k_{y,1} k_{y,2} \dots k_{y,n} u_{y,1}(\overrightarrow{\mathbf{y}}_1) u_{y,2}(\overrightarrow{\mathbf{y}}_2) \dots u_{y,n}(\overrightarrow{\mathbf{y}}_n).
\end{aligned} \tag{4}$$

Here, $u_{y,j}(.)$ are d_j-dimensional bounded utility functions defined over all possible values $\overrightarrow{\mathbf{y}}_j$ of Y_j, $k_{y,j} \in [0; 1]$ are scaling constants that indicate the relative significance of each fundamental vector attribute for the preferences of the DM over the multi-dimensional prizes, and K_y is a general constant that depends on the values of the scaling constants $k_{y,j}$. The polynomial form (4) is traditionally called *multi-linear* [8]. If $d_j > 1$ then each function $u_{y,j}(.)$ has to be additionally decomposed if there are conditionally utility independent attributes among those it is defined on [5]. Otherwise $u_{y,j}(.)$ has to be constructed according to algorithm 1. The functions $u(.)$ and $u_{y,j}(.)$ are normalized so that

$$u_{y,j}(\overrightarrow{\mathbf{y}}_{j,best}) = 1,\ j = 1,\ 2,\ \dots,\ n, \tag{5}$$

$$u_{y,j}(\overrightarrow{\mathbf{y}}_{j,worst}) = 0,\ j = 1,\ 2,\ \ldots,\ n, \tag{6}$$

$$u(\overrightarrow{\mathbf{x}}_{best}) = u(\overrightarrow{\mathbf{y}}_{1,best},\ \overrightarrow{\mathbf{y}}_{2,best}, \ldots,\ \overrightarrow{\mathbf{y}}_{n,best}) = 1, \tag{7}$$

$$u(\overrightarrow{\mathbf{x}}_{worst}) = u(\overrightarrow{\mathbf{y}}_{1,worst},\ \overrightarrow{\mathbf{y}}_{2,worst}, \ldots,\ \overrightarrow{\mathbf{y}}_{n,worst}) = 0, \tag{8}$$

The scaling constants in (4) correspond to the utility of the so called corner consequences $\overrightarrow{\mathbf{x}}_{j,corner} = [(\overrightarrow{\mathbf{y}}_1)_{worst},\ (\overrightarrow{\mathbf{y}}_2)_{worst}, \ldots,\ (\overrightarrow{\mathbf{y}}_j)_{best}, \ldots,\ (\overrightarrow{\mathbf{y}}_n)_{worst}]$, whose fundamental vector attributes are set to their worst levels except for the j-th vector attribute, which is set to its best level:

$$k_{y,j} = u[(\overrightarrow{\mathbf{y}}_1)_{worst},\ (\overrightarrow{\mathbf{y}}_2)_{worst}, \ldots,\ (\overrightarrow{\mathbf{y}}_j)_{best}, \ldots,\ (\overrightarrow{\mathbf{y}}_n)_{worst})], \tag{9}$$

$$1 + K_y = \prod_{j=1}^{n} (K_y \times k_{y,j} + 1). \tag{10}$$

2.1 Generic algorithm for the construction of a multi-dimensional utility

As an alternative to algorithm 1, it is possible to use another sequence of steps, which could help construct a multi-dimensional utility function of a FRDM.

Algorithm 2. Constructing a multi-dimensional utility function of a FRDM

1^0. Establish mutual utility independence between the fundamental vector attributes $Y_1,\ Y_2,\ \ldots,\ Y_n$.

2^0. Decompose each function $u_{y,j}(.)$ for $d_j > 1$, if there are conditionally utility independent attributes among those it is constructed on [5].

3^0 Construct all non-decomposed $u_{y,j}(.)$ for $d_j > 1$ and all multi-dimensional decomposed parts of $u_{y,j}(.)$ using Algorithm 1.

4^0. Construct all one-dimensional utilities and one-dimensional decomposed parts of $u_{y,j}(.)$:

$a)^0$ in the case of non-monotonic preferences: define the number of local extrema and divide the one-dimensional set of prizes into sectors with pseudo-unimodal preferences; elicit the uncertainty interval of the extremum in each sector using the procedures from [19] depending on the type of preferences; define all sections with strictly monotonic preferences between the extremum platforms;

$b)^0$ in the case of monotonic preferences, the entire one-dimensional set of prizes is one section;

$c)^0$ elicit z number of nodes of the local utility function in each section using PE, CE, LE, UE, etc.; arctg-approximate the utility function following the procedures from [28], and if it is of poor quality - replace it by linear interpolation;

$d)^0$ in the case of non-monotonic preferences: construct the global by utility function using the algorithms from [19] several times; in the case of monotonic preferences the local utility function coincides with the global one.

5^0. Elicit the uncertainty intervals of the scaling constants.

6^0. Use the numerical [25] or the simulation [16] realization of the uniform method to analyze the sum of the scaling constants and to find their point estimates.

7^0. Construct the multi-dimensional utility function in a form defined by (4).

What follows is a more detailed description of the techniques and methods necessary for the execution of each step of Algorithm 2.

1^0. Establish mutual utility independence between the fundamental vector attributes

At this step it is necessary to establish mutual utility independence between a set of fundamental vector attributes according to (3).

2^0. Decompose the fundamental utility functions

This step is executed provided there are attributes in a fundamental vector attribute that are conditionally utility independent of the other attributes in this subgroup. At a later stage that allows to construct the utility function on the decomposed parts, where prizes are of lower dimension thus easier to work with.

3^0. Construct the non-decomposed and the multi-dimensional decomposed utility functions using Algorithm 1

Although it has its limitations, outlined by its authors, algorithm 1 is the only option to the problem of multi-dimensional utilities when decomposition of the initial multi-dimensional prizes is not possible, or when decomposition of the prizes still leads to the identification of multi-dimensional prizes (though of lower dimension than the initial ones)

4^0. Construct all one-dimensional utilities and one-dimensional decomposed parts of $u_{y,j}(.)$

A classical task in decision analysis is to build a utility function over a one-dimensional set X. The function is constructed in the interval $[x_{worst}; x_{best}]$, where $x_{best} = sup(X)$, $x_{worst} = inf(X)$. The procedures depend on whether preferences of the FRDM are monotonic or non-monotonic.

The preferences of the FRDM are usually monotonically increasing, i.e.

$$x_i \succ x_j \Longleftrightarrow x_i > x_j,\ x_i \in X,\ x_j \in X. \tag{11}$$

It is reasonable to elicit only several (z) nodes and construct the utility function using approximation/interpolation. The utility elicitation techniques solve preferential equations by changing one parameter until compared options (prizes and/or prizes) become indifferent [7]. Assume that a FRDM has elicited $z-2$ inner nodes of $u(.)$ with coordinates $(x_{u_l};\, u_l)$, $l = 2, 3, \dots, z-1$ (x_{u_l} and u_l are respectively an utility quantile and an utility quantile index). The end nodes are known: $(x_{u_1};\, u_1) = (x_{worst};\, 0)$, $(x_{u_z};\, u_z) = (x_{best}; 1)$. Methods like PE or LE select several quantiles ($l = 2, 3, \dots, z-1$) and elicit the corresponding $\hat{u}_l$. The FRDM elicits uncertainty intervals of the form $\hat{u}_l \in [\hat{u}_l^d;\, \hat{u}_l^u](l = 2, 3, \dots, z-1)$, where $\hat{u}_l^d$ and $\hat{u}_l^u$ are the lower and upper bounds of the uncertainty interval of $\hat{u}_l$. Other methods, like CE or UE select several quantile indices u_l ($l = 2, 3, \dots, z-1$) and elicit their corresponding $\hat{x}_{u_l}$. The FRDM elicits uncertainty intervals of the form $\hat{x}_{u_l} \in [\hat{x}^d_{u_l}; \hat{x}^u_{u_l}]$ $(l = 2, 3, \ldots, z-1)$, where $\hat{x}^d_{u_l}$ and $\hat{x}^u_{u_l}$ are the lower and upper bound of the uncertainty interval of $\hat{x}_{u_l}$. Similar considerations apply for the case of monotonically decreasing preferences.

Once elicited nodes are available, the utility function may be constructed using linear interpolation or analytical approximation. The method should precisely interpret the data, and should preserve the risk attitude of the DM [10], modeled by the local risk aversion function $r(x) = -u''(x)/u'(x)$ [22]. The work [28] discusses several methods for analytical approximation of a monotonically increasing 1-D $u(.)$. Analytical approximation is appropriate if few elicited nodes are available and/or the uncertainty intervals are too wide. A rich source of analytical forms is [13]. The work [24] proposed the Harrington's desirability function, whereas [17] introduces an arctg-approximation that applies over set of monetary gains and loses, and incorporates proper prior information for the risk attitude. The parameters a (measuring risk sensitivity) and x_0 (the inflex point of the function) of that form are estimated using a weighted least square method, and the goodness-of-fit is analyzed by a χ^2 measure. The advantage of that form over other analytical forms is justified in [17], [28].

A more complex situation arises when the DM has quasi-unimodal preferences, i.e. when there is a value x_{opt} with extreme utility (either a minimum or a maximum) within the interval of prizes [19]. There are two types of quasi-unimodal preferences - hill (with a maximum extremum) and valley (with a minimum extremum) preferences. Both occur due to two contradicting factors related to the analyzed variable. Difficulties arise when the DM has to compare values on both sides of the extreme interval $-$ if a sufficient difference in utilities exists, the DM would be able to state preference otherwise she would be indifferent being unable to compare the options. This leads to mutual non-transitivity of preferences and even a very motivated and rational DM would express fuzzy, rather than classical rationality. As a result, she would identify an extreme interval rather than a single value x_{opt}. Since all elicitation techniques need the reference points (the prizes with extreme utility), identifying the extreme interval is mandatory.

The models of hill and valley utility functions are based on two separate sets of assumptions that refer to the discriminating abilities of the FRDM and the characteristics of the extreme interval. Two 20-step algorithms are elaborated (one per each type of quasi-unimodal preferences), which find the extreme interval via a dialog with the FRDM. Both algorithms combine the golden section search [14] and bisection [23] methods. Golden section serves to locate the extreme interval, whereas bisection estimates its lower and upper bounds. As argued in [19] in an algebraic case of one-dimensional optimization, the Kiefer-Johnson method [9] is probably the best, followed by golden section search and bisection. If preferential equations are solved, then the number of comparisons is what matters. That is why the Kiefer-Johnson approach is inapplicable, whereas bisection is more effective than golden section search in terms of reduction of the interval after each comparison. However it requires comparison of close prizes that is likely to generate biased estimates. That is why although less effective, the golden section search is more appropriate as it generates less biased results. After the extreme interval is identified, the next step is to construct local utility functions in the sections with monotonic preferences (the sections on each side of the extremum) using the techniques discussed earlier in this section. Finally, it is possible to construct the global utility function over the entire set of prizes by rescaling the local functions. Two other algorithms are proposed for that purpose in [19]. The discussion and algorithms in that work focus on the case when there is a single extremum within the prize interval. It is though possible that the DM may identify several extrema of each type. Then the elaborated algorithms would have to be applied several times first to elicit all extreme intervals, then construct the monotonic utilities in the resulting sections, and rescale them. The latter would require eliciting the global utilities of the extreme prizes that are different from the best and the worst ones and rescale each monotonic function accordingly, for example presuming that linear dependence should exist between the initial and the rescaled utilities.

5^0. Elicit the uncertainty intervals of the scaling constants

Scaling constants are defined by (9) and are the utilities of the corner consequences. The work [16] suggests that a scaling constant should be elicited by solving the preferential equation $< 0.5,\ \overrightarrow{\mathbf{x}}_{j,corner};\ 0.5,\ \overrightarrow{\mathbf{x}}_{worst} > \sim < \overrightarrow{\mathbf{x}}_{best}(p)\overrightarrow{\mathbf{x}}_{worst} >$. Here, $< \overrightarrow{\mathbf{x}}_{best}(p)\overrightarrow{\mathbf{x}}_{worst} >$ is a reference lottery that gives $\overrightarrow{\mathbf{x}}_{best}$ and $\overrightarrow{\mathbf{x}}_{worst}$ with probabilities p and $(1-p)$, whereas $< 0.5, \overrightarrow{\mathbf{x}}_{j,corner}; 0.5, \overrightarrow{\mathbf{x}}_{worst} >$ is a simple ordinary lottery that gives $\overrightarrow{\mathbf{x}}_{j,corner}$ and $\overrightarrow{\mathbf{x}}_{worst}$ with equal chances. The reference lottery may be visualized as an urn of N balls, of which $M = p \times N$ are white, and the rest are black. The prizes $\overrightarrow{\mathbf{x}}_{best}$ or $\overrightarrow{\mathbf{x}}_{worst}$ may be received if a white or a black ball is drawn from the urn. Then $< \overrightarrow{\mathbf{x}}_{best}(p)\overrightarrow{\mathbf{x}}_{worst} >$ transforms into $< \overrightarrow{\mathbf{x}}_{best}(M/N)\overrightarrow{\mathbf{x}}_{worst} >$. As the DM is only fuzzy rational, she should identify the greatest $M = \hat{M}_{down}$, such that $< 0.5, \overrightarrow{\mathbf{x}}_{j,corner}; 0.5, \overrightarrow{\mathbf{x}}_{worst} > \succ < \overrightarrow{\mathbf{x}}_{best}(\hat{M}_{down}/N)\overrightarrow{\mathbf{x}}_{worst} >$, and the smallest $M = \hat{M}_{up}$, such that $< \overrightarrow{\mathbf{x}}_{best}(\hat{M}_{up}/N)\overrightarrow{\mathbf{x}}_{worst} > \succ < 0.5, \overrightarrow{\mathbf{x}}_{j,corner}; 0.5, \overrightarrow{\mathbf{x}}_{worst} >$ ($\succ$ stands for the binary relation "strict preference"). Then $M^* \in (\hat{M}_{down};\ \hat{M}_{up})$, and $k_{y,j} \in (2 \times \hat{M}_{down}/N;\ 2 \times \hat{M}_{up}/N)$, which is the uncertainty interval of $k_{y,j}$. The latter may be elicited using triple bisection [26]. That is why the opinion of the DM regarding the scaling constant $k_{y,j}$ takes the form

$$k_{y,j} \in [\hat{k}^d_{y,j};\ \hat{k}^u_{y,j}], \text{for} j = 1,\ 2, \ldots,\ n. \tag{12}$$

The interval in (12) is closed to include the cases when constants are known.

6^0. Use the uniform method to analyze the scaling constants sum

It is possible to know for sure the values of some constants, but practically speaking, those are always assessed subjectively as uncertainty intervals, as demonstrated earlier. Then a question arises of how to find the sum of the constants, which cannot be answered in a straightforward way before point estimates were identified. Let the constants be renumbered in descending order of the length of their uncertainty intervals. The following conditions hold:

$$\begin{aligned} k_{y,j} \in [k^d_{y,j};\ k^u_{y,j}], \text{for} j = 1,\ 2, \ldots,\ n, \\ k^u_{y,j} - k^d_{y,j} \geq k^u_{y,j+1} - k^d_{y,j+1}, \text{for} j = 1,\ 2, \ldots,\ n-1, \\ 0 < k^d_{y,j} < k^u_{y,j} < 1, \text{for} j = 1,\ 2, \ldots,\ m, \\ k^d_{y,j} = k^u_{y,j}, \text{for} j = m+1,\ m+2, \ldots,\ n. \\ 0 \leq m \leq n. \end{aligned} \tag{13}$$

The construction of the utility function over the multi-dimensional consequences with

n number of fundamental vector attributes requires to find whether $k_{y,1}+k_{y,2}+\cdots+k_{y,n}=1$, and then find point estimates of the constants $\hat{k}_{y,j}$, for $j = 1, 2, \ldots, n$.

The uniform method has been proposed to solve the scaling constants' problem [16], [25]. Assume that $a_n = \sum_{j=1}^{n} k^d_{y,j}$, $b_n = \sum_{j=1}^{n} k^u_{y,j}$, $y_n = \sum_{j=1}^{n} k_{y,j}, (j = 1, 2, \ldots, n)$, $s = \frac{a_n+b_n}{2}$ and that each unknown constant is uniformly distributed in its uncertainty interval and has a density

$$f_{k_{y,j}}(k_{y,j}) = \begin{cases} 0 & \text{for } k_{y,j} < k^d_{y,j} \\ \frac{1}{k^u_{y,j}-k^d_{y,j}} & \text{for } k^d_{y,j} \le k_{y,j} \le k^u_{y,j} \\ 0 & \text{for } k^u_{y,j} < k_{y,j} \end{cases} \quad \text{for } j = 1, 2, \ldots, m; \tag{14}$$

If $m > 0$, $a_n < 1$, $b_n > 1$, then it is necessary to find the distribution law of the random variable y_n.

An analytical procedure to construct the density $f_{y_n}(.)$ of the sum of scaling constants is elaborated in [25], as well as its numerical approximation. A Monte-Carlo based simulation approximation is presented in [16], which finds the distribution law in the form of a cumulative distribution function $F^n_{y_n}(.)$.

The sum of the scaling constants depends on the sum of their lower and upper bounds, and cannot be deduced if $a_n < 1$, $b_n > 1$. For that non-trivial case, the work [16] proposes to use a two-tail statistical test with a null hypothesis "the sum of the scaling constants equals to one" ($H_0 : y_n = 1$) and alternative hypothesis "the sum of the scaling constants is not equal to one" ($H_1 : y_n \neq 1$). At a level of significance α and calculated probability p_{value} to reject H_0 that is true (i.e. the type I error of the test), H_0 is rejected and H_1 is accepted if $p_{value} \le \alpha$, or H_0 fails to be rejected if $p_{value} > \alpha$, as suggested in [12]. It is necessary to assess p_{value}. An appendix in [16] proves an analytical dependence of p_{value} and the density $f_{y_n}(.)$:

$$\hat{p}_{value} = \begin{cases} 2\int_{a_n}^{1} \hat{f}_{y_n}(y)dy & \text{if } s > 1; \\ 2 - 2\int_{a_n}^{1} \hat{f}_{y_n}(y)dy & \text{if } s \le 1. \end{cases} \tag{15}$$

which is numerically approximated in [25], whereas a simulation approximation in [16] proves that

$$\hat{p}_{value} = \begin{cases} 2\hat{F}^n_{y_n}(1) & \text{if } s > 0.5; \\ 2 - 2\hat{F}^n_{y_n}(1) & \text{if } s \le 0.5. \end{cases} \tag{16}$$

Point estimates of the constants are defined depending on the result of the test, according to the dependence given below, where $\beta = (b_n - 1)/(b_n - a_n)$:

$$\hat{k}_{y,j} = \begin{cases} \beta k^d_{y,j} + (1-\beta)k^u_{y,j} & \text{if } H_0 \text{ fails to be rejected;} \\ \frac{(k^d_{y,j}+k^u_{y,j})}{2} & \text{if } H_1 \text{ is accepted.} \end{cases} \tag{17}$$

The sum of $k_{y,j}$ defines the form of the utility function, as follows:

$$u(\overrightarrow{\mathbf{x}}) = \begin{cases} \sum_{i=1}^{n} k_{y,j} u_{y,j}(\overrightarrow{\mathbf{y}}_j) & \text{if } \sum_{j=1}^{n} k_{y,j} = 1; \\ \frac{1}{K_y} \times \prod_{j=1}^{n} [K_y k_{y,j} u_{y,j}(\overrightarrow{\mathbf{y}}_j) + 1] - 1 & \text{if } \sum_{j=1}^{n} k_{y,j} \neq 1. \end{cases} \tag{18}$$

3 Numerical example

Let's consider a numerical example, which demonstrates the application of the procedures discussed in the previous sections. A decision problem requires to rank alternatives, whose consequences are multi-dimensional vectors with $d = 9$ attributes $\overrightarrow{\mathbf{x}} = (x_1, x_2, \ldots, x_9)$. It is necessary to construct the utility function over the multi-dimensional consequences following Algorithm 2. For the purpose of the utility analysis, the FRDM has established mutual utility independence between $n = 6$ groups of fundamental vector attributes: $\overrightarrow{\mathbf{y}}_1 = (x_1, x_2, x_3)$, $\overrightarrow{\mathbf{y}}_2 = (x_4, x_5)$, $\overrightarrow{\mathbf{y}}_3 = (x_6)$, $\overrightarrow{\mathbf{y}}_4 = (x_7)$, $\overrightarrow{\mathbf{y}}_5 = (x_8)$, $\overrightarrow{\mathbf{y}}_6 = (x_9)$.

4 Constructing the fundamental utility function $u_{y,1}(\vec{y}_1)$

The fundamental vector attribute $\overrightarrow{\mathbf{y}}_1$ consists of three attributes – x_1, x_2, x_3. The attribute x_1 is a continuous random variable in the interval [1; 7], x_2 is a continuous random variable in the interval [30; 110], whereas x_3 is a continuous random variable in the interval [6; 20]. Four values for each attribute are investigated, as follows: $x_{1,1} = 1, x_{1,2} = 3, x_{1,3} = 5, x_{1,4} = 7, x_{2,1} = 30, x_{2,2} = 60, x_{2,3} = 90, x_{2,4} = 110, x_{3,1} = 6, x_{3,2} = 12, x_{3,3} = 18$, and $x_{3,4} = 20$. A total of 64 values of $u_{y,1}(\overrightarrow{\mathbf{y}}_1)$ one per each possible combination of x_1, x_2, and x_3 are elicited. The results (point estimates) are given in Table 1. The corresponding utility functions are also depicted on Fig. 1. All other utility values of $u_{y,1}(\overrightarrow{\mathbf{y}}_1)$, other than the investigated ones, may be found using 3-D linear interpolation using the data from Table 1.

Table 1: Utility values of $u_{y,1}(\vec{\mathbf{y}}_1)$ for 64 combinations of x_1, x_2, and x_3

	Utility at $x_{3,1} = 6$					Utility at $x_{3,2} = 12$			
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$		$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$
$x_{1,1}$	0	0.10	0.30	0.21	$x_{1,1}$	0.15	0.22	0.40	0.18
$x_{1,2}$	0.10	0.20	0.42	0.18	$x_{1,2}$	0.18	0.48	0.50	0.23
$x_{1,3}$	0.28	0.40	0.68	0.23	$x_{1,3}$	0.20	0.41	0.45	0.27
$x_{1,4}$	0.15	0.23	0.35	0.18	$x_{1,4}$	0.17	0.35	0.37	0.30
	Utility at $x_{3,3} = 18$					Utility at $x_{3,4} = 20$			
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$		$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$
$x_{1,1}$	0.58	0.60	0.62	0.57	$x_{1,1}$	0.40	0.48	0.51	0.46
$x_{1,2}$	0.63	0.88	0.90	0.68	$x_{1,2}$	0.45	0.42	0.60	0.55
$x_{1,3}$	0.67	0.96	1.00	0.70	$x_{1,3}$	0.55	0.46	0.68	0.60
$x_{1,4}$	0.58	0.89	0.91	0.59	$x_{1,4}$	0.52	0.59	0.61	0.58

4.1 Constructing the fundamental utility function $u_{y,2}(\vec{\mathbf{y}}_2)$

The fundamental vector attribute $\overrightarrow{\mathbf{y}}_2$ consists of two attributes – x_4 and x_5. The utility of the first attribute is not dependent on the values of x_5, whereas the opposite is not true. That is why, $u_{y,2}(\overrightarrow{\mathbf{y}}_2) = u_4(x_4) \times u_5(x_5|x_4)$. Thus it is necessary to construct $u_4(x_4)$, as

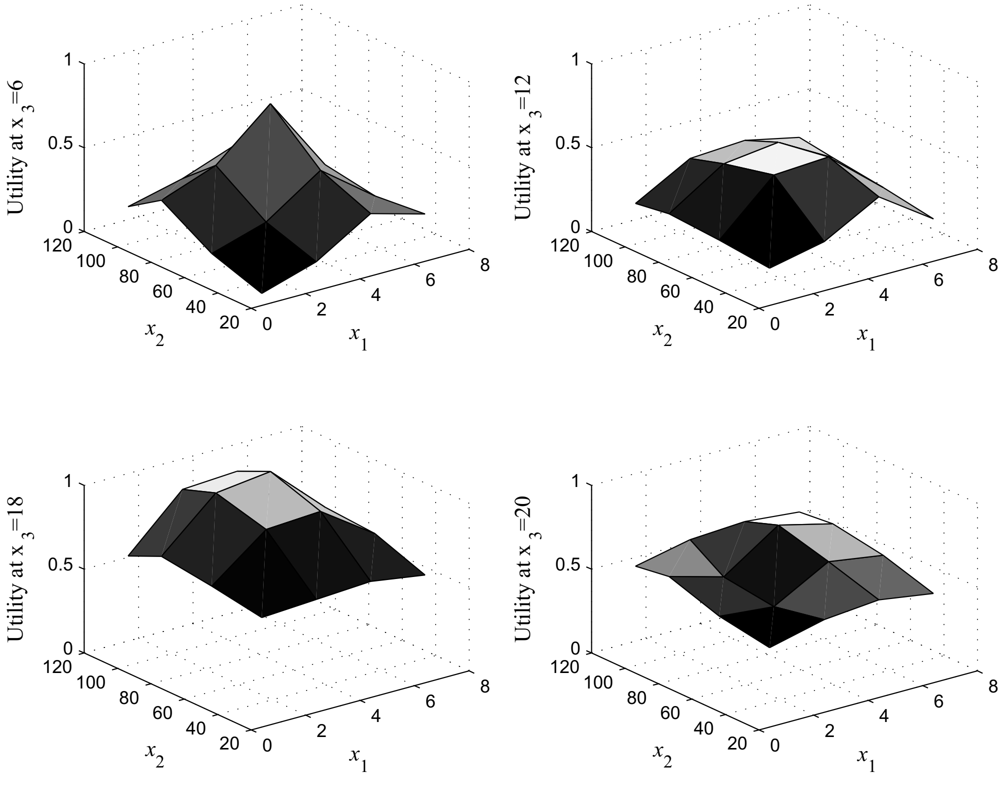


Figure 1: Visualization of the three-dimensional utility function over x_1 and x_2 at the four analyzed value of x_3

well as $u_5(x_5|x_4)$ at several values of x_4, and then interpolate on the resulting functions in order to assess $u_{y,2}(\overrightarrow{\mathbf{y}}_2)$.

The attribute x_4 is a continuous random variable in the interval $[-2; 12]$. The preferences of the FRDM over the values of x_4 are independent on the values of x_5 and are strictly increasing. It is necessary to construct the one-dimensional utility function in that interval. LE is used to elicit 3 inner quantile indices of the quantiles $x_{4,u_2} = 1, x_{4,u_3} = 5, x_{4,u_4} = 9$. The acquired results are: $\hat{u}_{4,2} \in [\hat{u}^d_{4,2}; \hat{u}^u_{4,2}] \equiv [0.06; 0.12]$, $\hat{u}_{4,3} \in [\hat{u}^d_{4,3}; \hat{u}^u_{4,3}] \equiv [0.35; 0.47]$, $\hat{u}_{4,4} \in [\hat{u}^d_{4,4}; \hat{u}^u_{4,4}] \equiv [0.73; 0.82]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt} = 0.1988, x_{0,opt} = 6.8669, \chi^2 = 0.1442$. The resulting utility function is depicted on Fig. 2a. The utilities of the values of x_4 in the interval $[-2; 12]$, other than the investigated ones may be found using linear interpolation on the elicited nodes taking into account the direction of increase of the function.

The attribute x_5 is a continuous random variable in the interval $[-5; 35]$. The preferences of the FRDM over the values of x_5 are dependent on the values of x_4 and again are strictly increasing. Thus it is necessary to construct five utility functions of x_5 one per each investigated value of $x_4(x_{4,u_1} = -2, x_{4,u_2} = 1, x_{4,u_3} = 5, x_{4,u_4} = 9, x_{4,u_5} = 12)$. LE is used to elicit 3 inner quantile indices of the quantiles $x_{5,u_2} = 1, x_{5,u_3} = 5, x_{5,u_4} = 9$ from each of the five functions. The elicitation results, as well as the optimal parameters of the arctg-approximated utility functions are summarized in Table 2. The resulting utility functions are depicted on Fig. 2b - 2f.

Table 2: Elicited nodes and parameters of the arctg-approximated utility functions $u_5(x_5|x_4)$ at five possible values of x_5. The bolded values are those not elicited

		x_5					a_{opt}	$x_{0,opt}$	χ^2_{opt}
		-5	5	15	25	35			
$u_5(x_5\|x_{4,u_1}=-2)$	lower bound	**0**	0.05	0.12	0.35	**1**	64.4018	47.1146	0.0218
	upper bound	**0**	0.11	0.25	0.46	**1**			
$u_5(x_5\|x_{4,u_2}=1)$	lower bound	**0**	0.07	0.16	0.75	**1**	0.1421	20.1046	0.4783
	upper bound	**0**	0.13	0.29	0.85	**1**			
$u_5(x_5\|x_{4,u_3}=5)$	lower bound	**0**	0.30	0.79	0.88	**1**	0.1488	6.5533	0.3347
	upper bound	**0**	0.38	0.90	0.95	**1**			
$u_5(x_5\|x_{4,u_4}=9)$	lower bound	**0**	0.40	0.88	0.92	**1**	0.2611	5.1572	0.2260
	upper bound	**0**	0.48	0.97	0.98	**1**			
$u_5(x_5\|x_{4,u_5}=12)$	lower bound	**0**	0.73	0.86	0.92	**1**	98.5003	−9.3281	0.0429
	upper bound	**0**	0.82	0.98	0.99	**1**			

The utility of any arbitrary combination of x_5 and x_4, other than the investigated ones, can be found using bilinear interpolation [23]. For that purpose, a matrix of utilities at the investigated values of x_5 and x_4 is created (see Table 3). The quantities in it are generated from the corresponding arctg-approximated utility functions constructed earlier (see Table 2) using the original MATLAB program function *universal_utility*, which finds the utility values that correspond to given prizes using the universal increasing arctg-utility function.

Table 3: Matrix for bilinear interpolation of $u_5(x_5|x_4)$

	x_4				
x_5	-2	1	5	9	12
-5	**0**	**0**	**0**	**0**	**0**
5	0.0719	0.0671	0.3428	0.4406	0.7734
15	0.1886	0.2761	0.8153	0.9083	0.9110
25	0.4109	0.7850	0.9507	0.9762	0.9684
35	**1**	**1**	**1**	**1**	**1**

4.2 Constructing the fundamental utility function $u_{y,3}(\vec{\mathbf{y}}_3)$

The fundamental vector attribute $\vec{\mathbf{y}}_3$ consists of the attribute x_6, which is a continuous random variable in the interval [1000; 7000]. Therefore, $u_{y,3}(\vec{\mathbf{y}}_3) \equiv u_6(x_6)$. The preferences of the FRDM are strictly increasing over the values of x_6. It is necessary to construct the one-dimensional utility function in that interval. LE is used to elicit 5 inner quantile indices of the quantiles $x_{6,u_2} = 2000, x_{6,u_3} = 3000, x_{6,u_4} = 4000, x_{6,u_5} = 5000$,

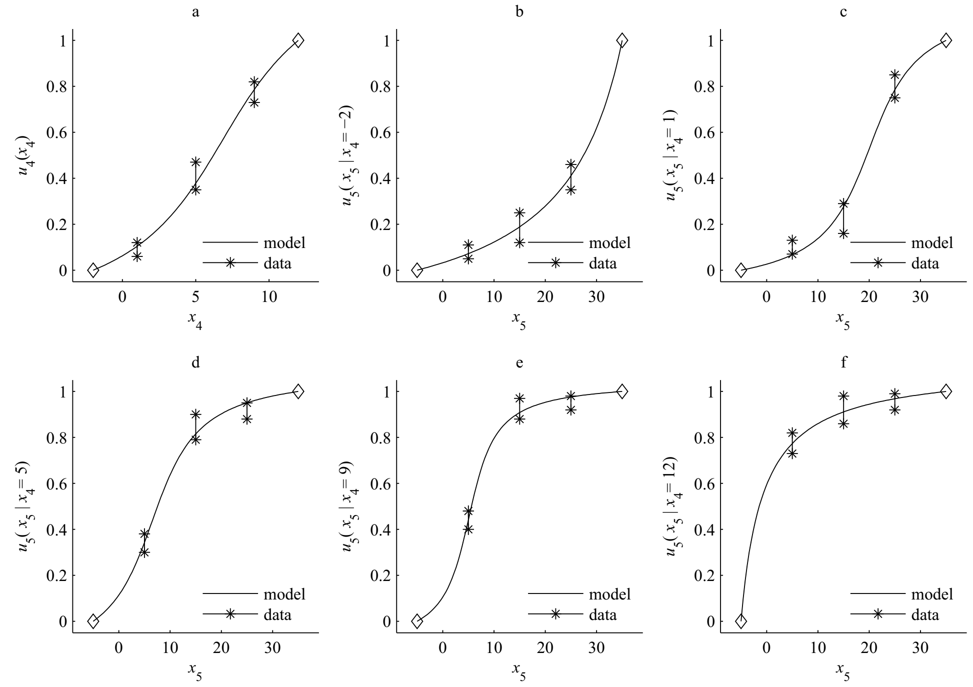


Figure 2: Graphics of the utility function $u_4(x_4)$, and the utility functions $u_5(x_5|x_4)$ one per each analyzed value of x_4

and $x_{6,u_6} = 6000$. The acquired results are: $\hat{u}_{6,2} \in [\hat{u}^d_{6,2}; \hat{u}^u_{6,2}] \equiv [0.03; 0.08]$, $\hat{u}_{6,3} \in [\hat{u}^d_{6,3}; \hat{u}^u_{6,3}] \equiv [0.18; 0.25]$, $\hat{u}_{6,4} \in [\hat{u}^d_{6,4}; \hat{u}^u_{6,4}] \equiv [0.38; 0.50]$, $\hat{u}_{6,5} \in [\hat{u}^d_{6,5}; \hat{u}^u_{6,5}] \equiv [0.70; 0.79]$, $\hat{u}_{6,6} \in [\hat{u}^d_{6,6}; \hat{u}^u_{6,6}] \equiv [0.83; 0.89]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt} = 6.1275e - 4$, $x_{0,opt}$=4095.8, χ^2=0.2557. The resulting utility function is depicted on fig. 3. The utilities of the values of x_6 in the interval [1000; 7000], other than the investigated ones may be found using linear interpolation on the elicited nodes taking into account the direction of increase of the function.

4.3 Constructing the fundamental utility function $u_{y,4}(\vec{\mathbf{y}}_4)$

The fundamental vector attribute $\overrightarrow{\mathbf{y}}_4$ consists of the attribute x_7, which is a continuous random variable in the interval [0; 9000]. Therefore, $u_{y,4}(\overrightarrow{\mathbf{y}}_4) \equiv u_7(x_7)$. The preferences of the FRDM are strictly decreasing over the values of x_7. It is necessary to construct the one-dimensional utility function in that interval. LE is used to elicit 5 inner quantile indices of the quantiles $x_{7,u_2} = 1500, x_{7,u_3} = 3000, x_{7,u_4} = 4500, x_{7,u_5} = 6000$, and $x_{7,u_6} = 7500$. The acquired results are: $\hat{u}_{7,2} \in [\hat{u}^d_{7,2}; \hat{u}^u_{7,2}] \equiv [0.94; 0.98]$, $\hat{u}_{7,3} \in [\hat{u}^d_{7,3}; \hat{u}^u_{7,3}] \equiv [0.75; 0.85]$, $\hat{u}_{7,4} \in [\hat{u}^d_{7,4}; \hat{u}^u_{7,4}] \equiv [0.45; 0.58]$, $\hat{u}_{7,5} \in [\hat{u}^d_{7,5}; \hat{u}^u_{7,5}] \equiv [0.14; 0.25]$, $\hat{u}_{7,6} \in [\hat{u}^d_{7,6}; \hat{u}^u_{7,6}] \equiv [0.05; 0.12]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt} = 7.1343e - 4$, $x_{0,opt}$=4413, χ^2=0.4232. The resulting utility function is depicted on fig. 4. The utilities of the values of x_7 in the interval [0; 9000], other than the investigated ones may be found using linear interpolation on the elicited nodes taking

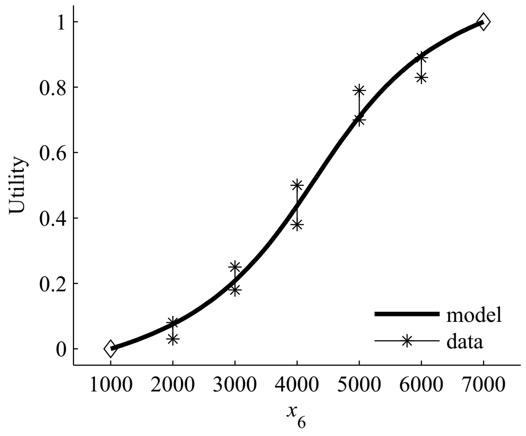


Figure 3: Arctg-approximated utility function over the values of x_6 in the interval [1000; 7000]

into account the direction of increase of the function.

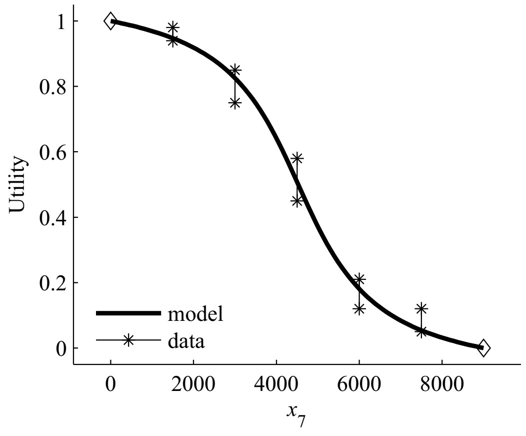


Figure 4: Arctg-approximated utility function over the values of x_7 in the interval [0; 9000]

4.4 Constructing the fundamental utility function $u_{y,5}(\vec{\mathbf{y}}_5)$

The fundamental vector attribute $\overrightarrow{\mathbf{y}}_5$ consists of the attribute x_8, which is a continuous random variable in the interval [60; 130]. Therefore, $u_{y,5}(\overrightarrow{\mathbf{y}}_5) = u_8(x_8)$, and $x_{8,min} = 60$, $x_{8,max} = 130$. The FRDM has hill preferences over the values of x_8. Following the algorithms from [19], the FRDM has elicited that the uncertainty interval of the extremum is $\hat{x}_{8,opt} \in [\hat{x}_{8,optmin}; \hat{x}_{8,optmax}] \equiv [82; 88]$. This divides the interval of x_8 into two sections with monotonic preferences from 60 to 82, and from 88 to 130.

The monotonically increasing utility function $u_{8,l}(.)$ is arctg-approximated in the interval [60; 82]. UE is used to elicit 3 inner quantiles with quantile indices $u_{8,l_2} =$

$0.25, u_{8,l_3} = 0.5, u_{8,l_4} = 0.75$. The acquired results are: $\hat{x}_{8,l_2} \in [\hat{x}^d_{8,l_2}; \hat{x}^u_{8,l_2}] \equiv [64; 68]$, $\hat{x}_{8,l_3} \in [\hat{x}^d_{8,l_3}; \hat{x}^u_{8,l_3}] \equiv [70; 75]$, $\hat{x}_{8,l_4} \in [\hat{x}^d_{8,l_4}; \hat{x}^u_{8,l_4}] \equiv [74; 78]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt} = 0.0155$, $x_{0,opt} = 94.2599$, $\chi^2 = 0.0839$.

The monotonically decreasing utility function $u_{8,r}(.)$ is arctg-approximated in the interval [88; 130]. LE is used to elicit 3 inner quantile indices of the quantiles $u_{8,r_2} = 0.75$, $u_{8,r_3} = 0.5$, $u_{8,r_4} = 0.25$. The acquired results are: $\hat{x}_{8,r_2} \in [\hat{x}^d_{8,r_2}; \hat{x}^u_{8,r_2}] \equiv [96; 100]$, $\hat{x}_{8,r_3} \in [\hat{x}^d_{8,r_3}; \hat{x}^u_{8,r_3}] \equiv [99; 108]$, $\hat{x}_{8,r_4} \in [\hat{x}^d_{8,r_4}; \hat{x}^u_{8,r_4}] \equiv [108; 114]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: a_{opt}=0.0932, $x_{0,opt}$=102.4850, χ^2=0.0026.

According to the FRDM, $60 \succ 130$, i.e. the global worst prize in the interval [60; 130] is identified as $x_{8,worst} = x_{8,max}$ $(u(x_{8,max}) = u_{8,max} = 0)$. The global utility of $x_{8,min}$ is elicited using LE, and the result is $u(x_{8,min}) = \hat{u}_{8,min} \in [\hat{u}^d_{8,min}; \hat{u}^u_{8,min}] \equiv [0.27; 0.37]$ with a point estimate $\hat{u}_{8,min} = 0.32$.

These results imply that it is necessary to rescale the utility function $u_{8,l}(.)$ following the ideas from [20]. The utilities $u_{8,l_q}(.)(q = 1, 2, \ldots, 5)$ can be rescaled as follows: $u^{resc}_{8,l_q} \in [u^{d,resc}_{8,l_q}; u^{u,resc}_{8,l_q}]$, where $u^{d,resc}_{8,l_q} = u_{8,l_q} \times (1 - u^d_{8,max}) + u^d_{8,max}$, $u^{u,resc}_{8,l_q} = u_{8,l_q} \times (1 - u^u_{8,max}) + u^d_{8,max}$.

The results from the rescaling are presented in Table 4. Figure 5 depicts the resulting utility function over the entire set of prizes [60; 130], along with the initial uncertainty intervals on the values of x_8 and the rescaled uncertainty intervals on u.

Table 4: Initial and rescaled utility values of the utility functions $u_{8,l}(.)$. The bolded values are the reference points in the rescaling of the function

q	u_{8,l_q}	$u^{d,resc}_{8,l_q}$	$u^{u,resc}_{8,l_q}$	$\hat{u}^{resc}_{8,l_q}$
1	0	**0.27**	**0.27**	**0.32**
2	0.25	0.4525	0.5275	0.49
3	0.5	0.6350	0.6850	0.66
4	0.75	0.8175	0.8425	0.83
5	1	**1**	**1**	**1**

4.5 Constructing the fundamental utility function $u_{y,6}(\vec{\mathbf{y}}_6)$

The fundamental vector attribute $\overrightarrow{\mathbf{y}}_6$ consists of the attribute x_9, which is a continuous random variable in the interval [60; 240]. Therefore, $u_{y,6}(\overrightarrow{\mathbf{y}}_6) \equiv u_9(x_9)$, and $x_{9,min}$=60, $x_{9,max}$=240. The FRDM has quasi-multi-modal preferences over the values of x_9, since there is a maximum and a minimum extremum within the prize interval. With the help of the algorithms from [19], the FRDM has elicited that the uncertainty interval of the maximum (hill) extremum is $\hat{x}^{hill}_{9,opt} \in [\hat{x}^{hill}_{9,optmin}; \hat{x}^{hill}_{9,optmax}] \equiv [120; 130]$, whereas the uncertainty interval of the minimum (valley) extremum is $\hat{x}^{valley}_{9,opt} \in [\hat{x}^{valley}_{9,optmin}; \hat{x}^{valley}_{9,optmax}] \equiv [185; 210]$. This divides the interval of x_9 into three sections with monotonic preferences – from 60 to 120, from 130 to 185, and from 210 to 240.

The monotonically increasing utility function $u_{9,1}(.)$ is arctg-approximated in the interval [60; 120]. UE is applied to elicit 3 inner quantiles with quantile indices $u_{9,1_2}$=0.25,

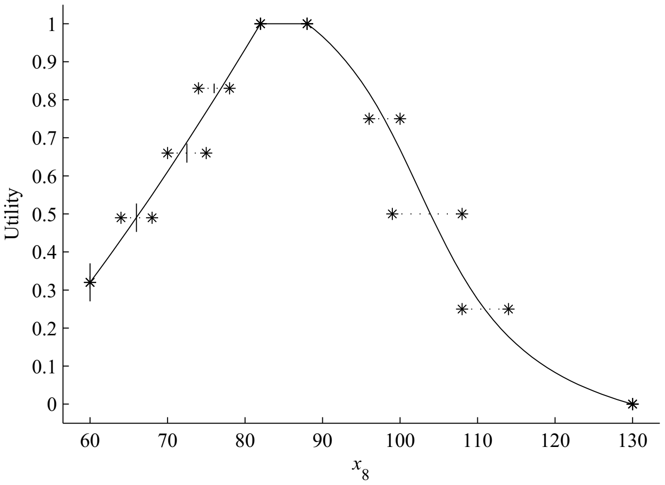


Figure 5: Hill utility function over the values of x_8 in the interval [60; 130], along with the initial and the rescaled uncertainty intervals

$u_{9,1_3}$=0.5, and $u_{9,1_4}$=0.75. The acquired results are: $\hat{x}_{9,1_2} \in [\hat{x}^d_{9,1_2}; \hat{x}^u_{9,1_2}] \equiv [77; 85], \hat{x}_{9,1_3} \in [\hat{x}^d_{9,1_3}; \hat{x}^u_{9,1_3}] \equiv [90; 102], \hat{x}_{9,1_4} \in [\hat{x}^d_{9,1_4}; \hat{x}^u_{9,1_4}] \equiv [104; 112]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: a_{opt}=0.0223, $x_{0,opt}$=110.5663, $\chi^2 = 5.0665e - 4$.

The monotonically decreasing utility function $u_{9,2}(.)$ is arctg-approximated in the interval [130; 185]. UE is applied to elicit 3 inner quantiles with quantile indices $u_{9,2_2}$=0.75, $u_{9,2_3}$=0.5, and $u_{9,2_4}$=0.25. The acquired results are: $\hat{x}_{9,2_2} \in [\hat{x}^d_{9,2_2}; \hat{x}^u_{9,2_2}] \equiv [142; 148], \hat{x}_{9,2_3} \in [\hat{x}^d_{9,2_3}; \hat{x}^u_{9,2_3}] \equiv [9148; 156], \hat{x}_{9,2_4} \in [\hat{x}^d_{9,2_4}; \hat{x}^u_{9,2_4}] \equiv [159; 165]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: a_{opt}=0.0715, $x_{0,opt}$=151.5196, $\chi^2 = 1.3123e - 2$.

The monotonically increasing utility function $u_{9,3}(.)$ is arctg-approximated in the interval [210; 240]. UE is applied to elicit 3 inner quantiles with quantile indices $u_{9,3_2}$=0.25, $u_{9,3_3}$=0.5, and $u_{9,3_4}$=0.75. The acquired results are: $\hat{x}_{9,3_2} \in [\hat{x}^d_{9,3_2}; \hat{x}^u_{9,3_2}] \equiv [220; 226], \hat{x}_{9,3_3} \in [\hat{x}^d_{9,3_3}; \hat{x}^u_{9,3_3}] \equiv [226; 236], \hat{x}_{9,3_4} \in [\hat{x}^d_{9,3_4}; \hat{x}^u_{9,3_4}] \equiv [231; 239]$. The optimal parameters of the arctg-approximated utility function, calculated using that data are: a_{opt}=0.0713, $x_{0,opt}$=235.9721, $\chi^2 = 1.0565e - 2$.

According to the FRDM, the global worst prize in the interval [60; 240] is identified as $x_{9,worst} = x_{9,min}$ $(u(x_{9,min}) = u_{9,min} = 0)$, whereas the global best prize is $x_{9,best} = x^{hill}_{9,opt}$ $(u(x^{hill}_{9,opt}) = u^{hill}_{9,opt} = 1)$ (this utility applies to all values of x_9 in the interval $[x^{hill}_{9,optmin}; x^{hill}_{9,optmax}]$). The global utility of x^{valley}_{opt} is elicited using LE, and the result is $u(x^{valley}_{9,opt}) = \hat{u}^{valley}_{9,opt} \in [\hat{u}^{d,valley}_{9,opt}; \hat{u}^{u,valley}_{9,opt}] \equiv [0.15; 0.25]$ with a point estimate $\hat{u}^{valley}_{9,opt} = 0.20$ (this utility applies to all values of x_9 in the interval $[\hat{x}^{valley}_{9,optmin}; \hat{x}^{valley}_{9,optmax}]$). The global utility of $x_{9,max}$ is elicited using LE, and the result is $u(x_{9,max}) = \hat{u}_{9,max} \in [\hat{u}^d_{9,max}; \hat{u}^u_{9,max}] \equiv [0.67; 0.73]$ with a point estimate $\hat{u}_{9,max}$=0.70.

These results imply that it is necessary to rescale the utility functions $u_{9,2}(.)$ and $u_{9,3}(.)$ following the ideas from [20]. The utilities $u_{9,2_l}(l = 1, 2, \ldots, 5)$ can be rescaled as follows: $u^{resc}_{9,2_l} \in [u^{d,resc}_{9,2_l}; u^{u,resc}_{9,2_l}]$, where $u^{d,resc}_{9,2_l} = u_{9,2_l} \times (\hat{u}^{hill}_{9,opt} - \hat{u}^{d,valley}_{9,opt}) + \hat{u}^{d,valley}_{9,opt}$,

$u^{u,resc}_{9,2_l} = u_{9,2_l} \times (\hat{u}^{hill}_{9,opt} - \hat{u}^{u,valley}_{9,opt}) + \hat{u}^{u,valley}_{9,opt}$. The utilities $u_{9,3_l} (l = 1, 2, \ldots, 5)$ can be rescaled as follows: $u^{resc}_{9,3_l} \in [u^{d,resc}_{9,3_l}; u^{u,resc}_{9,3_l}]$, where $u^{d,resc}_{9,3_l} = u_{9,3_l} \times (\hat{u}^{d}_{9,max} - \hat{u}^{d,valley}_{9,opt}) + \hat{u}^{d,valley}_{9,opt}$, $u^{u,resc}_{9,3_l} = u_{9,3_l} \times (\hat{u}^{u}_{9,max} - \hat{u}^{u,valley}_{9,opt}) + \hat{u}^{u}_{9,max}$.

The results from the rescaling are presented in Table 5. Figure 6 depicts the resulting utility function over the entire set of prizes [60; 240], along with the initial uncertainty intervals on the values of x_9 and the rescaled uncertainty intervals on u.

Table 5: Initial and rescaled utility values of the utility functions $u_{9,2}(.)$ and $u_{9,3}(.)$. The bolded values are the reference points in the rescaling of the function

l	$u_{9,2_l}$	$u^{d,resc}_{9,2_l}$	$u^{u,resc}_{9,2_l}$	$u^{resc}_{9,2_l}$	$u_{9,3_l}$	$u^{d,resc}_{9,3_l}$	$u^{u,resc}_{9,3_l}$	$u^{resc}_{9,3_l}$
1	1	**1**	**1**	**1**	0	**0.15**	**0.25**	**0.2**
2	0.75	0.7875	0.8125	0.8	0.25	0.28	0.37	0.325
3	0.5	0.5750	0.6250	0.6	0.5	0.41	0.49	0.45
4	0.25	0.3625	0.4375	0.4	0.75	0.54	0.61	0.575
5	0	**0.15**	**0.25**	**0.2**	1	**0.67**	**0.73**	**0.7**

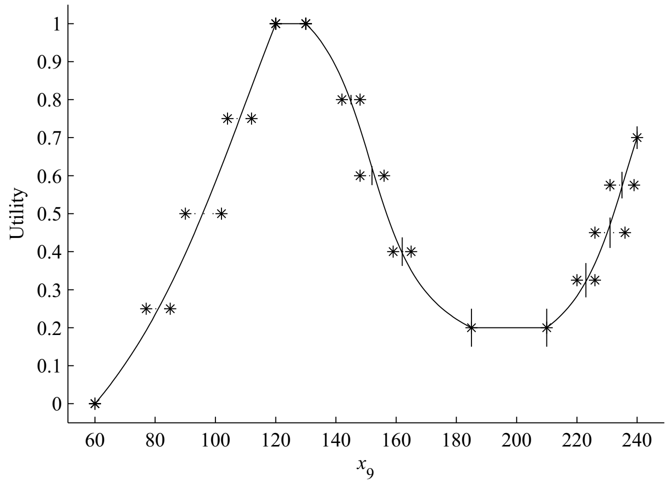


Figure 6: Graphics of the utility function over the entire set of values of x_9. The horizontal dotted lines indicate the elicited uncertainty intervals used to arctg-approximate the utility in the sections with monotonic preferences. The vertical solid lines indicate the uncertainty interval of the rescaled utilities

4.6 Eliciting the scaling constants and finding the form of the multi-dimensional utility function

Following (9) and the information from the previous subsections, it is possible to define six corner consequences as follows: $\overrightarrow{\mathbf{x}}_{1,corner} = [5, 90, 18, -2, -5, 1000, 9000, 130, 60]$, $\overrightarrow{\mathbf{x}}_{2,corner} = [1, 30, 6, 12, 35, 1000, 9000, 130, 60]$, $\overrightarrow{\mathbf{x}}_{3,corner} = [1, 30, 6, -2, -5, 7000, 9000, 130, 60]$,

$\overrightarrow{\mathbf{x}}_{4,corner} = [1, 30, 6, -2, -5, 1000, 0, 130, 60]$, $\overrightarrow{\mathbf{x}}_{5,corner} = [1, 30, 6, -2, -5, 1000, 9000, 82, 60]$, $\overrightarrow{\mathbf{x}}_{6,corner} = [1, 30, 6, -2, -5, 1000, 9000, 130, 120]$.

Using the algorithm from [17], the FRDM has elicited the uncertainty intervals of the corresponding scaling constants as follows ; $\hat{k}_{y,1} \in [\hat{k}^d_{y,1}; \hat{k}^u_{y,1}] \equiv [0.28; 0.31]$, $\hat{k}_{y,2} \in [\hat{k}^d_{y,2}; \hat{k}^u_{y,2}] \equiv [0.20; 0.27]$, $\hat{k}_{y,3} \in [\hat{k}^d_{y,3}; \hat{k}^u_{y,3}] \equiv [0.17; 0.22]$, $\hat{k}_{y,4} \in [\hat{k}^d_{y,4}; \hat{k}^u_{y,4}] \equiv [0.14; 0.20]$, $\hat{k}_{y,5} \in [\hat{k}^d_{y,5}; \hat{k}^u_{y,5}] \equiv [0.07; 0.15]$, $\hat{k}_{y,6} \in [\hat{k}^d_{y,6}; \hat{k}^u_{y,6}] \equiv [0.05; 0.12]$. It is necessary to construct the utility functions of the FRDM at a significance level of $\alpha = 0.05$. For the elicited data, $m = n = 6 > 0$, $a_6 = 0.91 < 1, b_6 = 1.27 > 1$, i.e. it is a non-trivial case. Using the simulation realization of the uniform method [17] at $N = 5000$, it was calculated that $\hat{p}_{value} \approx 0.0425$. Then H_0 (stating that the sum of the constants equals to one) is rejected and H_1 is accepted (stating that the sum of the constants is not equal to one) since $\hat{p}_{value} \approx 0.0425 < 0.05 = \alpha$. Then the estimates of the scaling constants are simply the midpoints of their uncertainty intervals: $\hat{k}_{y,1} = (\hat{k}^d_{y,1} + \hat{k}^u_{y,1})/2 = 0.295$, and similarly $\hat{k}_{y,2} = 0.235$, $\hat{k}_{y,3} = 0.195$, $\hat{k}_{y,4} = 0.17$, $\hat{k}_{y,5} = 0.11$, $\hat{k}_{y,6} = 0.085$. The value of the general constant $K_y = -0.1961$. Finally, the multi-dimensional utility function is multiplicative:

$$\begin{aligned} u(\overrightarrow{\mathbf{x}}) &= \frac{1}{K_y}\prod_{j=1}^{n}[K_y k_{y,j} u_{y,j}(\overrightarrow{\mathbf{y}}_j) + 1] - \frac{1}{K_y} = \\ &= \frac{1}{-0.1961} \times [1 - 0.0578 u_{y,1}(\overrightarrow{\mathbf{y}}_1)] \times [1 - 0.0461 u_{y,2}(\overrightarrow{\mathbf{y}}_2)] \times \\ &\times \ [1 - 0.0382 u_{y,3}(\overrightarrow{\mathbf{y}}_3)] \times [1 - 0.0333 u_{y,4}(\overrightarrow{\mathbf{y}}_4)] \times \\ &\times \ [1 - 0.0216 u_{y,5}(\overrightarrow{\mathbf{y}}_5)] \times [1 - 0.0167 u_{y,6}(\overrightarrow{\mathbf{y}}_6)] + \frac{1}{0.1961}. \end{aligned}$$

5 Summary and Conclusions

The discussion in the paper clearly demonstrated that constructing multi-dimensional utility functions is among the most difficult tasks in decision analysis. The formal setup of that task was given and the algorithm for direct construction of a multi-dimensional utility was presented. Following the arguments from [13] it was commented that this algorithm has no practical application in cases with more than three attributes. Therefore another seven-step algorithm was elaborated, which requires decomposing multi-dimensional consequences into mutually utility independent fundamental vector attributes. Then the multi-dimensional function is a combination of the utility functions over the vector attributes and their scaling constants. The algorithm took into account the fuzzy rationality of the real DM and the resulting interval estimates. It also accounted for the different type of preferences that may exist over the fundamental vector attributes. For that reason, the algorithm incorporated different utility elicitation techniques, procedures for elicitation of extrema of non-monotonic utility functions, approximation techniques, as well as methods for identification of the sum of the scaling constants (namely the uniform method), the latter indicating the final form (additive or multiplicative) of the multi-dimensional utility function. The presented numerical example well demonstrated all steps of the algorithm. Although the procedure is rather time consuming and complex, it is the only alternative to the analyzed problem, which allows generating adequate and consistent results.

Other practical examples should also be identified, which would allow to further verify the proposed algorithm.

References

[1] M. Abdellaoui, C. Barrios, P. Wakker, Reconciling Introspective Utility with Revealed Preference: Experimental Arguments Based on Prospect Theory, *Journal of Econometrics*, Vol. 138, pp. 356-378, 2007.

[2] F. Bacchus, A. Grove, Utility Independence in Qualitative Decision Theory, In: Aiello, L. C., Doyle, J., Shapiro, S. (Eds.), *Principles of Knowledge Representation and Reasoning*, CA, Morgan Kaufmann, pp. 542-552, 1996.

[3] R. Clemen, *Making Hard Decisions: an Introduction to Decision Analysis*, Second Edition. Duxbury Press, Wadsworth Publishing Company, 1996.

[4] C. De Boor, A Practical Guide to Splines, *Applied Mathematical Sciences*, No. 27, Springer, 2001.

[5] Y. Engel, M. Wellman, CUI Networks: A Graphical Representation for Conditional Utility Independence, *Proc. 21 National Conference on Artificial Intelligence*, pp. 1137-1142, 2006.

[6] P. H. Farquhar, Research Directions in Multi-Attribute Utility Analysis, In: Hansen, P. (Ed.), *Essays and Surveys on Multi-Criteria Decision Making*, Springler Verlag, pp. 63-85, 1983.

[7] P. H. Farquhar, Utility Assessment Methods, *Management Science*, Vol. 30, No. 11, pp. 1283-1300, 1984.

[8] P. Fishburn, F. Roberts, Mixture Axioms in Linear and Multilinear Utility Theories, *Theory and Decisions*, Vol. 9, No. 9, pp. 161-171, 2004.

[9] G. E. Forsythe, A. Malcolm, C. B. Moler, *Computer Methods for Mathematical Computations*, Prentice Hall, 1977.

[10] S. French, *Decision Theory: an Introduction to the Mathematics of Rationality*, Ellis Horwood, 1993.

[11] S. French, D. R. Insua, *Statistical Decision Theory*, Arnold, London, 2000.

[12] J. E. Hanke, A. G. Reitsch, *Understanding Business Statistics*, Irwin, 1991.

[13] R. L. Keeney, H. Raiffa, *Decisions with Multiple Objectives: Preference and Value Tradeoffs*, Cambridge University Press, 1993.

[14] J. Kiefer, Sequential Minimax Search for a Maximum, *Proc. American Mathematical Society*, Vol. 4, pp. 502-506, 1953.

[15] M. McCord, R. De Neufville, *Lottery Equivalents: Reduction of the Certainty Effect Problem in Utility Assessment*, Management Science, Vol. 32, pp. 56-60, 1986.

[16] N. D. Nikolova, Uniform Method for Estimation of Interval Scaling Constants, *Engineering and Automation Problems*, Vol. 1, pp. 79-90, 2007a.

[17] N. D. Nikolova, Empirical Connection between the Number of Elicited Knots and the Quality of Analytical Approximation of a One-Dimensional Utility Function, *Machine Building and Machine Learning, Economics and Management Series*, Year II, Book 1, pp. 99-111, 2007b (in Bulgarian)

[18] N. D. Nikolova, A. Shulus, D. Toneva, K. Tenekedjiev, Fuzzy Rationality in Quantitative Decision Analysis, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 9, No. 1, pp. 65-69, 2005.

[19] N. D. Nikolova, K. Hirota, C. Kobashikawa, K. Tenekedjiev, Elicitation of Non-Monotonic Preferences of a Fuzzy Rational Decision Maker, *Information Technologies and Control*, Year IV, Vol. 1, pp. 36-50, 2006.

[20] N. D. Nikolova, Elicitation errors in one-dimensional non-monotonic utility functions, *Computer Science and Technology*, No. 2,pp.48-59, 2007.

[21] N. D. Nikolova, S. Ahmed, K. Tenekedjiev, Generic Procedure for Construction of a Multi-Dimensional Utility Function Under Fuzzy Rationality, *International Journal of Computers, Communications & Control*, Vol. III, Supplementary Issue: Proceedings of the ICCCC 2008, pp. 422-426, 2008.

[22] J. W. Pratt, Risk Aversion in the Small and in the Large, *Econometrica*, Vol. 32, pp. 122-136, 1964.

[23] W. H. Press, S. A. Teukolski, W. T. Vetterling, B. P. Flannery, *Numerical Recipes the Art of Scientific Computing*, Cambridge University Press, 1992.

[24] S. Stoianov, *Optimization of Technological Processes*, Tehnika, 1993 (in Bulgarian).

[25] K. Tenekedjiev, N. D. Nikolova, Justification and Numerical Realization of the Uniform Method for Finding Point Estimates of Interval Elicited Scaling Constants, *Fuzzy Optimization and Decision Making*, Vol. 7, No. 2, pp. 119-145, 2008.

[26] K. Tenekedjiev, N. D. Nikolova, D. Dimitrakiev, Application of the Triple Bisection Method for Extraction of Subjective Utility Information, *Proc. Second International Conference "Management and Engineering'2004"*, Vol. 2, No. 70, pp. 115-117, 2004.

[27] K. Tenekedjiev, N. D. Nikolova, R. Pfliegl, Utility Elicitation with the Uncertain Equivalence Method, *Comptes Rendus De L'Academie Bulgare des Sciences*, Vol. 59, Book 3, pp. 283-288, 2006.

[28] K. Tenekedjiev, N. D. Nikolova, D. Dimitrakiev, Analytical One-Dimensional Utility Comparison of Power and Arctg-Approximation, *Engineering Science*, Vol. 4, pp. 19-32, 2007.

[29] J. Von Neumann, O. Morgenstern, *Theory of Games and Economic Behavior*, Second Edition, Princeton University Press, 1947.

[30] P. Wakker, D. Deneffe, Eliciting von Neumann-Morgenstern Utilities when Probabilities are Distorted or Unknown, *Management Science*, Vol. 42, pp. 1131-1150, 1996.

Natalia Nikolova, Daniela Toneva,
Sevda Ahmed, Kiril Tenekedjiev
Technical University - Varna
Dept. Economics and Management
1 Studentska Str., 9010 Varna, Bulgaria
E-mail: natalia@dilogos.com
d_toneva@abv.bg
sevdadan@hotmail.com
kiril@dilogos.com

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 150-171

Real Sets as Consistent Boolean Generalization of Classical Sets

Dragan G. Radojevic

Abstract: *Consistent Boolean generalization* of classical (two-valued) set theory into a real-valued theory of sets means preservation of all of its value indifferent - algebraic characteristics: Boolean axioms and theorems. Fuzzy set theories as the conventional generalizations of classical set theory are not in Boolean frame and/or they are not consistent Boolean generalization of classical set theory. Since theory of classical sets is based on the celebrated two-valued realization of Boolean algebra (BA) it follows that their Boolean consistent real-valued generalization should be based on a real-valued realization of BA. *Interpolative Boolean algebra* (IBA) is a real-valued ($[0,1]$-valued) realization of finite or atomic BA. The real-valued realization of atomic or finite BA is adequate for any real problem since gradation offers superior expressiveness in comparison to the black-white outlook. To every element of BA in IBA uniquely corresponds a generalized Boolean polynomial (GBP). GBP is a figure on a value level corresponding to a disjunctive canonical form from algebraic level. A disjunctive canonical form of analyzed element of BA is a disjunction (union, join) of relevant atoms and its GBP is a sum of atomic Boolean polynomial of these atoms. Which atoms are relevant for analyzed element of BA is determined by its structure function (containment). To structures there applies the algebraic principle of structure functionality: the structure of a combined element of BA can be calculated on the basis of structures of its components. The celebrated and very well known truth functionality principle is only a figure of structure functionality principle on a value level valid in and only in two-valued realization and/or truth functionality principle is not an algebraic principle. As a consequence truth functionality principle cannot be a base for consistent MV and real-valued generalization theories based on BA. GBP has the ability to process values of primary variables from real unit interval $[0,1]$ so as to preserve all algebraic characteristics on value level. In this paper is given the theory of real sets ($\boldsymbol{R}$-sets) based on IBA. Elements of IBA are proper properties which generate corresponding sets on the analyzed universe of discourse. GBP of analyzed property is a membership function of corresponding $\boldsymbol{R}$-sets. $\boldsymbol{R}$-sets are generalized sets, as fuzzy sets, but in Boolean frame.

Keywords: Boolean algebra, Interpolative realization of Boolean algebra - IBA, Structure of Boolean algebra element, Principle of structural functionality, Generalized Boolean polynomial - GBP, Generalized product, Fuzzy sets, Real sets - $\boldsymbol{R}$-sets.

1 Introduction

Classical mathematical theories in the Boolean frame [1] - the theory of sets, mathematical logic, theory of relations, probability, etc., are based on a black-white outlook. The black-white or two-valued outlook is inadequate for treating impreciseness inherent to many real problems (cognition as natural or artificial information processing systems for: perception, learning, reasoning, decision-making, communication and action; natural languages; quant physics phenomena etc.). This was a motive for the mathematical treatment of gradations and/or for developing real-valued theories: fuzzy sets [2], fuzzy logic [3, 4], fuzzy relations, etc. Another very important motive for treating gradations is reducing the complexity immanent to the mathematical analysis of real problems by classical approaches. For example: two elements of the analyzed universe of discourses can be discerned by the chosen property in the black-white or two-valued approaches, only if one has and the other hasn't this property. As a consequence, the number of necessary properties (complexity of problem) increases with the increasing number of elements which should be discerned between them. The complexity of this problem can be drastically reduced by introducing gradation since it is possible to discern elements by the same property on the basis of its realized intensities.

Classical set theory (as classical logic, classical theory of relation generally) is in the Boolean frame and/or relies on classical two-valued realization of Boolean algebra (BA). Two-valued realization of finite BA is based on its homomorphic mapping on two-element BA (by truth function in logic {truth, untruth}; by characteristic function in theory of sets {belong, does not belong}; by relation function in theory of relations {in relation, not in relation}, etc.). A truth table is only a table representation of homomorphic mapping of analyzed BA on two-element BA. All characteristics connected with a truth table such as the famous principle of truth functionality (and/or extensionality) are a direct consequence of homomorphic mapping and hold only in a classical two-valued case.

In many real applications the classical two-valued ("black and white") realization of BA [1] is not adequate. L. Zadeh, after his famous and distinguished contribution in the modern control theory, has ingeniously recognized the necessity of gradation in relations generally (theory of sets - fuzzy sets [2], logic - fuzzy logic [3], relations - fuzzy relations [4]).

Truth functionality principle

Conventional fuzzy approaches rely on the same principle as many-valued (MV) logics [5]. MV-logics are similar to classical logic because they accept the principle of truth-functionality [5]. Logic is truth functional if the truth value of a compound sentence depends only on the truth values of the constituent atomic sentences, not on their meaning or structure. The consequences of this direction are in the best way described by Lukasiewicz, the innovator of MV-logic: "Logic (truth functional) changes from its very foundations if we assume that in addition to truth and falsehood there is also some third logical value or several such values, " [6]. One "argument" for destroying the Boolean frame in treating gradation (MV-case) can be the definition of Boolean axioms of contradiction and excluded middle according to Aristotle: The same thing cannot at the same time both belong and not belong to the same object and in the same respect (Contradiction) Of any object, one thing must be either asserted or denied (Excluded middle). If the goal is mathematics for gradation then it seems "reasonable" to leave these axioms as inadequate and accept the principle of truth functionality with all consequences or to go to the very source of BA idea.

According to [7] fuzzy logic is based on truth functionality, since: "*This is very common and technically useful assumption*". A contrary example: "Our world is a flat plate" was also a very common and technically useful assumption in the Middle Ages!?

In the foundations of conventional MV approaches actually analyzed Boolean algebra is mapped onto a set of three or more scalars finally onto the real unit interval. Sets of three or more scalars $\{0, 1/2, 1\}, ..., \{0, 1/n, ..., 1\}$,...finally to the real unit interval $[0, 1]$ are not Boolean algebras. So, one cannot map Boolean algebra onto a set which is not Boolean algebra and in doing this preserve the properties of Boolean algebra and/or no conventional fuzzy set theory (fuzzy logic, theory of fuzzy relation) is in the Boolean frame. Structure $\langle [0, 1], T, S, N \rangle$, immanent to fuzzy set theories, can't be Boolean structure, and the implementation of particular properties of Boolean algebra by creating corresponding T norms irresistibly resembles alchemy.

Two fuzzy sets are equal according to extensionality if and only if they have the same elements with equal membership functions. Extensionality is accepted as a fundamental principle in conventional fuzzy set theories from classical set theory. That extensionality is not a natural assumption in a fuzzy case, is illustrated by the following simple but representative enough example: Suppose a trivial set with only one element - a glass. Let a glass be half full with beer. According to extensionality, it follows that the set generated by property "beer" - half full glass with beer and set generated by property "no beer" - half empty glass, are equal (membership functions in both case have equal vale 0.5)! Actually these two sets haven't anything in common except the fact that they are in the same glass.

It is interesting that in his seminal paper [1] G. Boole has said: "...the symbols of the (logic) calculus do not depend for their interpretation upon the idea of quantity..." and only "in their particular application..., conduct us to the quantitative conditions of inference". So, according to G. Boole, the principle of truth functionality is not a fundamental principle (and as a consequence this principle can't be the basis of any generalization).

Gradation in Boolean frame

A very important question is: *Can fuzziness and/or gradation be realized in a Boolean frame as a realization of BA?* We have obtained a positive answer to this question as an unexpected result, during solving the problem of fuzzy measure (or capacity) meaning in decision making by theory of capacity [8].

Boolean algebra (by axioms and theorems) defines value irrelevant properties which possess its elements. Classical two-valued realization of Boolean algebra (although extremely important) is only one of possible value realizations of Boolean algebra. The new approach to treating gradation in logic, theory of sets, relations etc., is based on interpolative realization of finite Boolean algebra (IBA) [9], [11].

IBA is real-valued and/or $[0, 1]$-valued realization of finite Boolean algebra. In IBA to any element of analyzed finite Boolean algebra uniquely corresponds a Generalized Boolean polynomial (GBP) (as, for example any element of finite Boolean algebra uniquely corresponds to a disjunctive canonical form.). GBP processes values from real unit interval $[0, 1]$. Values of GBP preserve partial order of corresponding Boolean algebra elements (based on value indifferent relation of inclusion) in all possible value realizations (so, for all values from real unit interval $[0, 1]$) by relation $(\leq)$.

IBA is real-valued and/or $[0, 1]$-valued realization of finite Boolean algebra. In IBA a Generalized Boolean polynomial (GBP) uniquely corresponds to any element of analyzed finite BA. GBP processes values from real unit interval $[0, 1]$. Values of GBP preserve partial order of corresponding BA elements (based on value indifferent relation of inclusion) in all possible value realizations (so, for all values from real unit interval $[0, 1]$) by relation $(\leq)$.

In case of $\boldsymbol{R}$-sets (idea of fuzzy sets realized in Boolean frame) properties are the domain of BA. Any element of Boolean algebra represents a corresponding property, which generates a corresponding R-set on the analyzed universe of discourses - value level. Membership function of analyzed R-set is defined by a corresponding GBP.

GBP of any property - element of BA is given by the superposition of relevant atomic GBP (as any element of finite Boolean algebra can be represented as a union of relevant atomic elements - a disjunctive canonical form). Any atomic element of analyzed BA of properties generates, on the universe of discourses, a corresponding atomic $\boldsymbol{R}$-set. GBP of atomic property is a membership function of corresponding atomic $\boldsymbol{R}$-set. Class of atomic $\boldsymbol{R}$-sets is the partition of analyzed universe of discourses. Intersection of any two different atomic $\boldsymbol{R}$-sets is an empty set and union of all atomic $\boldsymbol{R}$-sets is equal to the universe of discourses. In a classical case any object of universe can be the element of only one atomic set, but in case of R-sets it can be the element of a few atomic sets (in a special case, all of them) but so that the sum of values of corresponding atomic membership functions is equal to 1. Besides simultaneously owning a few atomic properties from the analyzed element of universe of discourses, intersection among atomic sets is always empty. So, simultaneously owning properties and the intersection of properties in the case of R-sets are not synonyms as in a classical case. Simultaneous is only a necessary but not a sufficient condition for the intersection of two $\boldsymbol{R}$-sets in a general case. As a consequence, the known Aristotelian definition of excluded middle and contradiction are valid only for a classical case and can't be of any value for a general case and/or for $\boldsymbol{R}$-sets. Excluded middle and contradiction as well as all other axioms and theorems of BA are value indifferent and they are valid in all possible realizations of $\boldsymbol{R}$-sets, as in the case of classical sets since classical sets can be treated as a special case of $\boldsymbol{R}$-sets.

All results based on the theory of classical sets can be consistently (preserving Boolean value irrelevant characteristics - Boolean axioms and theorems) generalized straightaway by $\boldsymbol{R}$-sets. For example: Using $\boldsymbol{R}$-sets instead of classical sets, classical theory of probability can be consistently generalized into $\boldsymbol{R}$-probability, preserving all value irrelevant characteristics with much richer interpretation.

2 Interpolative Boolean algebra: Real-valued realization of Boolean algebra

Interpolative Boolean algebra (IBA) [8, 13] is a real-valued realization of atomic or finite BA. The real-valued realization of atomic or finite BA is adequate for any real problem since gradation offers superior expressiveness in comparison to the black-white outlook.

Technically, IBA is based on generalized Boolean polynomials (GBP-s), [13]. GBP $\varphi^{\otimes}$ uniquely corresponds to the analyzed element $\varphi \in BA$ of atomic BA. GBP is a mapped disjunctive canonical form of analyzed BA element, on a value level as its polynomial "figure". Disjunctive canonical form is disjunction (union, join) of relevant atoms and GBP is sum of atomic Boolean polynomials of these atoms. As a consequence GBP can process values from a real unit interval $[0, 1]$ so as to preserve all algebraic properties of this element by corresponding arithmetic consequents. For example: (a) If two Boolean functions are equal $\varphi = \psi$ (algebraic property) then in all value realizations $\varphi^{\otimes} \equiv \psi^{\otimes}$ (arithmetic property). (b) If one element of BA is included in another $\varphi \subset \psi$, $(\varphi, \psi \in BA)$ (algebraic property) then, in all possible value realizations, the value of the first element is less or equal to the value of the second $\varphi^{\otimes} \leq \psi^{\otimes}$ (arithmetic property).

Disjunctive canonical form and structure (content) of BA element

The simplest elements of finite BA are its atoms, defined by the following expression:

$$\alpha(S) =_{def} \bigcap_{a_i \in S} a_i \bigcap_{a_j \in \Omega \backslash S} \mathcal{C}a_j,$$

$(S \in P(\Omega))$.

Combined elements of finite BA are built by relevant atoms. Which atoms are relevant for the analyzed element is determined by its *structure (content)*. Structure in a general case maps:

$$\sigma : P(\Omega) \times BA(\Omega) \to \{0, 1\},$$

and/or analyzed element $\varphi \in BA(\Omega)$ structure maps:

$$\sigma_{\varphi} : P(\Omega) \to \{0, 1\}.$$

Structure of analyzed element $\varphi \in BA(\Omega)$ is defined by the following set function:

$$\sigma_{\varphi}(S) =_{def} \begin{cases} 1, & \alpha(S) \subset \varphi; \quad (\alpha(S) \cap \varphi = \alpha(S)) \\ 0, & \alpha(S) \not\subset \varphi; \quad (\alpha(S) \cap \varphi = \underline{0}) \end{cases};$$

$(S \in P(\Omega)); \quad (\varphi, \underline{0} \in BA(\Omega))$

Any element $\varphi \in BA(\Omega)$ of analyzed BA, generated by the finite set of free (primary) variables $\Omega = \{a_1, \ldots, a_n\}$, can be represented in a *disjunctive canonical form* as the join (disjunction, union) of relevant atoms:

$$\varphi = \bigcup_{S \in P(\Omega) | \sigma_{\varphi}(S) = 1} \alpha(S).$$

Which atoms are relevant for the analyzed element is determined by its structure (content). Structure is a value indifferent - algebraic characteristic and/or it is invariant on the type of value realization (two-valued, many-valued and/or real-valued).

The structure or content σ_{φ} of any element $\varphi \in BA(\Omega)$ of the analyzed BA is homomorphism

$$\sigma_{\varphi \cap \psi}(S) = \sigma_{\varphi}(S) \wedge \sigma_{\psi}(S),$$

$$\sigma_{\varphi \cup \psi}(S) = \sigma_{\varphi}(S) \vee \sigma_{\psi}(S),$$

$$\sigma_{\mathcal{C}\varphi}(S) = \neg \sigma_{\varphi}(S);$$

$(S \in P(S), \quad \varphi, \psi \in BA(\Omega), \quad \sigma_{\varphi}(S), \sigma_{\psi}(S) \in 0, 1)$. and/or it homomorphically maps the analyzed $BA - \sigma_{\varphi} : P(\Omega) \to \{0, 1\}(\varphi \in BA(\Omega))$.

Structure functionality principle is a fundamental algebraic principle defined in the following way: the structure (content) of any combined element of analyzed BA can be determined directly on the basis of structures of its components.

All Boolean axioms and theorems defined for elements of BA satisfy the structures of these elements too. Since the values of structure components of analyzed element of BA are coincident with the values of this element in two-valued realizations:

$$\sigma_\varphi(S) = \varphi(a_1^S, \ldots, a_n^S).$$

Where:

$$a_i^S =_{def} \begin{cases} 1, & a_i \in S \\ 0, & a_i \notin S \end{cases};$$

$(a_i \in \Omega, \quad S \in P(\Omega))$.

From above it follows that *truth functionality principle* is the realization of structure functionality principle on a value level *in and only in the two-valued realization.* Truth functionality principle is arithmetic and as a non-algebraic characteristic it couldn't be a basis for consistent generalization.

Generalized Boolean polynomials

A generalized Boolean polynomial (GBP) uniquely corresponds to any element of analyzed atomic BA. GBP is the superposition of relevant atomic GBP-s. Which atoms are relevant for the analyzed element of BA is defined by its structure (content). *Atomic* GBP is defined by the following expression:

$$\alpha^\otimes(S)(a_1, \ldots, a_n) = \bigotimes_{a_i \in S} a_i \bigotimes_{a_j \in \Omega \setminus S} (1 - a_j).$$

Where: $\otimes$ is *generalized product* (from *min* function to the standard product ×).

Atomic GBPs have the following properties, as direct consequences of corresponding algebraic characteristics:

(a) Sum of values of all atoms is identically equal to 1:

$$\sum_{S \in P(\Omega)} \alpha^\otimes(S)(a_1, \ldots, a_n) = 1.$$

(b) Value of any atom is nonnegative:

$$\alpha^\otimes(S)(a_1, \ldots, a_n) \geq 0,$$

$(S \in P(\Omega), \quad a_1, \ldots, a_n \in [0,1])$.

GBP $\varphi^\otimes$ of analyzed element $\varphi \in BA(\Omega)$ is the superposition of relevant atomic GBP-s, as a polynomial figure of its disjunctive canonical form. Which atoms are relevant for analyzed element of BA is defined by its structure $\sigma_\varphi(S)$, $(S \in P(\Omega))$:

$$\begin{aligned}\varphi^{\otimes}(a_1,\dots,a_n) &= \sum_{S\in P(\Omega)|\sigma_{\varphi}(S)=1} \alpha^{\otimes}(S)(a_1,\dots,a_n),\\ &= \sum_{S\in P(\Omega)} \sigma_{\varphi}(S)\alpha^{\otimes}(S)(a_1,\dots,a_n).\end{aligned}$$

In a *classical two-valued* realization, as a special case, only one atom is equal to one and all others are equal to zero. All elements of BA in which a realized atom (with value 1) is included are equal to 1 and all other are equal to 0. As a consequence in a classical case for a given realized atom any element of BA has a value equal to 1 or 0 (*excluded middle*) and can't be 1 and 0 (*contradiction*).

Excluded middle is a *value-irrelevant* characteristic and from the atoms point of view, as a consequence that all atoms of analyzed $BA(\Omega)$ are included in the analyzed element $a \in BA(\Omega)$ and in its complement $\mathcal{C}a \in BA(\Omega)$, means, that all atoms are included in their join $a \cup \mathcal{C}a$ and/or join of any element of BA and its complement is equal to $\bar{1} \in BA(\Omega)$, since $\bar{1}$ includes in itself all atoms of analyzed BA:

$$a \cup \mathcal{C}A = \bar{1}.$$

Contradiction is a *value-irrelevant* characteristic and from the atoms point of view, as a consequence that no one atom of analyzed $BA(\Omega)$ is included in the analyzed element $a \in BA(\Omega)$ and in its complement $Ca \in BA(\Omega)$, means, that no one atom is included in their meet $a \cap \mathcal{C}a$ and/or meet of any element of BA and its complement is equal to $\underline{0} \in BA(\Omega)$, since $\underline{0}$ includes in itself no one atom of analyzed BA:

$$a \cap \mathcal{C}A = \underline{0}.$$

Transformation of Boolean function into GBP

Any Boolean function can be uniquely transformed into its GBP. Procedure of transformation of Boolean function into corresponding GBP is:

(a) For combined elements:

$$\begin{aligned}(\varphi\cap\psi)^{\otimes}(a_1,\dots,a_n) &= {}_{def}\varphi^{\otimes}(a_1,\dots,a_n)\otimes\psi^{\otimes}(a_1,\dots,a_n),\\ (\varphi\cup\psi)^{\otimes}(a_1,\dots,a_n) &= {}_{def}\varphi^{\otimes}(a_1,\dots,a_n)+\psi^{\otimes}(a_1,\dots,a_n)-\\ -\varphi^{\otimes}(a_1,\dots,a_n)\otimes\psi^{\otimes}(a_1,\dots,a_n),&\\ (\mathcal{C}\varphi)^{\otimes}(a_1,\dots,a_n) &= {}_{def}1-\varphi^{\otimes}(a_1,\dots,a_n).\end{aligned}$$

(b) For primary variables:

$$\begin{aligned}(a_i\cap a_j)^{\otimes} &= {}_{def}\begin{cases} a_i\otimes a_j, & i\neq j\\ a_i, & i=j\end{cases},\\ (a_i\cup a_j)^{\otimes} &= {}_{def}a_i+a_j-(a_i\cap a_j)^{\otimes},\\ (\mathcal{C}a_i)^{\otimes} &= {}_{def}1-a_i.\end{aligned}$$

***Example**: A few examples illustrate transformation of Boolean function into GBP:*

(a) *Contradiction*

$$\begin{aligned}(a \cap \mathcal{C}a)^{\otimes} &= a \otimes (1-a),\\ &= a - a,\\ &= 0\end{aligned}$$

(b) *Excluded middle*:

$$\begin{aligned}(a \cup \mathcal{C}a)^{\otimes} &= a + (1-a) - (a \cap \mathcal{C}a)^{\otimes},\\ &= 1.\end{aligned}$$

(c) *Nilpotent*

$$(a \cap a)^{\otimes} = a, \quad (a \cup a)^{\otimes} = a + a - (a \cap a)^{\otimes} = a.$$

(d) *Three Boolean functions*

d.1

$$\begin{aligned}(a \cap \mathcal{C}b)^{\otimes} &= a \otimes (1-b),\\ &= a - a \otimes b.\end{aligned}$$

d.2

$$\begin{aligned}(\mathcal{C}a \cap \mathcal{C}b)^{\otimes} &= (1-a) \otimes (1-b),\\ &= 1 - a - b + a \otimes b.\end{aligned}$$

d.3

$$\begin{aligned}((a \cap c) \cup (\mathcal{C}a \cap b))^{\otimes} &= (a \cap c)^{\otimes} + (\mathcal{C}a \cap b)^{\otimes} - (a \cap c)^{\otimes} \otimes (Ca \cap b)^{\otimes},\\ &= a \otimes c + (1-a) \otimes b - a \otimes c \otimes (1-a) \otimes b,\\ &= b + a \otimes c - a \otimes b - a \otimes (1-a) \otimes c \otimes b,\\ &= b + a \otimes c - a \otimes b.\end{aligned}$$

3 Real Sets based on Interpolative Boolean algebra

R-sets are generated by analyzed *proper properties* or Boolean properties - unary relations, on universe of discourses - value level. The set of proper properties, generators of ***R***-sets is *Boolean algebra of analyzed properties BA_p. Value indifferent (algebraic) characteristics* of proper properties are: *Boolean axioms and theorems.* In a general case any element of ***R***-set, as of fuzzy set, has an analyzed property with *intensity* and/or *gradation*:

$$F : X \to [0,1],$$

($F \in BA_p$) and X is analyzed universe of discourses.

Set of primary properties: $\Omega = A_1, \ldots, A_n$, generates finite Boolean algebra of analyzed properties $BA_p(\Omega)$. The basic characteristic of any primary property $A \in \Omega$ is the fact that it can't be represented by Boolean function of the remaining primary properties. Boolean algebraic structure of analyzing proper properties is:

$$\langle BA_p(\Omega), \cap, \cup, \mathcal{C} \rangle$$

Atomic property generates on analyzed universe of discourses (value level) atomic sets. Atomic properties are defined by following expressions:

$$A(S) = \bigcap_{A_i \in S} A_i \bigcap_{A_j \in \Omega \setminus S} A_j^{\mathcal{C}}, \quad (S \in P(\Omega));$$

$(S \in P(\Omega))$, where, P is power set - set of all subsets of analyzed set of primary properties Ω.

As a consequence if number of primary properties (cardinality of set Ω) is equal n then number of atomic properties is 2^n.

Structure function (containment) of any property $F \in BA_p(\Omega)$ determined which atoms are included in it and/or which are not included in it. Structure function of any property $F \in BA_p(\Omega)$ maps

$$\sigma_F : P(\Omega) \rightarrow \{0, 1\}$$

Structure function is given by the following expression:

$$\sigma_F(S) = \begin{cases} 1, & \alpha(S) \subset F; \quad (F \cap \alpha(S) = A(S)) \\ 0, & \alpha(S) \not\subset F; \quad (F \cap \alpha(S) = \underline{0}) \end{cases};$$

$(\underline{0}, F \in BA_p(\Omega); \quad S \in P(\Omega))$.

Fundamental structure's characteristic is ***principle of structural functionality***: *Structure of any combined property (element of Boolean algebra of analyzed properties) can be directly calculated on the base of structures of its components and the following rules*:

$$\begin{aligned} \sigma_{F \cup G}(S) &= \sigma_F(S) \vee \sigma_G(S), \\ \sigma_{F \cap G}(S) &= \sigma_F(S) \wedge \sigma_G(S), \\ \sigma_{F^c} &= \neg \sigma_F(S). \end{aligned}$$

$(F, G \in BA_p(\Omega), \quad S \in P(\Omega))$.

Where: $\neg$ unary and $\vee, \wedge$ binary classical two-valued Boolean operators:

$$\begin{array}{c|cc} \wedge & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}; \quad \begin{array}{c|cc} \vee & 0 & 1 \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \end{array}; \quad \begin{array}{c|c} & \neg \\ \hline 0 & 1 \\ 1 & 0 \end{array}.$$

Structures of primary properties are given by the following set functions:

$$\sigma_{A_i}(S) = \begin{cases} 1, & A_i \in S \\ 0, & A_i \notin S \end{cases};$$

$(S \in P(\Omega); \quad A_i \in \Omega)$.

Famous *principle of truth functionality* on value level is only the figure of value irrelevant principle of structural functionality and it is valid only for two-valued case.

Any element $F \in BA_p(\Omega)$ of finite Boolean algebra of proper properties can be uniquely represented by the following disjunctive canonical form:

$$F = \bigcup_{S \in P(\Omega) | \sigma_F(S)=1} A(S).$$

Since structure functions are homomorphism it follows:

$$F \cup G = \bigcup_{S \in P(\Omega) | \sigma_F(S) \vee \sigma_G(S)=1} A(S).$$

$$F \cap G = \bigcup_{S \in P(\Omega) | \sigma_F(S) \wedge \sigma_G(S)=1} A(S).$$

$$F^C = \bigcup_{S \in P(\Omega) | \neg \sigma_F(S)=1} A(S).$$

Generalized Boolean Polynomial

Generalized Boolean polynomials uniquely correspond to elements of Boolean algebra of analyzed properties - generators of ***R***-sets. ***R-characteristic function*** of any ***R***-set is its generalized Boolean polynomial.

Atomic **R**-*set* $A^{\otimes}(S)$ is value realization of corresponding atomic property $A(S)$, $(S \in P(\Omega))$ on analyzed universe of discourses $\times$. ***R***-characteristic function $A^{\otimes}(S) : X \to [0,1]$ of any atomic ***R***-set $A^{\otimes}(S)$, $(S \in P(\Omega))$is defined by corresponding ***atomic generalized Boolean polynomial***, [11, 13]:

$$A^{\otimes}(S)(x) = \sum_{K \in P(\Omega \setminus S)} (-1)^{|K|} \bigotimes_{A_i \in K \cup S} A_i(S),$$

$(A_i \in \Omega, \quad x \in X)$.

Example: *Atomic Boolean polynomials - atomic* **R**-*characteristic function for the case when set of primary properties is* $\Omega = \{a, b\}$, *are given in the following table:*

Any element of universe of discourses $x \in X$ has analyzed property $F \in BA_p(\Omega)$ with intensity which is given by corresponding ***generalized Boolean polynomial***:

$$\begin{aligned} F^{\otimes}(x) &= \sum_{S \in P(\Omega) | \sigma_F(S)=1} A^{\otimes}(S)(x) \\ &= \sum_{S \in P(\Omega)} \sigma_F(S) A^{\otimes}(S)(x), \end{aligned}$$

Table 1: *Example of Atomic Boolean polynomials.*

S	$A(S)$	$A^{\otimes}(S)(x)$
$\varnothing$	$A^c \cap B^c$	$1 - A(x) - B(x) + A(x) \otimes B(x)$
$\{A\}$	$A \cap B^c$	$A(x) - A(x) \otimes B(x)$
$\{B\}$	$A^c \cap B$	$B(x) - A(x) \otimes B(x)$
$\{A, B\}$	$A \cap B$	$A(x) \otimes B(x)$

$(S \in P(\Omega), \quad x \in X)$.

By definition ***structure vector*** *of* F and ***vector of atomic polynomials***, respectively

$$\begin{aligned} \vec{\sigma}_F &= [\sigma_F^v(S)|S \in P(\Omega)], \\ \vec{A}^{\otimes}(x) &= [A^{\otimes}(S)(x)|S \in P(\Omega)]^T, . \end{aligned}$$

Generalized Boolean polynomial - $\boldsymbol{R}$-characteristic function, of $F \in BA_p(\Omega)$ can be represented as scalar product of two vectors:

$$F^{\otimes}(x) = \vec{\sigma}_{\varphi}(S)\vec{A}^{\otimes}(x)$$

It is clear that the structure vector has algebraic nature, since it is value indifferent and preserves all Boolean axioms:

- Associativity

$$\begin{aligned} \vec{\sigma}_{F \cup (G \cup H)} &= \vec{\sigma}_{(F \cup G) \cup H}, \\ \vec{\sigma}_{F \cap (G \cap H)} &= \vec{\sigma}_{(F \cap G) \cap H}; \end{aligned}$$

- Commutativity

$$\begin{aligned} \vec{\sigma}_{F \cup G} &= \vec{\sigma}_{G \cup F}, \\ \vec{\sigma}_{F \cap G} &= \vec{\sigma}_{G \cap F}; \end{aligned}$$

- Absorption

$$\begin{aligned} \vec{\sigma}_{F \cup (G \cap H)} &= \vec{\sigma}_F, \\ \vec{\sigma}_{F \cap (G \cup H)} &= \vec{\sigma}_F; \end{aligned}$$

- Distributivity

$$\begin{aligned} \vec{\sigma}_{F \cup (G \cap H)} &= \vec{\sigma}_{(F \cup G) \cap (F \cup H)}, \\ \vec{\sigma}_{F \cap (G \cup \phi)} &= \vec{\sigma}_{(F \cap G) \cup (F \cap H)}; \end{aligned}$$

- Complements: Excluded middle and contradiction

$$\begin{aligned}\vec{\sigma}_{F\cup F^c} &= \vec{1},\\ \vec{\sigma}_{F\cap F^c} &= \vec{0}.\end{aligned}$$

And theorems of Boolean algebra:

- Idempotency

$$\begin{aligned}\vec{\sigma}_{F\cup F} &= \vec{\sigma}_F,\\ \vec{\sigma}_{F\cap F} &= \vec{\sigma}_F;\end{aligned}$$

- Boundedness

$$\begin{aligned}\vec{\sigma}_{F\cup\underline{0}} &= \vec{\sigma}_F, \quad \vec{\sigma}_{F\cap\bar{1}} = \vec{\sigma}_F;\\ \vec{\sigma}_{F\cup\bar{1}} &= \vec{1}, \quad\ \ \vec{\sigma}_{F\cap\underline{0}} = \vec{0}.\end{aligned}$$

- 0 and 1 are complements

$$\begin{aligned}\vec{\sigma}_{\underline{0}} &= \vec{1},\\ \vec{\sigma}_{\bar{1}^c} &= \vec{0};\end{aligned}$$

- De Morgan's laws

$$\begin{aligned}\vec{\sigma}_{(F\cup\psi)^c} &= \vec{\sigma}_{F^c\cap\psi^c},\\ \vec{\sigma}_{(F\cap\psi)^c} &= \vec{\sigma}_{F^c\cup\psi^c};\end{aligned}$$

- Involution

$$\vec{\sigma}_{(F^c)^c} = \vec{\sigma}_F.$$

Any $\boldsymbol{R}$-set $F^{\otimes}$, generated by corresponding property $F \in BA_p(\Omega)$, can be represented as union of relevant atomic sets:

$$F^{\otimes}(x) = \sum_{S\in P(\Omega)|\sigma_F(S)=1} A^{\otimes}(S)(x)$$

Structures of proper properties preserve value irrelevant characteristics of $\boldsymbol{R}$-sets, actually their Boolean nature.

Generalized Product

In GBPs there figure two standard arithmetic operators $+$ and $-$, and as a third generalized product $\otimes$. *Generalized product* is any function $\otimes : [0,1]\times[0,1] \to [0,1]$ that satisfies all four axioms of *T-norms* [12]:

Commutativity:

$$A(x) \otimes B(x) = B(x) \otimes A(x),$$

$(A, B \in \Omega, \quad A(x), B(x) \in [0,1], \quad x \in X)$.
Associativity:

$$A(x) \otimes B(x) \otimes C(x) = A(x) \otimes B(x) \otimes C(x),$$

$(A, B, C \in \Omega, \quad A(x), B(x), C(x) \in [0,1], \quad x \in X)$.
Monotonicity:

$$A(x) \leq B(x) \Rightarrow A(x) \otimes C(x) \leq B(x) \otimes C(x),$$

$(A, B, C \in \Omega, \quad A(x), B(x), C(x) \in [0,1], \quad x \in X)$.
Boundary:

$$A(x) \otimes 1 = A(x),$$

$(A \in BA(\Omega), \quad A(x) \in [0,1], \ \ x \in X)$.
and plus one additional axiom:

Non-negativity condition

$$\sum_{K \in P(\Omega \setminus S)} (-1)^{|K|} \bigotimes_{A_i \in K \cup S} A_i()x \geq 0,$$

$(\Omega = \{A_1, \ldots, A_n\}, \ \ S \in P(\Omega), \ \ A_i \in [0,1], \ \ x \in X)$.

Additional axiom “non-negativity” ensures that the values of atomic Boolean polynomials are non-negative: $A^{\otimes}(S)(x) \geq 0, \ (S \in P(\Omega), \ \ x \in X)$. As a consequence all elements of Boolean algebra and/or membership functions of all ***R***-sets are non-negative.

Comment: *Generalized product for **R**-sets is just an arithmetic operator and as a consequence has a crucially different role from the role of the T-norm in conventional fuzzy set theories, where it is a set algebraic operator.*

Example: In the case $\Omega = \{A, B\}$ generalized product, according to axioms of non-negativity can be in the following interval[1] :

$$max(A(x) + B(x) - 1, 0) \leq A(x) \otimes B(x) \leq min(A(x), B(x)).$$

Membership functions of intersection and union of two ***R***-sets are given by the following expressions:

$$(F \cap G)^{\otimes}(x) = (\vec{\sigma}_F \wedge \vec{\sigma}_G\vec{)} A^{\otimes}(x),$$

$$(F \cup G)^{\otimes}(x) = (\vec{\sigma}_F \vee \vec{\sigma}_G\vec{)} A^{\otimes}(x),$$

$(F \in BA_p(\Omega), \ \ x \in X)$.

Membership function of analyzed ***R***-set complement is:

[1] $max(A(x) + B(x) - 1, 0)$ is no more the low bound of feasible interval for generalized product in the case $|\Omega| \geq 3$.

$$(F^{\mathcal{C}})^{\otimes}(x) = 1 - F^{\otimes}(x),$$

$(F \in BA(\Omega),\ \ x \in X)$.

In new approach excluded middle and contradiction are always valid such as all other axioms and theorems of Boolean algebra for all possible generalized products:

$$\begin{aligned}(F \cup F^{\mathcal{C}})^{\otimes}(x) &= (\vec{\sigma}_F \vee \vec{\sigma}_{F^{\mathcal{C}}})\vec{A}^{\otimes}(x),\\ &= \vec{1}\vec{A}^{\otimes}(x),\\ &= 1;\end{aligned}$$

$$\begin{aligned}(F \cap F^{\mathcal{C}})^{\otimes}(x) &= (\vec{\sigma}_F \wedge \vec{\sigma}_{F^{\mathcal{C}}})\vec{A}^{\otimes}(x),\\ &= \vec{0}\vec{A}^{\otimes}(x),\\ &= 0;\end{aligned}$$

$(F \in BA(\Omega),\ \ x \in X)$.

Example: *The most simple characteristics are illustrated on the Boolean algebra of proper properties generated by only one primary property $\Omega = \{A\}$. Boolean algebra is the following set:*

$$BA_p(\{A\}) = \{\varnothing, \{\varnothing\}, \{\{A\}\}, \{\varnothing, \{A\}\}\},$$

After the following replacement:

$$\begin{aligned}\varnothing &\rightarrow \underline{0},\\ \{\varnothing\} &\rightarrow A^{\mathcal{C}},\\ \{\{A\}\} &\rightarrow A,\\ \{\varnothing, \{A\}\} &\rightarrow \bar{1}.\end{aligned}$$

Boolean algebra of properties is:

$$BA_p(\{A\}) = \{\underline{0}, A^{\mathcal{C}}, A, \bar{1}\}.$$

A and $A^{\mathcal{C}}$ are atomic elements of Boolean algebra $BA_p(\{A\})$ for which are valid the following identities:

$$\begin{aligned}A^{\mathcal{C}} \cap A &= \underline{0},\\ A^{\mathcal{C}} \cup A &= \bar{1}.\end{aligned}$$

Contradiction and excluded middle respectively. This simple Boolean algebra of property can be represented by Hasse diagrams. In figure 1 is illustrated Boolean algebra before and after introducing mentioned replacement

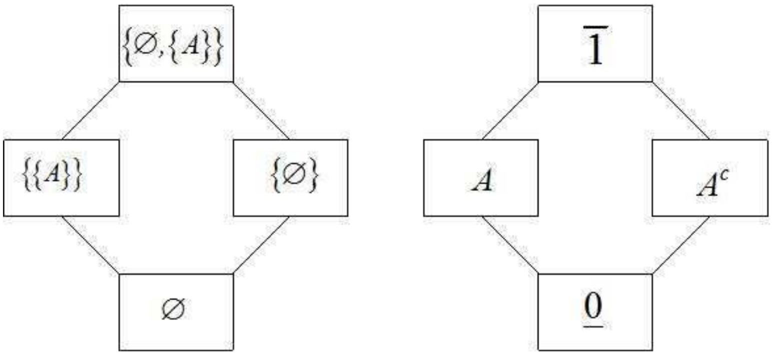


Figure 1: Hill utility function over the values of x_8 in the interval [60; 130], along with the initial and the rescaled uncertainty intervals

Elements of Boolean algebra $BA_p(\{A\})$ characterize their structure, which can be represented as the following vectors:

$$\begin{aligned}
\vec{\sigma}_{\underline{0}} &= [0\ \ 0]\\
\vec{\sigma}_{A^C} &= [1\ \ 0]\\
\vec{\sigma}_{A} &= [0\ \ 1]\\
\vec{\sigma}_{\overline{1}} &= [1\ \ 1]
\end{aligned}$$

Elements of Boolean algebra of properties $BA_p(\{A\})$ realized on universe of discourses corresponding sets, whose membership functions are:

$$\begin{aligned}
\underline{0}(x) &= 0\\
A^C(x) &= 1 - A(x)\\
A(x) &= A(x)\\
\overline{1}(x) &= 1
\end{aligned}$$

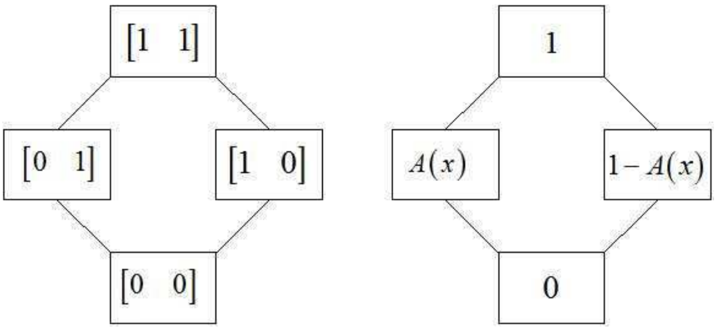


Figure 2: Structures of $BA_p(\{A\})$ elements and membership functions of corresponding sets

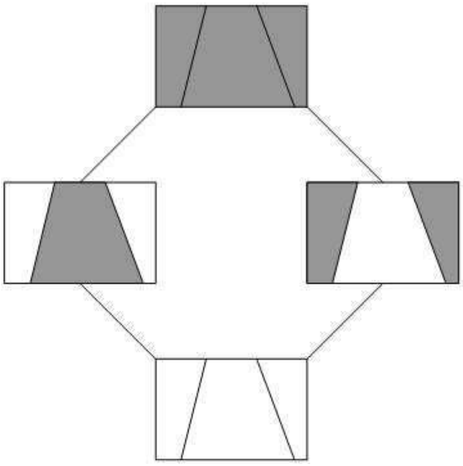


Figure 3: **R**-sets generated by properties - elements of $BA_p(\{A\})$

Examples of ***R***-sets generated by properties as element of Boolean algebra $BA_p(\{A\})$ are given in the following figure:

***Example**: The main characteristics of **R**-sets are illustrated on the example of Boolean lattice of **R**-sets generated by primary **R**-sets $\Omega = \{A, B\}$ represented in fig. 4.*

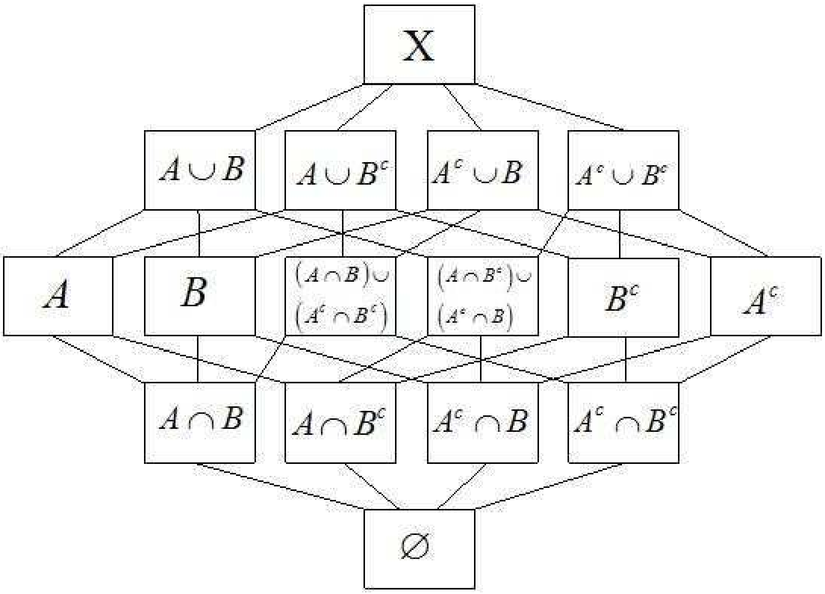


Figure 4: Boolean algebra $\Omega = \{A, B\}$ represented by Hasse diagrams

The corresponding structure vectors

The generalized Boolean polynomials of corresponding sets are

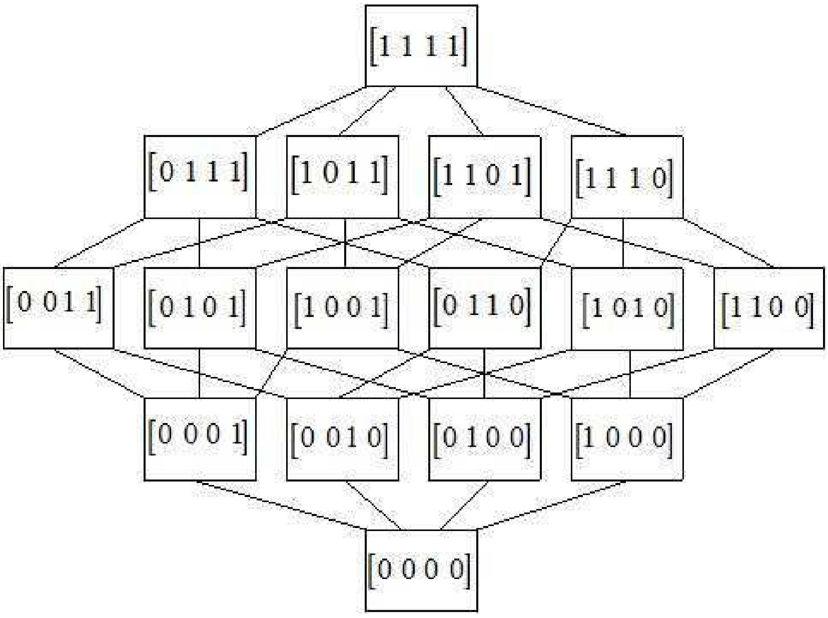


Figure 5: Structure vectors of element of Boolean algebra $\Omega = \{A, B\}$

$$\begin{aligned}
(A \cap B)^{\otimes}(x) &= A(x) \otimes B(x), \\
(A \cap B^{\mathcal{C}})^{\otimes}(x) &= A(x) - A(x) \otimes B(x), \\
(A^{\mathcal{C}} \cap B)^{\otimes}(x) &= B(x) - A(x) \otimes B(x), \\
(A^{\mathcal{C}} \cap B^{\mathcal{C}})^{\otimes}(x) &= 1 - A(x) - B(x) + A(x) \otimes B(x), \\
A(x) &= A(x), \\
B(x) &= B(x) \\
((A \cap B) \cup (A^{\mathcal{C}} \cap B^{\mathcal{C}}))^{\otimes}(x) &= 1 - A(x) - B(x) + 2A(x) \otimes B(x), \\
((A \cap B^{\mathcal{C}}) \cup (A^{\mathcal{C}} \cap B))^{\otimes}(x) &= A(x) + B(x) - 2A(x) \otimes B(x), \\
(B^{\mathcal{C}})^{\otimes}(x) &= 1 - B(x), \\
(A^{\mathcal{C}})^{\otimes}(x) &= 1 - A(x), \\
(A \cup B)^{\otimes}(x) &= A(x) + B(x) - A(x) \otimes B(x), \\
(A^{\mathcal{C}} \cup B)^{\otimes}(x) &= 1 - A(x) + A(x) \otimes B(x), \\
(A \cup B^{\mathcal{C}})^{\otimes}(x) &= 1 - B(x) + A(x) \otimes B(x), \\
(A^{\mathcal{C}} \cup B^{\mathcal{C}})^{\otimes}(x) &= 1 - A(x) \otimes B(x), \\
(A \cap A^{\mathcal{C}})^{\otimes}(x) &= 0, \\
(A \cap A^{\mathcal{C}})^{\otimes}(x) &= 0, \\
(A \cup A^{\mathcal{C}})^{\otimes}(x) &= 1, \\
(A \cup A^{\mathcal{C}})^{\otimes}(x) &= 1.
\end{aligned}$$

Realization of all possible set functions for the given ***R***-sets A and B, in the case when the generalized product is given as ***min*** function is represented in fig. 7.

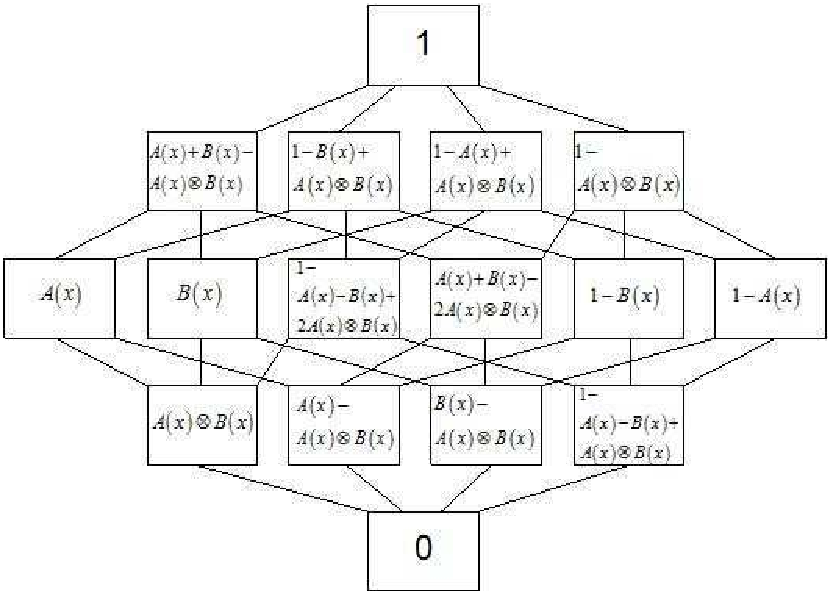


Figure 6: Generalized Boolean polynomials of $\Omega = \{A, B\}$

If we compare the example of ***R***-sets with corresponding classical sets given in fig 8:

It is clear that all properties of classical case are preserved one to one in generalized ***R***-sets case. Actually in the case of ***R***-sets interpretation is richer. In the case of classical sets one object of universe of discourses can be the element of only one atomic set, but in the case of ***R***-sets it can be the member of two and more atomic ***R***-sets but so that sum of corresponding membership function values is equal to 1.

R-partition

R-partition is consistent generalization of classical sets partition. Collection of atomic ***R***-sets $\{A^{\otimes}(s) | S \in P(\Omega)\}$ is ***R***-partition of analyzed universe of discourses X, since:

(a) atomic sets are pair wise mutually exclusive:

$$A^{\otimes}(S_i) \cup A^{\otimes}(S_j) = \begin{cases} A^{\otimes}(S), & i = j \\ \varnothing, & i \neq j \end{cases};$$

or

$$(A^{\otimes}(S_i) \cup A^{\otimes}(S_j))(x) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases};$$

$(x \in X)$.

And (b) they *cover* the universe X:

$$\bigcup_{S \in P(\Omega)} A^{\otimes}(S) = X;$$

or

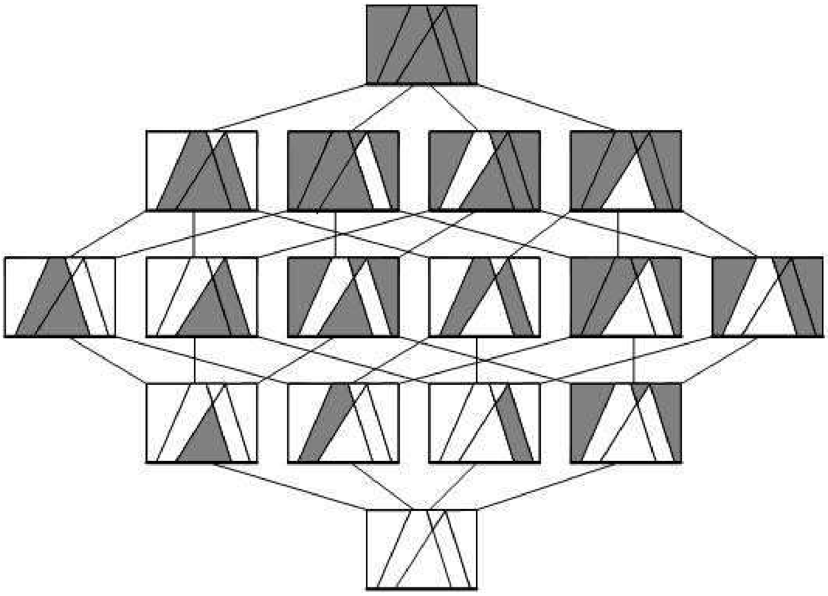


Figure 7: **R**-sets generated by properties - elements of $\Omega = \{A, B\}$ for $\otimes := min$

$$\sum_{S \in P(\Omega)} A^{\otimes}(S)(x) = 1;$$

$(x \in X)$.

Atomic ***R***-sets from the previous example are given in the following figure with GBP as their membership functions and their structures:

In the case of classical sets there is one additional constraint: any element of universe of discourses $x \in X$ belongs to only one classical atomic set $A(S)$, $(S \in P(\Omega))$:

$$A(S_j)(x) = 1 \Rightarrow A(S_j)(x) = 0, \;\; S_j \neq S_i,$$

$(S_i, S_j \in P(\Omega); \;\; x \in X)$.

This constraint is not general and it is not valid in general ***R***-sets case.

It is clear that all properties of the classical set algebra are preserved in the case of ***R***-sets algebra and/or by using ***R***-set approach, based on IBA one can treat gradation in the Boolean frame, contrary to all fuzzy sets approaches.

This is very important for generalization of probability, since additive probability is based on classical sets and as a consequence it is in Boolean frame, generalization of probability based on R-sets (R-probability) is in Boolean frame too, and/or R-probability is Boolean consistent, contrary to conventional fuzzy probabilities. R-probability preserves all value invariant characteristics of classical probability with much richer probability phenomena than in the case of classical probability.

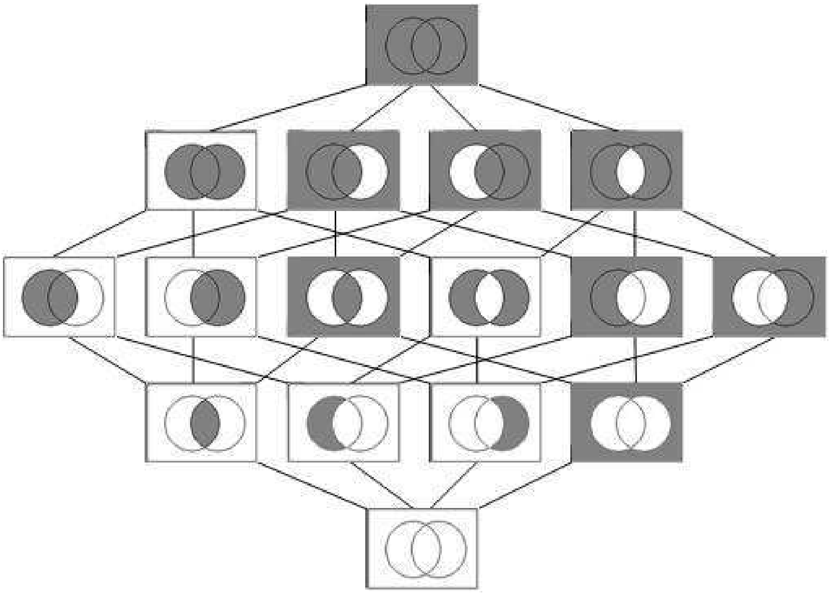


Figure 8: Hasse diagram of classical sets

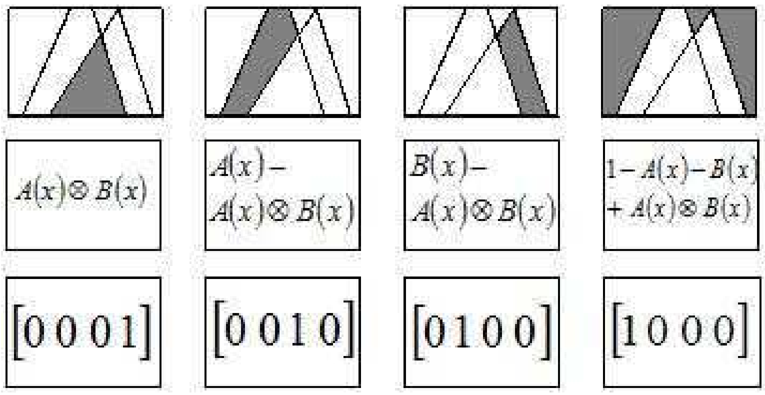

$A(x)\otimes B(x)$
$A(x)-$ $A(x)\otimes B(x)$
$B(x)-$ $A(x)\otimes B(x)$
$1-A(x)-B(x)$ $+A(x)\otimes B(x)$
$[0\,0\,0\,1]$
$[0\,0\,1\,0]$
$[0\,1\,0\,0]$
$[1\,0\,0\,0]$

Figure 9: Atomic $\boldsymbol{R}$-sets for $\otimes := min$

4 Conclusion

Having recognized the constraints of classical mathematical approaches to real problems (for example: complete theory in the domain of automatic control is possible only for linear systems), L. Zadeh concluded that computation by words is very important since natural languages can be used for description of completely new results (for example quant physics) without changing its fundament. Objects of natural language are not as precise as mathematical objects; on the contrary, the number of words in a dictionary should be infinite, what is of course impossible. Non-preciseness immanent to a natural language is formalized by introducing gradation so an analyzed object can possess a property with some intensity or it can be a member of a generalized and/or fuzzy set. To the grammar of a natural language corresponds in mathematics algebra. Conventional approaches to theory of fuzzy sets are far from the original idea of computing with words since one natural language has one grammar and/or one contents invariant fundament, but conventional approaches in fuzzy mathematics offer infinitely many algebras which are used depending on analyzed problem. One cannot imagine a poet who changes the grammar of a natural language dependent on a poem's contents!

The new approach offers as a solution the same algebra - Boolean algebra, both for the classical mathematical black and white view and for real applications with gradation. In the classical case one uses two-valued realization of Boolean algebra but in the generalized case one uses real-valued ([0, 1] - valued) realization of finite Boolean algebra. So, the new approach opens the door to the original Zadeh's idea of computing by words in one natural way. All results based on classical two-valued approaches by applying new approach can be straightaway generalized (theory of sets, logic, relations and their applications).

Classical theories in Boolean frame are based on black-white outlook - theory of sets, logic, theory of relations, probability etc. In the analysis of real problems, in a general case, by using gradation - real-valued realization, one can do much more (efficiency) by much less (complexity) than by a black-white outlook and/or two-valued realization. Consistent Boolean generalization of classical two-valued theory of sets into real-valued set theory means preservation of all value irrelevant - algebraic characteristics (all axioms and theorems of BA). Since classical theory of sets is based on celebrated two-valued realization of BA, consistent generalized theory of real-valued sets has to be based on real-valued realization of BA.

Interpolative Boolean algebra (IBA) [11] is a real-valued and/or [0, 1]-valued realization of BA. By IBA any element of a corresponding BA or any Boolean function is uniquely mapped into a generalized Boolean polynomial (GBP), able to process values from a real unit interval [0, 1], so that by corresponding arithmetic properties it preserves on a value level, all algebraic properties. GBP is figure on value level of corresponding disjunctive canonical form from algebraic level. Disjunctive canonical form of analyzed element of BA is disjunction (union, join) of relevant atoms and its GBP is sum of atomic Boolean polynomial of this atoms. Which atoms are relevant for analyzed element of BA is determined by its structure function (containment). For structures is valid algebraic ***principle of structure functionality:*** *structure of combined element of BA can be calculated on the basis of structures of its components.* Celebrated and very well known truth functionality principle is only figure of structure functionality principle on value level, valid in and only in two-valued realization and/or truth functionality principle is not algebraic principle. As a consequence truth functionality principle can't be base for Boolean consistent MV or real-valued generalizations.

All theories based on classical sets using R-sets can be Boolean consistently generalized. For example, classical theory of probability is based on classical sets and as a consequence it is in Boolean frame. The fact that black-white outlook, immanent to classical sets, is not adequate in many real problems, was motive for introducing fuzzy probability. Fuzzy probability is based on fuzzy sets and as a consequence it is not consistent Boolean generalization of classical probability. R-probability is consistently Boolean generalization of classical probability, based on R-sets.

References

[1] R. G. Boole: The Calculus of Logic, *Cambridge and Dublin Mathematical Journal*, Vol. III, pp. 183-198, 1848.

[2] L. Zadeh: Fuzzy Sets, *Information and Control*, no. 8, pages 338-353. 1965.

[3] L. Zadeh: Bellman R.E., Local and fuzzy logics, *Modern Uses of Multiple-Valued Logic*, J.M. Dunn and G. Epstein (eds.), Dordrecht: D. Reidel, 103-165, 1977.

[4] L. Zadeh: Man and Computer, Bordeaux, France, 130-165, *Outline of a new approach to the analysis of complex systems and decision processe*s, IEEE 1972.

[5] S. Gottwald: *A Treats on Many-Valued Logics*, volume 9 of Studies in Logic and Computation. Research Studies Press, Bladock, 2000.

[6] J. Lukasiewicz: *Selected Works*. (ed.: L. Borkowski), North-Holland Publ. Comp., Amsterdam and PWN, Warsaw, 1970.

[7] P. Hajek: *Metamathematics of Fuzzy Logic, Trends in Logica* - Studia logica library, Kluwer Academic Publishers, Dodrecth /Boston/London, 1998.

[8] D. Radojevic: New [0,1]-valued logic: A natural generalization of Boolean logic, *Yugoslav Journal of Operational Research - YUJOR*, Belgrade, Vol. 10, No 2, 185-216, 2000.

[9] D. Radojevic: Interpolative relations and interpolative preference structures, *Yugoslav Journal of Operational Research - YUJOR* , Belgrade, Vol. 15, No 2, 2005.

[10] D. Radojevic: Logical measure - structure of logical formula, in *Technologies for Constructing Intelligent Systems 2*: Tools, Springer, pp 417-430, 2002.

[11] D. Radojevic: Interpolative realization of Boolean algebra as consistent frame for gradation and/or fuzziness, Studies in Fuzziness and Soft Computing: Forging New Frontiers: Fuzzy Pioneers II, Editors M. Nikravesh, J. Kacprzyk, L. Zadeh, Springer, pp. 295-318, 2007.

[12] R. Sikorski: *Boolean Algebras*, Springer-Verlag, Berlin, New York, 1964.

[13] D. Radojevic: Fuzzy Set Theory in Boolean Frame, *Workshop invited key lecture*, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. III, Suppl. issue: Proceedings of ICCCC 2008, pp. 121-131, 2008.

[14] E. P. Klement, Mesiar R., Pap E.: *Triangular Norms*, Kluwer Academic Publ, Dodrecht, 2000.

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 172-184

Simulating NEPs in a Cluster with jNEP

Emilio del Rosal, Rafael Nuñez, Carlos Castañeda and Alfonso Ortega

Abstract: This paper introduces *jNEP*: a general, flexible, and rigorous implementation of NEPs (the basic model) and some interesting variants; it is specifically designed to easily add the new results (filters, stopping conditions, evolutionary rules, and so on) of the research in the area. jNEP is written in Java; there are two different versions that implement the concurrency of NEPs by means of the Java classes *Process* and *Thread*s respectively. There are also extended versions that run on clusters of computers under JavaParty. *jNEP* reads the description of the currently simulated NEP from a XML configuration file. This paper shows how *jNEP* tackles the SAT problem with polynomial performance by simulating an ANSP.

Keywords: NEPs, natural computing, simulation, clusters of computers

1 Introduction

1.1 NEPs

NEP stands for *Network of Evolutionary Processors*. NEPs are an abstract model of distributed/parallel symbolic processing presented in [1, 12, 2]. NEPs are inspired by biological cells. These are represented by words which describe their DNA sequences. Informally, at any moment of time, the evolutionary system is described by a collection of words, where each word represents one cell. Cells belong to species and their community evolves according to mutations and division which are defined by operations on words. Only those cells are accepted as surviving (correct) ones which are represented by a word in a given set of words, called the genotype space of the species. This feature parallels the natural process of evolution. Each node in the net is a very simple processor containing words which performs a few elementary tasks to alter the words, send and receive them to/from other processors. Despite the simplicity of each processor, the entire net can carry out very complex tasks efficiently. Many different works demonstrate the computational completeness of NEPs [4][10] and their ability to solve NP problems with linear or polynomial resources [11][2]. The emergence of such a computational power from very simple units acting in parallel is one of the main interests of NEPs.

NEPs can be used to accept families of languages. When they are used in this way they are called Accepting NEPs (ANEPs). Several variants of NEPs have been proposed in the scientific literature. NEP (the original model) [2], hybrid nets of evolutionary processor (HNEP) [4] and nets of splicing processors NEPS or NSP [10]. This last model uses a splicing processor, which adds a new operation (splicing rules) to mimic crossover in genetic systems. In section 3.1 we show an example of ANSP (the accepting variant of NSPs) solving the SAT problem. Nevertheless, all of them share the same general characteristics.

A NEP is built from the following elements:

a) A set of symbols which constitutes the alphabet of the words which are manipulated by the processors.

b) A set of processors.

c) An underlying graph where each vertex represents a processor and the edges determine which processors are connected so they can exchange words.

d) An initial configuration defining which words are in each processor at the beginning of the computation.

e) One or more stopping rules to halt the NEP.

An evolutionary processor has three main components:

a) A set of evolutionary rules to modify its words.

b) An input filter that specifies which words can be received from other processors.

c) An output filter that delimits which words can leave the processor to be sent to others.

The variants of NEPs mainly differ in their evolutionary rules and filters. They perform very simple operations, like altering the words by replacing all the occurrences of a symbol by another, or filtering those words whose alphabet is included in a given set of words.

NEP's computation alternates evolutionary and communication steps: an evolutionary step is always followed by a communication step and vice versa. Computation follows the following scheme: when the computation starts, every processor has a set of initial words. At first, an evolutionary step is performed: the rules in each processor modify the words in the same processor. Next, a communication step forces some words to leave their processors and also forces the processors to receive words from the net. The communication step depends on the constraints imposed by the connections and the output and input filters. The model assumes that an arbitrary number of copies of each word exists in the processors, therefore all the rules applicable to a word are actually applied, resulting in a new word for each rule. The NEP stops when one of the stopping conditions is met, for example, when the set of words in a specific processor (the output node of the net) is not empty. A detailed formal description of NEPs can be found in [1], [4] or [10]. For example, the following definition is literally taken from [1] and will help the reader to understand the model.

A network of evolutionary processors (NEP for short) of size n is a construct $\Gamma = (V, N_1, N_2, ..., N_n, G)$, where V is an alphabet and for each $1 \leq i \leq n, N_i = (M_i, A_i, PI_i, PO_i)$ is the i-th evolutionary node processor of the network. The parameters of every processor are:

- M_i is a finite set of evolution rules of one of the following forms only
  - $a \to b, a, b \in V$ (substitution rules),
  - $a \to \lambda, \in V$ (deletion rules),
  - $\lambda \to a, a \in V$ (insertion rules),

  More clearly, the set of evolution rules of any processor contains either substitution or deletion or insertion rules.

- A_i is a finite set of strings over V . The set A_i is the set of initial strings in the i-th node. Actually, we consider that each string appearing in any node at any step has an arbitrarily large number of copies in that node.

- PI_i and PO_i are subsets of V^* representing the input and the output filter, respectively. These filters are defined by the membership condition, namely a string $w \in V^*$ can pass the input filter (the output filter) if $w \in PI_i(w \in PO_i)$.

Finally, $G = (N_1, N_2, ..., N_n, E)$ is an undirected graph called the underlying graph of the network. The edges of G, that is the elements of E, are given in the form of sets of two nodes. The complete graph with n vertices is denoted by K_n.

1.2 Clusters of computers

Running NEPs simulators on cluster is one of the possible ways of exploiting the inherent parallel nature of NEPs. The Java Virtual Machine (JVM), which can be considered the standard Java, cannot be run on clusters.

Several attempts have tried to overcome this limitation, for example: Java-Enabled Single-System-Image Computing Architecture 2 (JESSICA2) [8], the cluster virtual machine for Java developed by IBM (IBM cJVM) [3], Proactive PDC [15], DO! [9], JavaParty [6], and Jcluster [7].

The simulator described in this paper has been developed with both JVM and JavaParty.

2 jNEP

A lot of research effort has been devoted to the definition of different families of NEPs and to the study of their formal properties, such as their computational completeness and their ability to solve NP problems with polynomial performance. However, no relevant effort, apart from [5], has tried to develop a NEP simulator or any kind of implementation. Unfortunately, the software described in this reference gives the possibility of using only one kind of rules and filters and, what is more important, violates two of the main principles of the model: 1) NEP's computation should not be deterministic and 2) evolutionary and communication steps should alternate strictly. Indeed, the software is focused in solving decision problems in a parallel way, rather than simulating the NEP model with all its details.

jNEP tries to fill this gap in the literature. It is a program written in Java which is capable of simulating almost any NEP in the literature. In order to be a valuable tool for the scientific community, it has been developed under the following principles:

a) It rigorously complies with the formal definitions found in the literature.

b) It serves as a general tool, by allowing the use of the different NEP variants and is ready to adapt to future possible variants, as the research in the area advances.

c) It exploits as much as possible the inherent parallel/distributed nature of NEPs.

The jNEP code is freely available in http://jnep.e-delrosal.net.

2.1 jNEP design

jNEP offers an implementation of NEPs as general, flexible and rigorous as has been described in the previous paragraphs. As shown in figure 1, the design of the NEP class mimics the NEP model definition. In jNEP, a NEP is composed of evolutionary processors and an underlying graph (attribute *edges*) to define the net topology and the allowed inter processor interactions. The *NEP* class coordinates the main dynamic of the computation and rules the processors (instances of the *EvolutionaryProcessor* class), forcing them to perform alternate evolutionary and communication steps. It also stops the computation when needed. The core of the model includes these two classes, together with the *Word* class, which handles the manipulation of words and their symbols.

We keep *jNEP* as general and rigorous as possible by means of the following mechanisms: Java interfaces and the develop of different versions to widely exploit the parallelism available in the hardware platform.

jNEP offers three interfaces:

a) *StoppingCondition*, which provides the method *stop* to determine whether a *NEP* object should stop according to its state.

b) *Filter*, whose method *applyFilter* determines which objects of class *Word* can pass it.

c) *EvolutionaryRule*, which applies a *Rule* to a set of *Word*s to get a new set.

jNEP tries to implement a wide set of NEPs' features.
The *jNEP user guide* (http://jnep.e-delrosal.net) contains the updated list of filters, evolutionary rules and stopping conditions implemented.

Currently *jNEP* has two list of choices to select the parallel/distributed platform on which it runs (any combination of them is also available in http://jnep.e-delrosal.net). Concurrency is implemented by means of two different Java approaches: *Thread*s and *Process*es. The first needs more complex synchronization mechanisms. The second uses heavier concurrent threads. The supported platforms are standard JVM and clusters of computers (by means of JavaParty).

More precisely, in the case of the *Process*es option each processor in the net is actually an independent program in the operating system. The communication between nodes is carried out through the standard input/output streams of the program. The class NEP has access to those streams and coordinates the nodes. The mandatory alternation of communication and evolutionary steps in the computations of NEPs greatly eases their synchronization and communication. The following protocol has been followed for the communication step:

1 NEP class sends the message to communicate to every node in the graph. Then it waits for their responses.

2 Every node finishes its communication step after sending to the net the words that pass their outputs filters. Then, they indicate to the NEP class that they have finished the communication step.

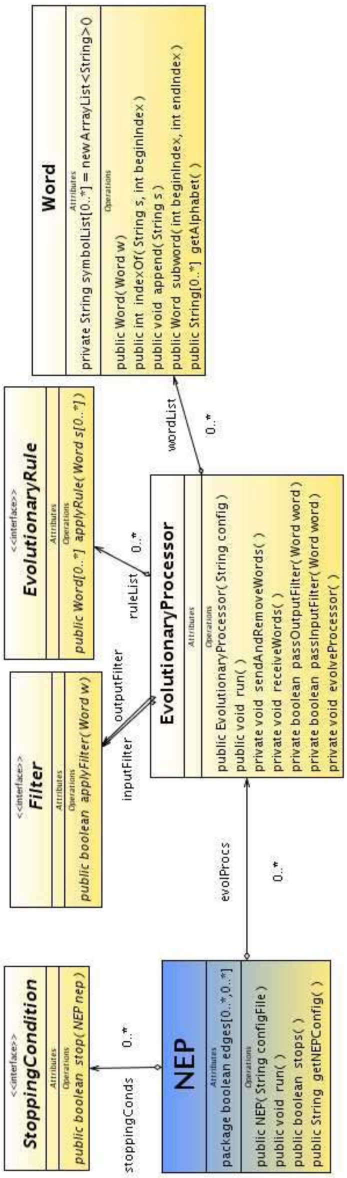

Word
Attributes
private String symbolList[0..*] = new ArrayList<String>0
Operations
public Word(Word w)
public int indexOf(String s, int beginIndex)
public void append(String s)
public Word subword(int beginIndex, int endIndex)
public String[0..*] getAlphabet( )
<<interface>>
EvolutionaryRule
Attributes
Operations
public Word[0..*] applyRule(Word s[0..*])
<<interface>>
Filter
Attributes
Operations
public boolean applyFilter(Word w)
EvolutionaryProcessor
Attributes
Operations
public EvolutionaryProcessor(String config)
public void run( )
private void sendAndRemoveWords( )
private void receiveWords( )
private boolean passOutputFilter(Word word)
private boolean passInputFilter(Word word)
private void evolveProcessor( )
<<interface>>
StoppingCondition
Attributes
Operations
public boolean stop(NEP nep)
NEP
Attributes
package boolean edges[0..*,0..*]
Operations
public NEP(String configFile)
public void run( )
public boolean stops( )
public String getNEPConfig( )
wordList
0..*
ruleList
0..*
outputFilter
inputFilter
evolProcs
0..*
stoppingConds
0..*

Figure 1: Simplified class diagram of jNEP

3 The NEP class moves all the words from the net to the input filters of the corresponding nodes.

The evolutionary step is synchronized by means of an initial message sent by the NEP class to make all the nodes evolve. Afterwards, the NEP class waits until all the nodes finish.

The implementation with *Java Thread*s has other implications. In this option, each processor is an object of the *Java Thread* class. Thus, each processor execute its tasks in parallel as independent execution lines, although all of them belong to the same program. Data exchange between them is performed by direct access to memory. The principles of communication and coordination are the same as in the previous option. The main difference is that, instead of waiting for all the streams to finish or to send a certain message, *Thread*s are coordinated by means of basic concurrent programming mechanisms as semaphores, monitors, etc.

In conclusion, jNEP is a very flexible tool that can run in many different environments. Depending on the operating system, the Java Virtual Machine used and the concurrency option chosen, jNEP will work in a slightly different manner. The user should select the best combination for his needs.

3 jNEP in practice

jNEP is written in Java, therefore to run jNEP one needs a Java virtual machine (version 1.4.2 or above) installed in a computer. Then one has to write a configuration file describing the NEP. The *jNEP user guide* (available at http://jnep.e-delrosal.net) contains the details concerning the commands and requirements needed to launch jNEP. In this section, we want to focus on the configuration file which has to be written before running the program, since it has some complex aspects important to be aware of the potentials and possibilities of jNEP.

The configuration file is an XML file specifying all the features of the NEP. Its syntax is described below in BNF format, together with a few explanations. Since BNF grammars are not capable of expressing context-dependent aspects, context-dependent features are not described here. Most of them have been explained informally in the previous sections. Note that the traditional characters `<>` used to identify non-terminals in BNF have been replaced by `[]` to avoid confusion with the use of the `<>` characters in the XML format.

- [configFile] ::= ¡?xml version=”1.0”?¿ ¡NEP nodes=“[integer]”¿ [alphabetTag] [graphTag] [processorsTag] [stoppingConditionsTag] ¡/NEP¿

- [alphabetTag] ::= ¡ALPHABET symbols=“[symbolList]”/¿

- [graphTag] ::= ¡GRAPH¿ [edge] ¡/GRAPH¿

- [edge] ::= ¡EDGE vertex1=“[integer]” vertex2=“[integer]”/¿ [edge]

- [edge] ::= λ

- [processorsTag] ::= ¡EVOLUTIONARY.PROCESSORS¿ [nodeTag] ¡/EVOLUTIONARY.PROCESSORS¿

The above rules show the main structure of the NEP: the alphabet, the graph (specified through its vertices) and the processors. It is worth remembering that each processor is identified implicitly by its position in the processors tag (first one is number 0, second is number 1, and so on).

- [stoppingConditionsTag] ::= ¡STOPPING_CONDITION¿ [conditionTag] ¡/STOPPING_CONDITION¿
- [conditionTag] ::= ¡CONDITION type=“MaximumStepsStoppingCondition” maximum=“[integer]”/¿ [conditionTag]
- [conditionTag] ::= ¡CONDITION type=“WordsDisappearStoppingCondition” words=“[wordList]”/¿ [conditionTag]
- [conditionTag] ::= ¡CONDITION type=“ConsecutiveConfigStoppingCondition”/¿ [conditionTag]
- [conditionTag] ::= ¡CONDITION type=“NonEmptyNodeStoppingCondition” nodeID=“[integer]”/¿ [conditionTag]
- [conditionTag] ::= λ

The syntax of the stopping conditions shows that a NEP can have several stopping conditions. The first one which is met causes the NEP to stop. The different types try to cover most of the stopping conditions used in the literature. If needed, more of them can be added to the system easily.

At this moment jNEP supports 4 stopping conditions, the *jNEP user guide* explains their semantics in detail:

1. **ConsecutiveConfigStoppingCondition**: It produces the NEP to stop if two consecutive configurations are found as communication and evolutionary steps are performed.
2. **MaximumStepsStoppingCondition**: It produces the NEP to stop after a maximum number of steps.
3. **WordsDisappearStoppingCondition**: It produces the NEP to stop if none of the words specified are in the NEP. It is useful for generative NEPs where the lack of non-terminals means that the computation have reached its goal.
4. **NonEmptyNodeStoppingCondition**: It produces the NEP to stop if one of the nodes is non-empty. Useful for NEPs with an output node.

- [nodeTag] ::= ¡NODE initCond=”[wordList]” [auxWordList]¿ [evolutionaryRulesTag] [nodeFiltersTag] ¡/NODE¿ [nodeTag]
- [nodeTag] ::= λ
- [auxWordList] ::= λ — auxiliaryWords=”[wordList]”
- [evolutionaryRulesTag] ::= ¡EVOLUTIONARY_RULES¿ [ruleTag] ¡/EVOLUTIONARY_RULES¿
- [ruleTag] ::= ¡RULE ruleType=“[ruleType]” actionType=“[actionType]” symbol=“[symbol]” newSymbol=“[symbol]”/¿ [ruleTag]
- [ruleTag] ::= ¡RULE ruleType=”splicing” wordX=“[symbolList]” wordY=“[symbolList]” wordU= “[symbolList]” wordV=“[symbolList]”/¿ [ruleTag]
- [ruleTag] ::= ¡RULE ruleType=”splicingChoudhary” wordX=“[symbolList]” wordY=“[symbolList]” wordU=“[symbolList]” wordV=“[symbolList]”/¿ [ruleTag]
- [ruleTag] ::= λ
- [ruleType] ::= insertion — deletion — substitution
- [actionType] ::= LEFT — RIGHT — ANY

- [nodeFiltersTag] ::= [inputFilterTag] [outputFilterTag]
- [nodeFiltersTag] ::= [inputFilterTag]
- [nodeFiltersTag] ::= [outputFilterTag]
- [nodeFiltersTag] ::= λ
- [inputFilterTag] ::= ¡INPUT [filterSpec]/¿
- [outputFilterTag] ::= ¡OUTPUT [filterSpec]/¿
- [filterSpec] ::= type=[filterType] permittingContext="[symbolList]" forbiddingContext="[symbolList]"
- [filterSpec] ::= type="SetMembershipFilter" wordSet="[wordList]"
- [filterSpec] ::= type="RegularLangMembershipFilter" regularExpression="[regExpression]"
- [filterType] ::= 1 — 2 — 3 — 4

Above, we describe the elements of the processors: their initial conditions, rules, and filters. jNEP treats rules with the same philosophy as in the case of stopping conditions, which means that our systems supports almost all kinds found in the literature at the moment and, more important, future types can also be added.

jNEP can work with any of the rules found in the original model [1, 12, 2]. Moreover, we support splicing rules, which are needed to simulate a derivation of the original model presented in [14] and [10]. The two splicing rule types are slightly different. It is important to note that if you use Manea's splicing rules, you may need to create an auxiliary word set for those processor with splicing rules.

With respect to filters, jNEP is prepared to simulate nodes with filters based on random context conditions. To be more specific, any of the four filter types traditionally used in the literature since [13]. Besides, jNEP is capable of creating filters based in membership conditions. A few works use them, for instance [1]. They are in some way non-standard and could be defined as follows:

1. **SetMembershipFilter**: It permits to pass only words that are included in a specific set.

2. **RegularLangMembershipFilter**: This filter contains a regular language to which words need to belong. The language have to be defined as a Java regular expression.

We will finish the explanation of the grammar for our xml files with the rules needed to describe some of the pending non-terminals. They are typical constructs for lists of words, list of symbols, boolean and integer data and regular expressions.

- [wordList] ::= [symbolList] [wordList]
- [wordList] ::= λ
- [symbolList] ::= `a string of symbols separated by the character '\_'`
- [boolean] ::= true — false
- [integer] ::= `an integer number`
- [regExpression] ::= `a Java regular expression`

The reader may refer to the *jNEP user guide* for further detailed information.

3.1 An example: solving the SAP problem with linear resources

Reference [10] describes a NEP with splicing rules (ANSP) which solves the boolean satisfiability problem (SAT) with linear resources, in terms of the complexity classes also present in [10].

ANSP stands for Accepting Networks of Splicing Processors. In sort, a ANSP is a NEP where the transformation rules of its nodes are *splicing rules*. The transformation performed by those rules is very similar to the genetic crossover. To be more precise, a *splicing rule* σ is a quadruple of words written as $\sigma = [(x, y); (u, v)]$. Given this *splicing rule* σ and two words (w,z), the action of σ on (w,z) is defined as follows:

$$\sigma(w, z) = \{t \mid w = \alpha xy\beta, z = \gamma uv\delta \text{ for any words } \alpha, \beta, \gamma, \delta \text{ and } t = \alpha xv\delta \text{ or } t = \gamma uy\beta\}$$

We can use jNEP to actually build and run the ANSP that solves the boolean satisfiability problem (SAT). We will see how the features of NEPs and the *splicing rules* can be used to tackle this problem. The following is a broad summary of the configuration file for such a ANSP, applied to the solution of the SAT problem for three variables. The entire file can be downloaded from jnep.e-delrosal.net.

```
<NEP nodes="9">
  <ALPHABET symbols="A_B_C_!A_!B_!C_AND_OR_(_)_[A=1]_[B=1]_[C=1]_[A=0]_[B=0]_[C=0]_#_UP_{_}_1"/>
  <!-- WE IGNORE THE GRAPH TAG TO SAVE SPACE. THIS NEP HAVE A COMPLETE GRAPH -->
  <STOPPING_CONDITION>
    <CONDITION type="NonEmptyNodeStoppingCondition" nodeID="1"/>
  </STOPPING_CONDITION>
  <EVOLUTIONARY_PROCESSORS>
    <!-- INPUT NODE -->
    <NODE initCond="{_(_A_)_AND_(_B_OR_C_)_}" auxiliaryWords="{_[A=1]_# {_[A=0]_# {_[B=1]_#
                                                              {_[B=0]_# {_[C=1]_# {_[C=0]_#">
      <EVOLUTIONARY_RULES>
        <RULE ruleType="splicing" wordX="{" wordY="(" wordU="{_[A=1]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="(" wordU="{_[A=0]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[A=0]" wordU="{_[B=0]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[A=0]" wordU="{_[B=1]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[A=1]" wordU="{_[B=0]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[A=1]" wordU="{_[B=1]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[B=0]" wordU="{_[C=0]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[B=0]" wordU="{_[C=1]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[B=1]" wordU="{_[C=0]" wordV="#"/>
        <RULE ruleType="splicing" wordX="{" wordY="[B=1]" wordU="{_[C=1]" wordV="#"/>
      </EVOLUTIONARY_RULES>
      <FILTERS>
        <INPUT type="4" permittingContext=""
                        forbiddingContext="[A=1]_[B=1]_[C=1]_[A=0]_[B=0]_[C=0]_#_UP_{_}_1"/>
        <OUTPUT type="4" permittingContext="[C=1]_[C=0]" forbiddingContext=""/>
      </FILTERS>
    </NODE>
    <!-- OUTPUT NODE -->
    <NODE initCond="">
      <EVOLUTIONARY_RULES>
      </EVOLUTIONARY_RULES>
      <FILTERS>
        <INPUT type="1" permittingContext="" forbiddingContext="A_B_C_!A_!B_!C_AND_OR_(_)"/>
        <OUTPUT type="1" permittingContext=""
                         forbiddingContext="[A=1]_[B=1]_[C=1]_[A=0]_[B=0]_[C=0]_#_UP_{_}_1"/>
      </FILTERS>
    </NODE>
    <!-- COMP NODE -->
    <NODE initCond="" auxiliaryWords="#_[A=0]_} #_[A=1]_} #_} #_1_)_}">
      <EVOLUTIONARY_RULES>
        <RULE ruleType="splicing" wordX="" wordY="A_OR_1_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="!A_OR_1_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="B_OR_1_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="!B_OR_1_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="C_OR_1_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="!C_OR_1_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="AND_(_1_)_}" wordU="#" wordV="}"/>
        <RULE ruleType="splicing" wordX="" wordY="[A=1]_(_1_)_}" wordU="#" wordV="[A=1]_}"/>
        <RULE ruleType="splicing" wordX="" wordY="[A=0]_(_1_)_}" wordU="#" wordV="[A=0]_}"/>
      </EVOLUTIONARY_RULES>
      <FILTERS>
        <INPUT type="1" permittingContext="1" forbiddingContext=""/>
        <OUTPUT type="1" permittingContext="" forbiddingContext="#_1"/>
      </FILTERS>
    </NODE>
    <!-- A=1 NODE -->
    <NODE initCond=""  auxiliaryWords="#_1_)_} #_)_}">
      <EVOLUTIONARY_RULES>
        <RULE ruleType="splicing" wordX="" wordY="A_)_}" wordU="#" wordV="1_)_}"/>
        <RULE ruleType="splicing" wordX="" wordY="(_!A_)_}" wordU="#" wordV="UP"/>
        <RULE ruleType="splicing" wordX="" wordY="OR_!A_)_}" wordU="#" wordV=")_}"/>
        <RULE ruleType="splicing" wordX="" wordY="B_)_}" wordU="#" wordV="UP"/>
        <RULE ruleType="splicing" wordX="" wordY="C_)_}" wordU="#" wordV="UP"/>
      </EVOLUTIONARY_RULES>
      <FILTERS>
        <INPUT type="1" permittingContext="[A=1]" forbiddingContext="[A=0]_1"/>
        <OUTPUT type="1" permittingContext="" forbiddingContext="#_UP"/>
      </FILTERS>
    </NODE>
    <!-- A=0 NODE -->
    <NODE initCond=""  auxiliaryWords="#_1_)_} #_)_}">
      <EVOLUTIONARY_RULES>
        <RULE ruleType="splicing" wordX="" wordY="OR_A_)_}" wordU="#" wordV=")_}"/>
        <RULE ruleType="splicing" wordX="" wordY="(_A_)_}" wordU="#" wordV="UP"/>
        <RULE ruleType="splicing" wordX="" wordY="!A_)_}" wordU="#" wordV="1"/>
        <RULE ruleType="splicing" wordX="" wordY="B_)_}" wordU="#" wordV="UP"/>
        <RULE ruleType="splicing" wordX="" wordY="C_)_}" wordU="#" wordV="UP"/>
      </EVOLUTIONARY_RULES>
      <FILTERS>
        <INPUT type="1" permittingContext="[A=0]" forbiddingContext="[A=1]_1"/>
        <OUTPUT type="1" permittingContext="" forbiddingContext="#_UP"/>
      </FILTERS>
    </NODE>
    <!-- NODES FOR 'B' AND 'C' ARE ANALOGOUS TO THOSE FOR 'A'. WE DO NOT PRESENT THEM TO SAVE SPACE-->
  </EVOLUTIONARY_PROCESSORS>
</NEP>
```

With this configuration file, at the end of its computation, jNEP outputs the interpretation which satisfies the logical formula contained in the file, namely:

```
(_A_)_AND_(_B_OR_C_): {_[C=0]_[B=1]_[A=1]_}{_[C=1]_[B=1]_[A=1]_}{_[C=1]_[B=0]_[A=1]_}
```

This ANSP is able to solve any formula with three variables. The formula to be solved must be specified as the value of the *initCond* attribute for the input node.

```
***************                NEP INITIAL CONFIGURATION                ***************
--- Evolutionary Processor 0 ---
{_(_A_)_AND_(_B_OR_C_)_}
```

Our ANSP works as follows. Firstly, the first node creates all the possible combinations for the 3 variables values. We show below the jNEP output for the first step:

```
*************** NEP CONFIGURATION - EVOLUTIONARY STEP - TOTAL STEPS: 1 ***************
```

```
--- Evolutionary Processor 0 ---
{_# {_[A=1]_(_A_)_AND_(_B_OR_C_)_}
{_[A=0]_(_A_)_AND_(_B_OR_C_)_}
```

the splicing rules of the initial node has appended the two possible values of A to two copies of the logical formula. The concerning rules are:

```
 <RULE ruleType="splicing" wordX="{"
wordY="(" wordU="{_[A=1]" wordV="#"/> <RULE ruleType="splicing"
wordX="{" wordY="(" wordU="{_[A=0]" wordV="#"/>
```

This kind of rules (Manea's splicing rules) uses some auxiliary words that are never removed from the nodes. In our ANSP we use the following auxiliary words:

```
 auxiliaryWords="{_[A=1]_# {_[A=0]_#
{_[B=1]_# {_[B=0]_# {_[C=1]_# {_[C=0]_#"
```

The end of this first stage arise after $2n-1$ steps, where n is the number of variables:

```
--- Evolutionary Processor 0 ---
{_# {_[C=0]_[B=0]_[A=0]_(_A_)_AND_(_B_OR_C_)_}
{_[C=0]_[B=0]_[A=1]_(_A_)_AND_(_B_OR_C_)_}
{_[C=1]_[B=0]_[A=0]_(_A_)_AND_(_B_OR_C_)_}
{_[C=1]_[B=0]_[A=1]_(_A_)_AND_(_B_OR_C_)_}
{_[C=0]_[B=1]_[A=0]_(_A_)_AND_(_B_OR_C_)_}
{_[C=0]_[B=1]_[A=1]_(_A_)_AND_(_B_OR_C_)_}
{_[C=1]_[B=1]_[A=0]_(_A_)_AND_(_B_OR_C_)_}
{_[C=1]_[B=1]_[A=1]_(_A_)_AND_(_B_OR_C_)_}
```

We would like to remark that NEPs take advantage of the possibility of applying all the rules to one word in the same step. This is because the model states that each word has in an arbitrary number of copies in its processor. Therefore, the above task (which is $\Theta(2^n)$) can be completed in n steps, since each step double the number of words by including to each word a new variable with the value 1 or 0.

After this first stage, the words can leave the initial node and travel to the rest of the nodes. In the net, there is one node per variable and value, in other words, there is one node for $A = 1$, another for $C = 0$ and so on. Each of this node reduces, from right to left, the word formula according to the variable values. For example, the sixth node is the responsible for $C = 1$ and, thus, makes the following modification to the word `{_[C=1]_[B=1]_[A=1]_(_A_)_AND_(_B_OR_C_)_}`:

`{_[C=1]_[B=1]_[A=1]_(_A_)_AND_(_B_OR_C_)_}` $\Longrightarrow$

`{_[C=1]_[B=1]_[A=1]_(_A_)_AND_(_B_OR_1_)_}`

However, the ninth node is responsible for $C = 0$ and, therefore, produces the following change:

`{_[C=0]_[B=1]_[A=1]_(_A_)_AND_(_B_OR_C_)_}` $\Longrightarrow$

`{_[C=0]_[B=1]_[A=1]_(_A_)_AND_(_B_)_}`

In this way, the nodes share the results of their modifications until one of them produces a word where the formula is empty and, thus, it only contains the left side with the variable values. This kind of words is allowed to pass the input filter of the output node, therefore, they will enter it. At this point the NEP halts, since the stopping condition of the NEP says that a non-empty output node is the signal to stop the computation.

For more details we refer to [10] and the implementation in jnep.e-delrosal.net.

4 Conclusions and further research lines

jNEP is one of the first and more complete implementations of the family of abstract computing devices called NEPs.

jNEP simulates not only the basic model, but also some of its variants, and is able to run on clusters of computers.

We would like to remark that NEP performance is improved thanks to the following decisions:

- Each processor contains as much copies as it needs of its strings without any additional restriction.
- All these words are simultaneously modified by the rules of the processors in the same step.
- All the processors in the net perform simultaneously their steps, that is, all the communication steps are done at the same time as well as all the evolutionary steps.

These kind of decisions (inherent parallelism and an unrestricted amount of available memory) are frequent in the so called natural computing devices and usually are needed to get polynomial performance for NP problems.

In the future we plan to offer full access to the cluster version by means of the web. We also plan to develop a graphic user interface to ease the definition of the NEP being simulated and a graphic module to show the evolution of the computations.

jNEP will be used as a module in the design of an automatic programming methodology to design NEPs to solve a given problem.

Acknowledgment: This work was supported in part by the Spanish Ministry of Education and Science (MEC) under Project TSI2005-08225-C07-06 and by the Comunidad Autónoma de Madrid and the Universidad Autónoma de Madrid under Project CCG08-UAM/TIC-4425.

References

[1] J. Castellanos, C. Martin-Vide, V. Mitrana, and J. M. Sempere. “Networks of evolutionary processors”. *Acta Informatica*, 39(6-7):517-529, 2003.

[2] Juan Castellanos, Carlos Martin-Vide, Victor Mitrana, and Jose M. Sempere. “Solving NP-Complete Problems With Networks of Evolutionary Processors.” *Connectionist Models of Neurons, Learning Processes and Artificial Intelligence : 6th International Work-Conference*

on Artificial and Natural Neural Networks, IWANN 2001 Granada, Spain, June 13-15, Proceedings, Part I, 2001.

[3] http://www.haifa.il.ibm.com/projects/systems/cjvm/index.html

[4] E. Csuhaj-Varju, C. Martin-Vide, and V. Mitrana. "Hybrid networks of evolutionary processors are computationally complete." *Acta Informatica*, 41(4-5):257-272, 2005.

[5] M. A. Diaz, N. Gomez Blas, E. Santos Menendez, R. Gonzalo, and F. Gisbert. "Networks of evolutionary processors (nep) as decision support systems." In Fith International Conference. *Information Research and Applications*, volume 1, pages 192-203. ETHIA, 2007.

[6] http://wwwipd.ira.uka.de/JavaParty/

[7] http://vip.6to23.com/jcluster/

[8] http://i.cs.hku.hk/ wzzhu/jessica2/index.php

[9] Pascale Launay, Jean-Louis Pazat. "A Framework for Parallel Programming in Java." *INRIA Rapport de Recherche* Publication Internet - 1154 decembre 1997 - 13 pages

[10] Florin Manea, Carlos Martin-Vide, and Victor Mitrana. "Accepting networks of splicing processors: Complexity results." *Theoretical Computer Science*, 371(1-2):72-82, February 2007.

[11] Florin Manea, Carlos Martin-Vide, and Victor Mitrana. "All np-problems can be solved in polynomial time by accepting networks of splicing processors of constant size." *DNA Computing*, pages 47-57, 2006.

[12] C. Martin-Vide, V. Mitrana, M. J. Perez-Jimenez, and F. Sancho-Caparrini. "Hybrid networks of evolutionary processors." *Genetic and Evolutionary Computation.* GECCO 2003, PT I, Proceedings, 2723:401-412, 2003.

[13] C. Martin-Vide and V. Mitrana. "Solving 3CNF-SAT and HPP in linear time using WWW." *MACHINES, COMPUTATIONS, AND UNIVERSALITY.* 3354:269-280, 2005.

[14] Choudhary, A. and Krithivasan, K. "Network of evolutionary processors with splicing rules." *MECHANISMS, SYMBOLS AND MODELS UNDERLYING COGNITION, PT 1, PROCEEDINGS*, 3561:290-299, 2005.

[15] http://www-sop.inria.fr/sloop/javall/

Emilio del Rosal[1,2], Rafael Nuñez[2], Carlos Castañeda[1] and Alfonso Ortega[1]
[1]Departamento de Ingeniería Informática, Universidad Autónoma de Madrid
[2]Escuela Politécnica Superior, Universidad San Pablo CEU
E-mail: emilio.delrosal@uam.es

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 185-195

Soft Computing for Intelligent Systems

Imre J. Rudas and János Fodor

Abstract: Intelligent systems are developed for handling problems in which algorithmic solutions require "almost infinite" amount of computation, or even more: there exist no algorithms for solving them. An overview of features and constituents of intelligent systems are given, with special emphasis on soft computing techniques and computational intelligence. Essential features of fuzzy logic, neural networks and evolutionary computing are briefly summarized. Characteristics of hybrid systems are also touched upon.

Keywords: intelligent systems, soft computing, computational intelligence, hybrid systems.

1 Introduction

Engineering and science problems may be divided into two main categories: is *intelligence* required to solve them or not? Some problems can be solved by using numerical algorithms; the theory behind is known, necessary equations can be formulated. Such problems may require high-performance computing, but no intelligence: just press the key and wait for the answer. On the other hand, some problems may be easily formulated, but all algorithms solving them need almost infinite amount of computations to solve them. Even more: other problems have no algorithms at all.

Problems where solution requires intelligence include *perception* (e.g. recognition of signals, phoneme recognition, olfactory signals - first step in robotics), *computer vision* problems (e.g. face recognition), hand-written *character recognition*, *control* in robotics and nonlinear complex systems), *medical applications* (diagnosis, interpretation of medical images and biomedical signals), *playing complex games* (like go or strategic war games), *natural language* analysis problems (understanding of meaning of sentences).

The main aim of the present paper is to give an overview of diverse features and constituents of intelligent systems. After highlighting the notion of computational intelligence and its relationship to artificial intelligence, we go on with soft computing, and hybrid systems close the paper. We also refer to our paper [18].

2 Intelligent Systems

Intelligent systems (IS) provide a standardized methodological approach to solve important and fairly complex problems and obtain consistent and reliable results over time [3]. Extracting from diverse dictionaries, *intelligence* means the ability to comprehend; to understand and profit from experience. There are, of course, other meanings such as ability to acquire and retain knowledge; mental ability; the ability to respond quickly and successfully to a new situation; etc.

The definition of intelligent systems is a difficult problem and is subject to a great deal of debate. From the perspective of computation, the intelligence of a system can be characterized by its flexibility, adaptability, memory, learning, temporal dynamics, reasoning, and the ability to manage uncertain and imprecise information [11].

Independently from the definition, there is not much doubt that artificial intelligence (AI) is an essential basis for building intelligent systems. According to [19], AI consists of two main directions. One is *humanistic AI* (HAI) that studies machines that think and act like humans. The other one is *rationalistic AI* (RAI) that examines machines that can be built on the understanding of intelligent human behaviour. Here are some illustrative explanations from [11] where references to their original sources can also be found.

HAI is the art of creating machines that perform functions that require intelligence when performed by people. It is the study of how to make computers do things at which, at the moment, people are better. *RAI* is a field of study that seeks to explain and emulate intelligent behavior in terms of computational processes. It is the branch of computer science that is concerned with the automation of intelligent behavior.

Intelligent systems as seen nowadays have more to do with rationalistic than with humanistic AI. In addition to HAI features, IS admits intelligent behaviour as seen in nature as a whole; think, for example, on evolution, chaos, natural adaptation as intelligent behaviour. Moreover, IS are motivated by the need to solve complex problems with improving efficiencies.

Based on these and other similar considerations, an acceptable definition of intelligent systems was formulated in [11]. We adopt it here as follows.

> An *intelligent system* is a system that emulates some aspects of intelligence exhibited by nature. These include learning, adaptability, robustness across problem domains, improving efficiency (over time and/or space), information compression (data to knowledge), management of uncertain and imprecise information, extrapolated reasoning.

3 Computational Intelligence

The development of digital computers made possible the invention of human engineered systems that show intelligent behaviour or features. The branch of knowledge and science that emerged together and from such systems is called *artificial intelligence*. Instead of using this general name to cover practically any approach to intelligent systems, the AI research community restricts its meaning to *symbolic representations and manipulations in a top-down way* [4]. In other words, AI builds up an intelligent system by studying first the structure of the problem (typically in formal logical terms), then formal reasoning procedures are applied within that structure.

Alternatively, non-symbolic and bottom-up approaches (in which the structure is discovered and resulted from an unordered source) to intelligent systems are also known. The conventional approaches for understanding and predicting the behavior of such systems based on analytical techniques can prove to be inadequate, even at the initial stages of establishing an appropriate mathematical model. The computational environment used in such an analytical approach may be too categoric and inflexible in order to cope with the intricacy and the complexity of the real world industrial systems. It turns out that in dealing with such systems, one has to face a high degree of uncertainty and tolerate imprecision, and trying to increase precision can be very costly [16].

In the face of difficulties stated above fuzzy logic (FL), neural networks (NN) and evolutionary computation (EC) were integrated under the name *computational intelligence* (CI) as a hybrid system. Despite the relatively widespread use of the term CI, there is no commonly accepted definition of the term.

The birth of CI is attributed to the IEEE World Congress on Computational Intelligence in 1994 (Orlando, Florida). Since that time not only a great number of papers and scientific events have been dedicated to it, but numerous explanations of the term have been published. In order to have a brief outline of history of the term the founding and most interesting definitions will be summarized now.

The first one was proposed by Bezdek [1] as follows.

> A system is called *computationally intelligent* if it deals only with numerical (low-level) data, has a pattern recognition component, and does not use knowledge in the AI sense; and additionally, when it (begins to) exhibit (i) computational adaptivity; (ii) computational fault tolerance; (iii) speed approaching human-like turnaround, and (iv) error rates that approximate human performance.

Notice how the role of pattern recognition is emphasized here. In addition, remark that Bezdek concerns an artificially intelligent system as a CI system whose "added value comes from incorporating knowledge in a nonnumerical way."

In [14], one of the pioneering publications on computational intelligence, Marks defined CI by listing the building blocks being neural nets, genetic algorithms, fuzzy systems, evolutionary programming, and artificial life. Note that in more recent terminology genetic algorithms and evolutionary programming are called by the common name evolutionary computing.

In the book [6] Eberhart *et al.* formulated their own definition and its relationship to the one of Bezdek.

> Computational intelligence is defined as a methodology involving computing that exhibits an ability to learn and/or deal with new situations such that the system is perceived to possess one or more attributes of reason, such as generalisation, discovery, association, and abstraction.

Eberhart *et al.* stress adaptation rather than pattern recognition (Bezdek). They say it explicitly: *computational intelligence and adaptation are synonymous.* That is, in this sense CI do not rely on explicit human knowledge [9]. Notice that adaptability is one of the key features of intelligent systems also in the first definition above.

Closing this section, we briefly recall three typical opinions on the relationship between AI and CI, leaving to the reader to judge them.

In [14] Marks wrote: "Although seeking similar goals, CI has emerged as a sovereign field whose research community is virtually distinct from AI." This opinion declares that CI means an alternative to AI.

Bezdek in [1], after an analysis based on three levels of system complexity, came up with the conclusion that CI is a subset of AI. This viewpoint was criticized in [6].

Fogel formulated a third opinion in [9]. Starting from *adaptation* as the key feature of intelligence, and observing that traditional symbolic AI systems do not adapt to new problems in new ways, he declares that AI systems emphasize *artificial* and not the *intelligence*. Thus, it may be inferred that AI systems are not intelligent, while CI systems are.

4 Soft Computing

Prof. Lotfi A. Zadeh [20] proposed a new approach for Machine Intelligence, separating Hard Computing techniques based Artificial Intelligence from Soft Computing techniques based Computational Intelligence (Figure 1).

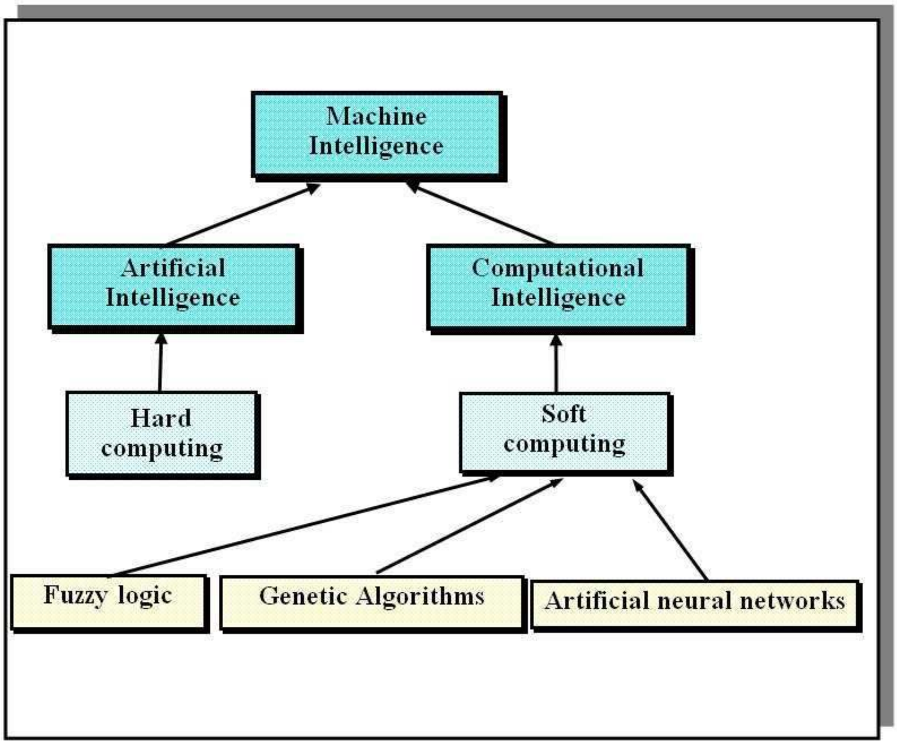


Figure 1: Artificial Intelligence vs. Computational Intelligence

Hard computing is oriented towards the analysis and design of physical processes and systems, and has the characteristics precision, formality, categoricity. It is based on binary logic, crisp systems, numerical analysis, probability theory, differential equations, functional analysis, mathematical programming, approximation theory and crisp software.

Soft computing is oriented towards the analysis and design of intelligent systems. It is based on fuzzy logic, artificial neural networks and probabilistic reasoning including genetic algorithms, chaos theory and parts of machine learning and has the attributes of approximation and dispositionality.

Although in hard computing imprecision and uncertainty are undesirable properties, in soft computing the tolerance for imprecision and uncertainty is exploited to achieve an acceptable solution at a low cost, tractability, high Machine Intelligence Quotient (MIQ). Prof. Zadeh argues that soft computing, rather than hard computing, should be viewed as the foundation of real machine intelligence.

Soft computing, as he explains, is

- a consortium of methodologies providing a foundation for the conception and design of intelligent systems,
- aimed at a formalization of the remarkable human ability to make rational decision in an uncertain and imprecise environment.

The guiding principle of soft computing is: *Exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, low solution cost and better rapport with reality.*

Fuzzy logic (FL) is mainly concerned with imprecision and approximate reasoning, neural networks (NN) mainly with learning and curve fitting, evolutionary computation (EC) with searching and optimization. FL systems are based on *knowledge-driven* reasoning, while NN and EC are *data-driven* approaches.

The main reason for the popularity of soft computing is the *synergy* derived from its components. SCs main characteristic is its intrinsic capability to create hybrid systems that are based on an integration of constituent technologies. This integration provides complementary reasoning and searching methods that allow us to combine domain knowledge and empirical data to develop flexible computing tools and solve complex problems.

The experiences gained over the past decade have also stressed that it can be more effective to use SC technologies in a combined manner, rather than exclusively. For example an integration of fuzzy logic and neural nets has already become quite popular (neuro-fuzzy control) with many diverse applications, ranging from chemical process control to consumer goods.

The constituents and the characteristics of hard and soft computing are summarized in Table I.

4.1 Fuzzy Logic

Fuzziness refers to nonstatistical imprecision and vagueness in information and data. Most concepts dealt with or described in our world are fuzzy. In classical logic, known as crisp logic, an element either is or is not a member of a set. That is, each element has a membership degree of either 1 or 0 in the set. In a fuzzy set, fuzzy membership values reflect the membership grades of the elements in the set. Membership function is the basic idea in fuzzy set theory based on fuzzy logic, which is the logic of "approximate reasoning." It is a generalization of conventional (two-valued, or crisp) logic. Fuzzy sets model the properties of imprecision, approximation, or vagueness. Fuzzy logic solves problems where crisp logic would fail.

Fuzzy logic gives us a language, with syntax and local semantics, in which we can translate qualitative knowledge about the problem to be solved. In particular, FL allows us to use *linguistic variables* to model dynamic systems. These variables take fuzzy values that are characterized by a label (a sentence generated from the syntax) and a meaning (a membership function determined by a local semantic procedure). The meaning of a linguistic variable may be interpreted as an elastic constraint on its value. These constraints are propagated by fuzzy inference operations, based on the generalized modus ponens. This reasoning mechanism, with its interpolation properties, gives FL a robustness with respect to variations in the system's parameters, disturbances, etc., which is one of FL's main characteristics [2].

Fuzzy logic is being applied in a wide range of applications in engineering areas ranging from robotics and control to architecture and environmental engineering. Other areas of application include medicine, management, decision analysis, and computer science. New applications appear almost daily. Two of the major application areas are *fuzzy control* and *fuzzy expert systems.*

4.2 Neural Networks

An artificial neural network (briefly: neural network) is an analysis paradigm that is roughly modeled after the massively parallel structure of the brain. It simulates a highly interconnected, parallel computational structure with many relatively simple individual processing elements. It is known for its ability to deal with noisy and variable information. For much more details on neural networks we refer to [8].

Based on some earlier ideas, Rosenblatt was the first who proposed the perceptron (a single-layer feedforward network) and showed that it can be used for pattern classification [15]. However, it turned out later that single-layer perceptrons cannot separate nonlinear or nonconvex regions. With the introduction of backpropagation, a new way opened to train multi-layered, feed-forward networks with nonlinear activation functions. Feedforward multilayer NNs are computational structures that can be trained to learn patterns from examples. They are composed of a network of processing units or neurons. Each neuron performs a weighted sum of its input, using the resulting sum as the argument of a non-linear activation function. One can use a training set that samples the relation between inputs and outputs, and a learning method that trains their weight vector to minimize a quadratic error function [2].

There are five areas where neural networks are best applicable [7]:

- Classification;
- Content Addressable Memory or Associative Memory;
- Clustering or Compression;
- Generation of Sequences or Patterns;
- Control Systems.

4.3 Evolutionary Computing

Evolutionary computing (EC) comprises machine learning optimization and classification paradigms roughly based on mechanisms of evolution such as biological genetics and natural selection. EC algorithms exhibit an adaptive behavior that allows them to handle non-linear, high dimensional problems without requiring differentiability or explicit knowledge of the problem structure. These algorithms are very robust to time-varying behavior, even though they may exhibit low speed of convergence [2].

The evolutionary computation field includes genetic algorithms, evolutionary programming, genetic programming, evolution strategies, and particle swarm optimization. It is known for its generality and robustness. Genetic algorithms are search algorithms that incorporate natural evolution mechanisms, including crossover, mutation, and survival of the fittest. They are used for optimization and for classification. Evolutionary programming algorithms are similar to genetic algorithms, but do not incorporate crossover. Rather, they rely on survival of the fittest and mutation. Evolution strategies are similar to genetic algorithms but use recombination to exchange information between population members instead of crossover, and often use a different type of mutation as well. Genetic programming is a methodology used to evolve computer programs. The structures being manipulated are usually hierarchical tree structures. Particle swarm optimization flies potential solutions, called particles, through the problem space. The particles are accelerated toward selected points in the problem space where previous fitness values have been high.

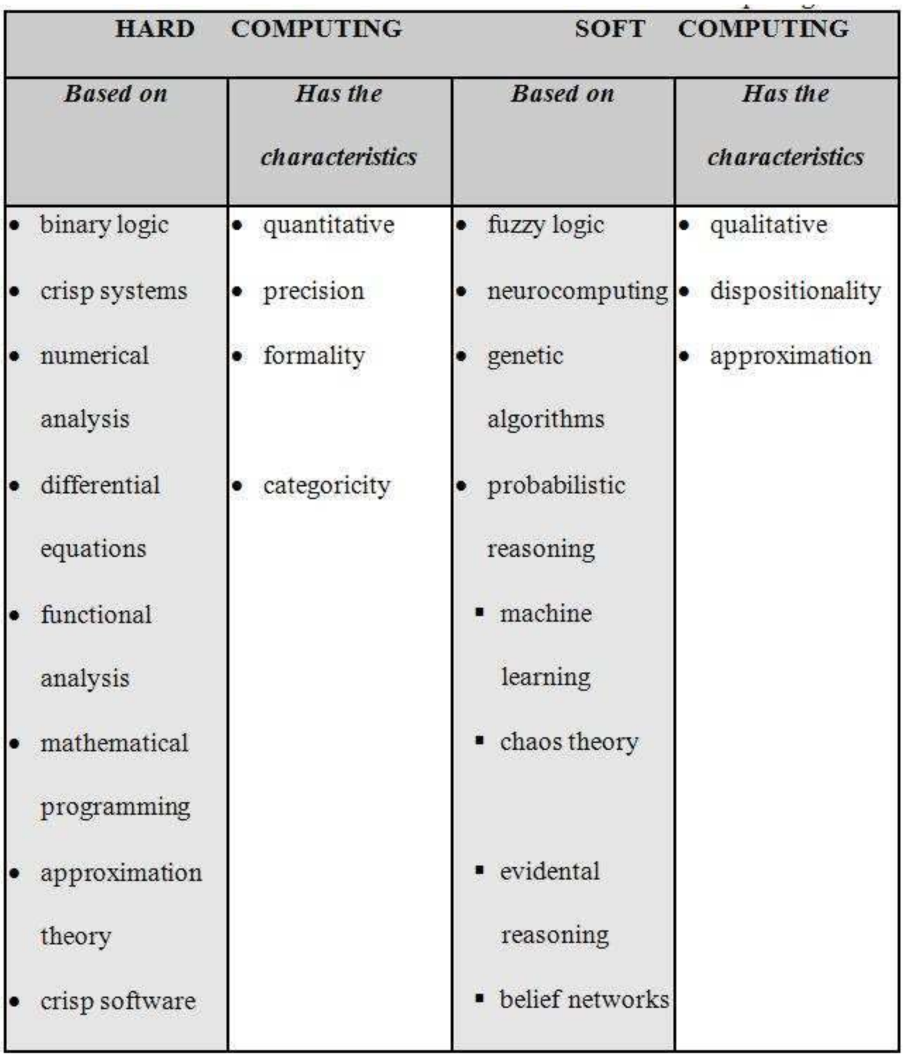
Evolutionary algorithms have been applied in *optimization* to multiple-fault diagnosis, robot track determination, schedule optimization, conformal analysis of DNA, load distribution by an electric utility, neural network explanation facilities, and product ingredient mix optimization. *Classification* applications include rule-based machine learning systems and classifier systems for high-level semantic networks. An application area of both optimization and classification is the evolution of neural networks.

5 Hybrid Systems

Hybrid systems combine two or more individual technologies (fuzzy logic, neural networks and genetic algorithms) for building intelligent systems. The individual technologies represent the various aspects of human intelligence that are necessary for enhancing performance. However, all individual technologies have their constraints and limitations. Having the possibility to put two or more of them together in a hybrid system increases the systems capabilities and performance, and also leads to a better understanding of human cognition. Over the past decade we have seen an increasing number of hybrid algorithms, in which two or more soft computing technologies have been integrated to leverage the advantages of individual approaches. By combining smoothness and embedded empirical qualitative knowledge with adaptability and general learning ability, these hybrid systems improve the overall algorithm performance [2]. Examples of such combinations include the following ones:

- fuzzy logic controller tuned by neural networks;
- fuzzy logic controller tuned by evolutionary computing;
- synthesis and tuning of neural networks by evolutionary computing;
- neural networks controlled by fuzzy logic;
- evolutionary computing controlled by fuzzy logic.

Table I. The constituents and the characteristics of hard and soft computing after [16].



HARD COMPUTING		SOFT COMPUTING	
Based on	*Has the characteristics*	*Based on*	*Has the characteristics*
• binary logic • crisp systems • numerical analysis • differential equations • functional analysis • mathematical programming • approximation theory • crisp software	• quantitative • precision • formality • categoricity	• fuzzy logic • neurocomputing • genetic algorithms • probabilistic reasoning ▪ machine learning ▪ chaos theory ▪ evidental reasoning ▪ belief networks	• qualitative • dispositionality • approximation

Hybrid systems illustrates the interaction of knowledge and data in soft computing. To tune knowledge-derived models we first translate domain knowledge into an initial structure and parameters and then use global or local data search to tune the parameters. To control or limit search by using prior knowledge we first use global or local search to derive the models (structure + parameters), we embed knowledge in operators to improve global search, and we translate domain knowledge into a controller to manage the solution convergence and quality of the search algorithm [2].

Several models are used for integrating intelligent systems. The one used in [10] classifies hybrid architectures into the following four categories:

Combination: typical hybrid architecture of this kind is a sequential combination of neural networks and rule- or fuzzy rule-based systems.

Integration: this architecture usually uses three or more individual technologies and introduces some hierarchy among the individual subsystems. For example, one subsystem may be dominant and may distribute tasks to other subsystems.

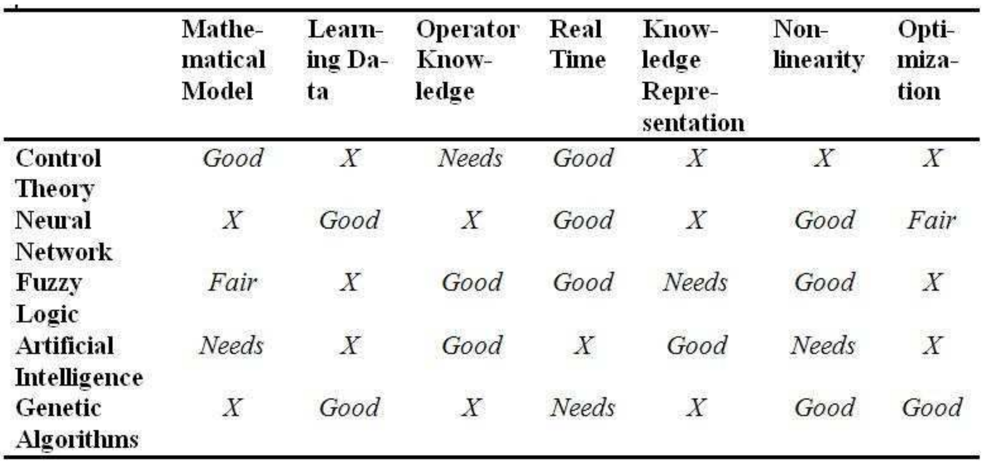
Fusion: a tight-coupling and merging architecture, usually based on the strong mathematical optimization capability of genetic algorithms and neural networks. When other techniques incorporate these features, the learning efficiency of the resulting system is increased.

Association: the architecture that includes different individual technologies, interchanging knowledge and facts on a pairwise basis.

Lotfi A. Zadeh expectation was explained as follows: *"in coming years, hybrid systems are likely to emerge as a dominant form of intelligent systems. The ubiquity of hybrid systems is likely to have a profound impact on the ways in which man-made systems are designed, manufactured, deployed and interacted with."*

Fuzzy logic is mainly concerned with imprecision and approximate reasoning, neural networks mainly with learning and curve fitting, evolutionary computation with searching and optimization. Table II gives a comparison of their capabilities in different application areas, together with those of control theory and artificial intelligence.

Table II. after [17]



	Mathematical Model	Learning Data	Operator Knowledge	Real Time	Knowledge Representation	Nonlinearity	Optimization
Control Theory	*Good*	*X*	*Needs*	*Good*	*X*	*X*	*X*
Neural Network	*X*	*Good*	*X*	*Good*	*X*	*Good*	*Fair*
Fuzzy Logic	*Fair*	*X*	*Good*	*Good*	*Needs*	*Good*	*X*
Artificial Intelligence	*Needs*	*X*	*Good*	*X*	*Good*	*Needs*	*X*
Genetic Algorithms	*X*	*Good*	*X*	*Needs*	*X*	*Good*	*Good*

Explanation of Symbols: Good=Good or suitable, Fair=Fair, Needs=Needs some other knowledge or techniques, X=Unsuitable or does not require.

Hybrid systems illustrates the interaction of knowledge and data in soft computing. To tune knowledge-derived models we first translate domain knowledge into an initial structure and parameters and then use global or local data search to tune the parameters. To control or limit search by using prior knowledge we first use global or local search to derive the models (structure + parameters), we embed knowledge in operators to improve global search, and we translate domain knowledge into a controller to manage the solution convergence and quality of the search algorithm.

6 Summary

In this paper we gave an overview of intelligent systems, computational intelligence and soft computing, and hybrid systems. The notions have been discussed in considerable details. Essential features were highlighted together with typical applicational areas.

References

[1] J.C. Bezdek, "What is computational intelligence?", In: J.M. Zurada, R.J. Marks II, and C.J. Robinson, Eds., *Computational Intelligence, Imitating Life,* IEEE Computer Society Press, pp. 1-12, 1994.

[2] P. Bonissone, "Hybrid soft computing systems: where are we going?", In: *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*,pp. 739-746, 2000.

[3] T.A. Byrd and R.D. Hauser, "Expert systems in production and operations management: research directions in assessing overall impact," *Int. J. Prod. Res.*, Vol. 29, pp. 2471-2482, 1991.

[4] B.G.W. Craenen, A.E. Eiben, "Computational Intelligence," *Encyclopedia of Life Support System,* EOLSS Co. Ltd., http://www.eolss.net, 2003.

[5] W. Duch, "What is Computational Intelligence and where is it going?", In: W. Duch and J. Mandziuk, Eds., *Challenges for Computational Intelligence,* Springer Studies in Computational Intelligence, Vol. 63, pp. 1-13, 2007.

[6] R. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools,* Academic Press, Boston, 1996.

[7] R.C. Eberhart and Y. Shui, *Computational Intelligence - Concepts to Implementations,* Elsevier, 2007.

[8] E. Fiesler and R. Beale, *Handbook of Neural Computation*, Oxford University Press, New York, 1997.

[9] D. Fogel, "Review of computational intelligence imitating life,"*IEEE Transactions on Neural Networks,* Vol. 6, pp. 1562-1565, 1995.

[10] M. Funabashi, A. Maeda, Y. Morooka and K. Mori, "Fuzzy and neural hybrid expert systems: synergetic AI," *IEEE Expert*, August, pp. 3240, 1995.

[11] K. Krishnakumar, "Intelligent systems for aerospace engineering – an overview," *NASA Technical Report*, Document ID: 20030105746, 2003.

[12] L. Madarász, *Intelligent Technologies and Their Applications in Large Scale Systems*, in Slovak, Elfa, Kosice, 2004.

[13] L. Madarász, R. Andoga, "Development and perspectives of situational control of complex systems", *Gép*, Vol. 55. No. 1, pp. 14-18, 2004.

[14] R. Marks, "Computational versus artificial," *IEEE Transactions on Neural Networks,* Vol. 4, pp. 737-739, 1993.

[15] F. Rosenblatt, "The perceptron, a perceiving and recognizing automaton", Project PARA, Cornell Aeronautical Lab. Rep., no. 85-640-1, Buffalo, NY, 1957.

[16] I.J. Rudas, Hybrid Systems (Integration of Neural Networks, Fuzzy Logic, Expert Systems, and Genetic Algorithms), In: *Encyclopedia of Information Systems,* Academic Press, pp. 114-1 - 114-8, 2002.

[17] I.J. Rudas and M.O. Kaynak, "Techniques in Soft Computing and their Utilization in Mechatronic Products," In: C.T. Leondes, Ed., *Diagnostic, Reliability and Control System Techniques,* Gordon and Beach International Series in Engineering, Technology and Applied Science Volumes on Mechatronics Systems Techniques and Applications, Vol. 5, Singapore, pp. 101-138, 2000.

[18] I.J. Rudas and J. Fodor, "Intelligent Systems," *International Journal of Computers Communications and Control,* Vol. 3, pp. 132-138, 2008.

[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach,* Prentice Hall, New Jersey, 1995.

[20] L.A. Zadeh, "Fuzzy Logic and Soft Computing: Issues, Contentions and Perspectives," In: *Proc. of IIZUKA'94: Third Int.Conf. on Fuzzy Logic, Neural Nets and Soft Computing,* Iizuka, Japan, pp. 1-2, 1994.

[21] L.A. Zadeh, "Fuzzy Logic, Neural Networks, and Soft Computing," *Communications of the ACM,* Vol. 37, pp. 77-84, 1994.

Imre J. Rudas
Budapest Tech
Institute of Intelligent Engineering Systems
Bécsi út 96/b, H-1034 Budapest, Hungary
E-mail: rudas@bmf.hu

János Fodor
Budapest Tech
Institute of Intelligent Engineering Systems
Bécsi út 96/b, H-1034 Budapest, Hungary
E-mail: fodor@bmf.hu

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 196-210

New Parallel Programming Language Design: A Bridge between Brain Models and Multi-Core/Many-Core Computers?

Gheorghe Stefanescu and Camelia Chira

Abstract: The recurrent theme of this paper is that sequences of long temporal patterns as opposed to sequences of simple statements are to be fed into computation devices, being them (new proposed) models for brain activity or multi-core/many-core computers. In such models, parts of these long temporal patterns are already committed while other are predicted. This combination of matching patterns and making predictions appears as a key element in producing intelligent processing in brain models and getting efficient speculative execution on multi-core/many-core computers. A bridge between these far-apart models of computation could be provided by appropriate design of massively parallel, interactive programming languages. Agapia is a recently proposed language of this kind, where user controlled long high-level temporal structures occur at the interaction interfaces of processes. In this paper, we link Agapia with HTMs brain models and with TRIPS multi-core/many-core architectures.

1 Introduction

We live in a paradox. On the one hand, recent technological advances suggest the possible transition to powerful multi-core/many-core computers in the near future. However, in order to be economically viable, such a major shift must be accompanied with a similar shift in software, where parallel programming should enter the mainstream of programming practice becoming the rule rather than the exception. Briefly, programs eager for more computing power are badly needed. On the other hand, there is a critical view that the promises of AI (Artificial Intelligence) are still to be fulfilled. No matter how much computer power we would have, the critics say, the advances in key AI areas as image recognition or understanding spoken languages will be slow. This means that the current AI approach is, according to critics, faulty.

AI already had a major restructuring in the nineties, by adopting the agent-oriented paradigm as a major research topic.[1] Jeff Hawkins [6] proposes another restructuring of AI by taking a closer look to the human brain. According to Hawkins, the efficient modelling of the human brain is of crucial importance if we really want to understand why human beings are so powerful on recognizing images or performing other similar tasks for which computers are still notoriously weak. Following suggestions provided by the anatomical structure of the brain, he proposes to use HTMs (Hierarchical Temporal Memories), hierarchical networks of nodes which work together processing continuous flows of data. While most of the pieces have been already used by other approaches (neural networks, associative memories, learning machines, interactive computing models, etc.), Hawkins stresses out the importance of having a unitary, coherent approach. In his view,

[1] See [2] for a recent presentation, centered on the use of agents for cooperative design in a distributed environment.

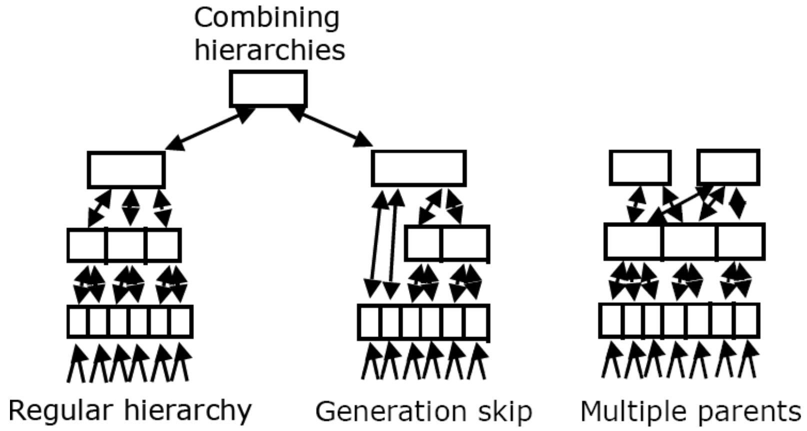


Figure 1: Hierarchical Temporal Memories

the interplay between a systematic study of the brain and a creative design approach for developing intelligent machines is the solution to the current AI crisis.

Our paper briefly presents HTMs and other key elements of Hawkins model of the brain [6]. Furthermore, it describes the specific features of a particular architecture called TRIPS (Tera-op, Reliable, Intelligently adaptive Processing System) for multi-core/many-core computers [1]. The main contribution of the paper might be the suggestion that Agapia, a recently proposed language for massively parallel, interactive programs [4, 7], or similar languages, can potentially be a bridge between brain models, such as those using HTMs, and multi-core/many-core computers, particularly those using the TRIPS architectures. To strengthen the suggestion, the paper shows how Agapia programs for modeling HTMs can be developed and sketches an approach for compiling and running Agapia programs on TRIPS computers.

2 HTMs - models for brain activity

Understanding brain activity was and still is so difficult that even speculative theories on "how the brain could work" are rare. In a recent book "On Intelligence" [6], Hawkins proposes a computation model that may explain the striking differences between the current computers and the brain capabilities, especially in such areas as visual pattern recognition, understanding spoken language, recognizing and manipulating objects by touch, etc. Hawkins includes a rich set of biological evidences on the anatomical structure of the brain that may support his computation model.

The model uses HTMs (Hierarchical Temporal Memories) to build up intelligent devices. HTMs consist of hierarchical tree-like P2P networks where each node runs a similar *learning and predicting* algorithm. The adequacy of these HTMs to discover the hidden structure of our outside world lies in the belief on the hierarchical structure of the world itself.

Fig. 1 illustrates various types of HTMs. On the left, they are simple tree structures. On the right, an example of an HTM with shared subtrees is given. This latter example shows that generally HTMs could have a DAG (Directly Acyclic Graph) structure. However, as the processing flow may go up and down, or even between nodes at the same level, the resulting graphs are pretty general. The hierarchical structure is useful mostly as a conceptual approach for understanding the complicated structure and the complex activity of the brain.

Getting the right level of processing. Briefly, the activity of a mature brain is as follows. At each node, sequences of temporal patterns arrive and are classified according to a learned schema. When this process is fully done, the patterns are replaced by short codes of the classes they were classified into and the sequences of these codes are forwarded to an upper node in the HTM hierarchy. This resembles the situation in a hierarchically structured company where an employee tells his/her superior “I have done this and this and this...,” without entering into details. During the classification process, a node may look at a few starting temporal data from its incoming pattern, guesses the class the pattern falls into, and predicts the rest of the sequence. This gives a robust processing in the case the input data are partially ambiguous or have errors.

Except for the explained forward flow of information in a HTM, there is also a feedback flow from upper to lower nodes in the hierarchy. In the case a pattern can not be certainly classified by a node, the node might forward the full pattern to his/her upper node in the HTM hierarchy asking for help. Using the above analogy, this is like an employee telling his/her superior “You see, I do not know what to do in this case. Could you help me?” Depending on the experience of the HTM (or of the brain that the HTM models), the process of getting the right level of processing of the input patterns may be at a lower or a higher level in the hierarchy. Experienced HTMs/brains tend to fully process the input patterns at lower levels, while inexperienced ones, for instance those found in a learning process, tend to process the input patterns at higher levels in the hierarchy. Once a learned process is stable at a hierarchy level, it can be shifted “down to the hierarchy” for latter processing. This way, upper levels of the hierarchy become free and may process more abstract patterns, concepts, or thoughts.

Associations. The focus in the previous paragraph was on the vertical structure in this HTM hierarchical model of the brain: how to learn and classify the incoming pattern in isolation. Actually, the brain and the corresponding HTM models have very powerful association mechanisms. These association mechanisms act either directly at a given level in a hierarchy (nodes are informed on the activity of their close neighbors), or far-apart via the hierarchical structure. In the latter case, information from different sensory systems may be combined (e.g., simultaneous recognition of sounds and visual images) for a better and faster processing.

Perception and action. While most of the intuition behind the above examples comes from the human perception system, this HTM model of the brain makes no difference between the perception and the action mechanisms: the same kind of hierarchal structure and the same processing mechanisms are present in the motor areas where brain thoughts are translated into visible behaviors.

Numenta and intelligent machines. The pitfalls of Hawkins’ approach may come from the yet-to-be-discovered *learning and predicting algorithm* used in the nodes of the

HTM models of the brain. While this might take long and might be very difficult to discover, actually Hawkins has paved another way focusing on using HTMs to build "intelligent machines." His new company Numenta is planing to build computing chips based on HTM models and using appropriate learning algorithms. Whether these algorithms do fit or not with the ones used by the brain may be irrelevant - in design, we do not have to copy the nature: our cars have no legs, our planes have not bird-like wings.

Turing test on intelligence. We close this brief presentation of Hawkins' model of the brain with a more philosophical discussion. What is "intelligence" and what it means for a computer to be "as intelligent as a human being" were (and still are) long debated questions. Alan Turing has invented Turing machines, a mechanical model of computation on which the modern computers are based. Turing has proposed this famous *Turing test*: a computer is as intelligent as a human being if it is behaviorally equivalent with a human being. In other words, an external observer can not see a difference between his/her interaction with a computer or with a human being.

Searle, a fervent critic of this kind of intelligence test, came up with a "Chinese Room" thought experiment, showing that an English-speaking person following a set of rules (the analogy of a computer program) can properly answer Chinese-written questions without actually understanding Chinese. His conclusion is that intelligence and understanding can not be reduced to behavior.

Hawkins' model places more emphasis on "prediction" in his attempt to capture a definition for intelligence. Understanding is closely linked with the capacity of prediction. Ultimately, understanding and intelligence may be completely internal, in the brain, without any visible external behavior.

Ironically, the seminal paper of Turing contains a remark saying that what he has introduced is an *a-machine*, an autonomous one, and there is another notion of *s-machine*, an interactive one, which was not considered there. This difference between a closed and an open (interactive) approach may explain the main difference between Turing and Hawkins: Turing has used his autonomous machine reducing intelligence to its external behavior, while Hawkins uses an interactive approach with a sophisticated dance between the processing of the real input patterns and what the machine expects from its own prediction.

3 Agapia - a parallel, interactive programming language

Interactive computing [5] is a step forward on system modularization. The approach allows to describe parts of the systems and to verify them in an open environment. A model for interactive computing systems (consisting of interactive systems with registers and voices - *rv-systems*) and a core programming language (for developing *rv-programs*) have been proposed in [8] based on register machines and a space-time duality transformation. Structured programming techniques for rv-systems and a kernel programming language Agapia have been later introduced [4], with a particular emphasis on developing a structural spatial programming discipline.

Structured process interaction greatly simplifies the construction and the analysis of interactive programs. For instance, method invocation in current OO-programming techniques may produce unstructured interaction patterns, with free `goto`'s from a process to another and should be avoided. Compared with other interaction or coordination calculi, the rv-systems approach paves the way towards a name-free calculus and facilitates the development of a modular reasoning with good expectations for proof scalability to systems with thousands of processes. A new and key element in this structured interaction model is the extension of temporal data types used on interaction interfaces. These new temporal data types (including voices as a time-dual version of registers) may be implemented on top of streams as the usual data types are implemented on top of Turing tapes.

Agapia [4, 7] is a kernel high-level massively parallel programming language for interactive computation. It can be seen as a coordination language on top of imperative or functional programming languages as C++, Java, Scheme, etc. Typical Agapia programs describe open processes located at various sites and having their temporal windows of adequate reaction to the environment. The language naturally supports process migration, structured interaction, and deployment of components on heterogeneous machines. Despite of allowing these high-level features, the language can be given simple denotational and operational semantics based on scenarios (scenarios are two-dimensional running patterns; they can be seen as the closure with respect to a space-time duality transformation of the running paths used to define operational semantics of sequential programs).

3.1 Scenarios

This subsection briefly presents temporal data, grids, scenarios, and operations on scenarios.

Temporal data. What we call "spatial data" are just the usual data occurring in imperative programming. For them, common data structures and the usual memory representation may be used. On the other hand, "temporal data" is a name we use for a new kind of (high-level) temporal data implemented on streams. A *stream* is a sequence of data ordered in time. (The time model in Agapia is discrete.) Typically, a stream results by observing data transmitted along a channel: it exhibits a datum (corresponding to the channel type) at each clock cycle.

A *voice* is defined as the time-dual of a register: *It is a temporal data structure that holds a natural number. It can be used ("heard") at various locations. At each location it displays a particular value.*

Voices may be implemented on top of a stream in a similar way registers are implemented on top of a Turing tape, for instance specifying their starting time and their length. Most of usual data structures have natural temporal representations. Examples include timed booleans, timed integers, timed arrays of timed integers, etc.

Grids and scenarios. A *grid* is a *rectangular* two-dimensional array containing letters in a given alphabet. A grid example is presented in Fig. 2(a). The default interpretation is that columns correspond to processes, the top-to-bottom order describing their progress in

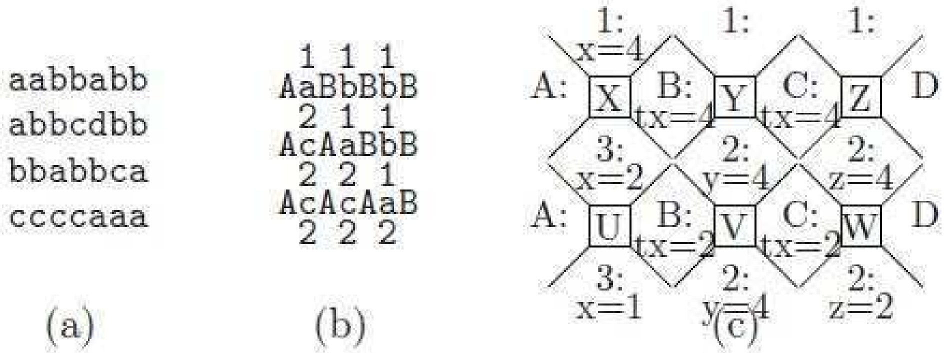


Figure 2: A grid (a), an abstract scenario (b), and a concrete scenario (c).

time. The left-to-right order corresponds to process interaction in a *nonblocking message passing discipline*: a process sends a message to the right, then it resumes its execution.

A *scenario* is a grid enriched with data around each letter. The data may be given in an abstract form as in Fig. 2(b), or in a more detailed form as in Fig. 2(c).

The type of a scenario interface is represented as $t_1; t_2; \ldots; t_k$, where each t_k is a tuple of simple types used at the borders of scenario cells. The empty tuple is also written 0 or *nil* and can be freely inserted to or omitted form such descriptions. The type of a scenario is specified as $f : \langle w|n\rangle \rightarrow \langle e|s\rangle$, where w, n, e, s are the types for its west, north, east, south interfaces.

Operations with scenarios. Two interfaces $t = t_1; t_2; \ldots; t_k$ and $t' = t'_1; t'_2; \ldots; t'_{k'}$ are *equal*, written $t = t'$, if $k = k'$ and the types and the values of each pair t_i, t'_i are equal. Two interfaces are *equal up to the insertion of nil elements*, written $t =_n t'$, if they become equal by appropriate insertions of *nil* elements.

Let $Id_{m,p} : \langle m|p\rangle \rightarrow \langle m|p\rangle$ denote the constant cells whose temporal and spatial outputs are the same with their temporal and spatial inputs, respectively; an example is the center cell in Fig. 3(c), namely $Id_{1,2}$.

Horizontal composition: Let $f_i : \langle w_i|n_i\rangle \rightarrow \langle e_i|s_i\rangle, i = 1, 2$ be two scenarios. Their *horizontal composition* $f_1 \triangleright f_2$ is defined only if $e_1 =_n w_2$. For each inserted *nil* element in an interface (to make the interfaces e_1 and w_2 equal), a dummy row is inserted in the corresponding scenario, resulting a scenario $\bar{f}_i$. The result $f_1 \triangleright f_2$ is obtained putting $\bar{f}_1$ on left of $\bar{f}_2$. The operation is briefly illustrated Fig. 3(b). The result is unique up to insertion or deletion of dummy rows. Its identities are $Id_{m,0}, m \geq 0$.

Vertical composition: The definition of *vertical composition* $f_1 \cdot f_2$ (see Fig. 3(a)) is similar, but now using the vertical dimension. Its identities are $Id_{0,m}, m \geq 0$.

Diagonal composition: The *diagonal composition* $f_1 \bullet f_2$ (see Fig. 3(c)) is a derived operation defined only if $e_1 =_n w_2$ and $s_1 =_n n_2$. The result is defined by the formula

$$f_1 \bullet f_2 = (f_1 \triangleright R_1 \triangleright \Lambda) \cdot (S_2 \triangleright Id \triangleright R_2) \cdot (\Lambda \triangleright S_1 \triangleright f_2).$$

for appropriate constants R, S, Id, Λ. Its identities are $Id_{m,n}, m, n \geq 0$. (The involved constants R, S, Id, Λ are described below.)

Constants: Except for the defined identities, we use a few more constants. Most of them can be found in Fig. 3(c): a recorder R (2nd cell in the 1st row), a speaker S (1st

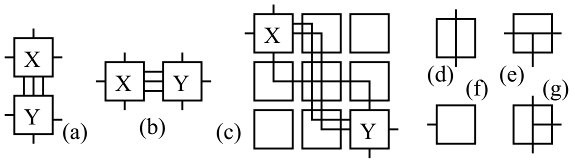


Figure 3: Operations on scenarios

cell in the 2nd row), an empty cell Λ (3rd cell in the 1st row). Other constants of interest are: transformed recorders Fig. 3(e) and transformed speakers Fig. 3(g).

3.2 Structured rv-programs

The syntax of structured rv-programs. The basic blocks for constructing structured rv-programs are called *modules*. A module gets input data from its west and north interfaces, process them (applying the module's code), and delivers the computed outputs to its east and south interfaces.

On top of modules, structured rv-programs are built up using "if" and both, composition and iterated composition statements for the vertical, the horizontal, and the diagonal directions. The composition statements capture at the program level the corresponding operations on scenarios. The iteration statements are also called the *temporal*, the *spatial*, and the *spatio-temporal while statements* - their scenario meaning is described below.

The *syntax for structured rv-programs* is given by the following BNF grammar

$$\begin{aligned} P &::= X \mid if(C)then\{P\}else\{P\} \mid P\%P \mid P\#P \mid P\$P \\ &\quad \mid while\_t(C)\{P\} \mid while\_s(C)\{P\} \mid while\_st(C)\{P\} \\ X &::= module\{listen\ t\_vars\}\{read\ s\_vars\} \\ &\quad \{code;\}\{speak\ t\_vars\}\{write\ s\_vars\} \end{aligned}$$

This is a core definition of structured rv-programs, as no data types or language for module's code are specified. Agapia, to be shortly presented, is a concrete incarnation of structured rv-programs into a fully running environment.

Notice that we use a different notation for the composition operators on scenarios $\cdot, \triangleright, \bullet$ and on programs $\%, \#, \$$; moreover, the extension of the usual composition operator ';' to structured rv-programs is denoted by "%".

Operational semantics. The operational semantics

$$|\ \ | : \text{Structured rv-programs} \rightarrow Scenarios$$

associates to each program the set of its running scenarios.

The type of a program P is denoted $P : \langle w(P)|n(P)\rangle \rightarrow \langle e(P)|s(P)\rangle$,
where $w(P)/n(P)/e(P)/s(P)$ indicate its types at the west/north/east/south borders. On each border, the type may be quite complex - the convention is to separate by "," the data from within a module and by ";" the data coming from different modules. This convention applies to both spatial and temporal data.

We say, two interface types *match* if they have a nonempty intersection.

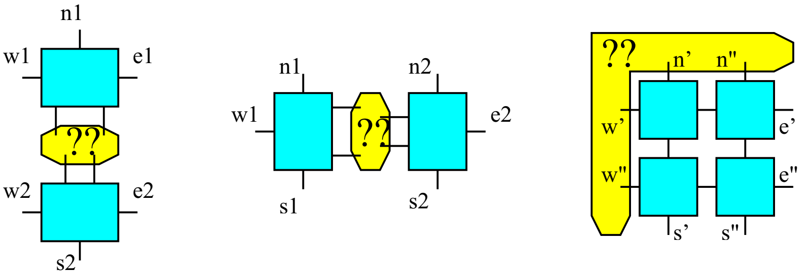


Figure 4: The vertical and the horizontal compositions and the "if" statement

Modules. The modules are the starting blocks for building structured rv-programs. The `listen (read)` instruction is used to get the temporal (spatial) input and the `speak (write)` instruction to return the temporal (spatial) output. The `code` consists of simple instructions as in the C code. No distinction between temporal and spatial variables is made within a module.

A scenario for a module consists of a unique cell, with concrete data on the borders, and such that the output data are obtained from the input data applying the module's code.

Composition. Programs may be composed "horizontally" and "vertically" as long as their types on the connecting interfaces agree. They can also be composed "diagonally" by mixing the horizontal and vertical compositions.

For two programs $P_i : \langle w_i|n_i\rangle \rightarrow \langle e_i|s_i\rangle$, $i = 1, 2$ we define the following composition operators.

Horizontal composition: $P_1\#P_2$ is defined if the interfaces e_1 and w_2 match, see Fig. 4(middle). The type of the composite is $\langle w_1|n_1; n_2\rangle \rightarrow \langle e_2|s_1; s_2\rangle$. A scenario for $P_1\#P_2$ is a horizontal composition of a scenario in P_1 and a scenario in P_2.

Vertical composition: $P_1\%P_2$ is similar.

Diagonal composition: $P_1\$P_2$ is defined if e_1 matches w_2 and s_1 matches n_2. The type of the composite is $\langle w_1|n_1\rangle \rightarrow \langle e_2|s_2\rangle$. A scenario for $P_1\$P_2$ is a diagonal composition of a scenario in P_1 and a scenario in P_2.

If. For two programs $P_i : \langle w_i|n_i\rangle \rightarrow \langle e_i|s_i\rangle$, $i = 1, 2$, a new program $Q = if\ (C)\ then\ \{P_1\}\ else\ \{P_2\}$ is constructed, where C is a condition involving both, the temporal variables in $w_1 \cap w_2$ and the spatial variables in $n_1 \cap n_2$, see Fig. 4(right). The type of the result is $Q : \langle w_1 \cup w_2|n_1 \cup n_2\rangle \rightarrow \langle e_1 \cup e_2|s_1 \cup s_2\rangle$.

A scenario for Q is a scenario of P_1 when the data on west and north borders of the scenario satisfy condition C, otherwise is a scenario of P_2 (with these data on these borders).

While. Three types of while statements are used for defining structured rv-programs, each being the iteration of a corresponding composition operation.

Temporal while: For $P : \langle w|n\rangle \rightarrow \langle e|s\rangle$, the statement $\mathit{while\_t}\ (C)\{P\}$ is defined if the interfaces n and s match and C is a condition on the spatial variables in $n \cap s$. The type of the result is $\langle (w;)^*|n \cup s\rangle \rightarrow \langle (e;)^*|n \cup s\rangle$. A scenario for $\mathit{while\_t}\ (C)\{P\}$ is either an

Interfaces

$SST ::= \ nil \mid sn \mid sb \mid (SST \cup SST)$
$\qquad \mid (SST, SST) \mid (SST)^*$
$ST ::= \ (SST) \mid (ST \cup ST)$
$\qquad \mid (ST;ST) \mid (ST;)^*$
$STT ::= \ nil \mid tn \mid tb \mid (STT \cup STT)$
$\qquad \mid (STT, STT) \mid (STT)^*$
$TT ::= \ (STT) \mid (TT \cup TT)$
$\qquad \mid (TT;TT) \mid (TT;)^*$

Expressions

$V ::= \ x : ST \mid x : TT \mid V(k)$
$\qquad \mid V.k \mid V.[k] \mid V@k \mid V@[k]$
$E ::= n \mid V \mid E + E \mid E * E \mid E - E \mid E/E$
$B ::= b \mid V \mid B\&\&B \mid B||B \mid !B \mid E < E$

Programs

$W ::= \ nil \mid new\ x : SST \mid new\ x : STT$
$\qquad \mid x := E \mid if(B)\{W\}else\{W\}$
$\qquad \mid W;W \mid while(B)\{W\}$
$M ::= \ module\{listen\ x : STT\}\{read\ x : SST\}$
$\qquad \{W;\}\{speak\ x : STT\}\{write\ x : SST\}$
$P ::= \ nil \mid M \mid if(B)\{P\}else\{P\}$
$\qquad \mid P\%P \mid P\#P \mid P\$P$
$\qquad \mid while\_t(B)\{P\} \mid while\_s(B)\{P\}$
$\qquad \mid while\_st(B)\{P\}$

Figure 5: The syntax of Agapia v0.1 programs

identity, or a repeated vertical composition $f_1 \cdot f_2 \cdot \cdots \cdot f_k$ of scenarios for P such that: (1) the north border of each f_i satisfies C and (2) the south border of f_k does not satisfy C.

Spatial while: $while\_s\ (C)\{P\}$ is similar.

Spatio-temporal while: $while\_st\ (C)\{P\}$, where $P : \langle w|n\rangle \to \langle e|s\rangle$, is defined if w matches e and n matches s and, moreover, C is a condition on the temporal variables in $w{\cap}e$ and the spatial variables in $n{\cap}s$. The type of the result is $\langle w{\cup}e|n{\cup}s\rangle \to \langle w{\cup}e|n{\cup}s\rangle$. A scenario for $while\_st\ (C)\{P\}$ is either an identity, or a repeated diagonal composition $f_1 \bullet f_2 \bullet \cdots \bullet f_k$ of scenarios for P such that: (1) the west and north border of each f_i satisfies C and (2) the east and south border of f_k does not satisfy C.

3.3 Agapia

Syntax of Agapia v0.1 programming language. The syntax for Agapia v0.1 programs is presented in Fig. 5. The v0.1 version is intentionally kept simple to illustrate the key features of the approach (see [7] for an extension v0.2 including high-level structured rv-programs). Agapia v0.1 forms a kind of minimal interactive programming languages: it describes what can be obtained from classical while programs allowing for spatial and temporal integer and boolean types and closing everything with respect to space-time duality.

The types for spatial interfaces are built up starting with integer and boolean sn, sb types, applying the rules for $\cup$, ',', $(\_)^*$ to get process interfaces, then the rules for $\cup$, ';', $(\_;)^*$ to get system interfaces. The temporal types are similarly introduced. For a spatial or temporal type V, the notations $V(k), V.k, V.[k], V@k, V@[k]$ are used to access its components. Expressions, usual while programs, modules, and programs are then naturally introduced.

4 Agapia programs for HTMs models

The current approach is to give Agapia scenario-based semantics with linear models for space and time. When different models are needed, as tree models for the HTMs presented in this paper, a linear representation of such models is required. Fortunately, there is a huge amount of work on similar topics involving representation of an endless number of data structures in the linear virtual memory model of conventional computers.

We focus our design of Agapia programs below on the HTM in the left side of Fig. 1 (the regular hierarchy, restricted to 2 levels: top, level 1, level 2) considered as a HTM model for a part of a visual sensory system.

Tree structures are represented recursively labeling their nodes by strings as follows: "if a node p in the tree is labelled by w and a node q in the tree is his/her i-th son (direct descendent), counting the positions from left to right, then the code of q is wi."

In our example, the codes of the nodes are: nil (for the top node), 1,2,3 (for the nodes on level 1), and 11,12,21,22,31,32 (for the nodes on level 2). The nodes are placed in a linear order using an extension of the Left-Right-Root parsing in binary trees. In our case the result is: 11,12,1,21,22,2,31,32,3,nil. With this convention, it is easier to describe Agapia programs for the forward flow of information, but slightly more complicated for the feedback flow. In the sequel, we suppose each process knows his/her code and the codes of other nodes in the structure.

Our approach to modeling HTMs with Agapia programs consists of three steps.

The first step requires more or less conventional programming for the basic modules. As one can develop Agapia on top of C++ or Java, code in such languages could be used in these modules. The code has to implement the following features.

1. Each node has a classifying algorithm, for instance using a "typical representative" for each class. Suppose the classes are $\{C_1, \ldots, C_n\}$ and their representatives are $\{t_1, \ldots, t_n\}$. Given a temporal pattern t (a voice, or a more complex temporal data structure) the node find the best matching of t against $t_1, \ldots, t_n$. According to Hawkins, this classification should be unique. However, increasingly sophisticated procedures may be used to reach this result, for instance using the feedback flow from top-to-bottom in the hierarchy or the association flow from the neighboring nodes.

2. An alternative to the *best full matching* is to use a *prefix matching*: once a prefix of t was parsed and it fits with a unique t_i, then the rest of t is ignored.

3. Each class C_i is supposed to have a name n_i (codified with much shorter sequences than for t or t_i's). The final product of the node is the passing of the code n_k of the class C_k for which t has the best fit to his/her HTM parent.

4. In contrast with 2, another alternative is to have *fully attentive* nodes, keeping track on all details. In such a case, if the input pattern does fit well with none of t_i's, the node passes the full t (not just a code for his/her better fitting class) to his/her parent for further processing.

5. Finally, higher level nodes, except for their own classification, have to process the exceptions in the classification procedures of their descendent nodes.

   The second step is to describe the forward flow of information in this HTM. It is just a particular format of MPI-like communication mechanisms for which Agapia macro-programs can be easily written (see, e.g., [3]). The shape of the program is as follows.

6. The program contains a main diagonal while statement. It repeats the processing for repeated incoming temporal patterns t's.

7. During a step of the diagonal while statement, each node gets input patterns from the left, processes them, and passes the results to the right. This means, for the leaves the input data come from outside (from the open temporal interfaces), while for the inner nodes they come from their own descendents. The results are passed to the parents, which fortunately are placed on the right in this order. This way, when a body of the diagonal while statement is executed, the forward flow in the HTM is fully modeled.

The third step is to show how the feedback flow in the HTM can be modeled. This is slightly more complicated, as we have chosen a linear order to facilitate the modeling of the forward flow. Indeed, as the interaction in Agapia programs goes from left to right, when a parent node wants to send a message to a son, an extra diagonal composition is needed to model this communication.

5 TRIPS - a multi-core/many-core architecture

With a privileged role between programming languages and hardware, the instruction set used in computer architecture design is very conservative. Changing or introducing new *ISA (Instruction Set Architecture)* is disruptive for computer systems and may be very risky. Nevertheless, the time for a radical change is imperative. The old CISC/RISC instruction sets no longer fit with the huge potential of the forthcoming multi-core/many-core computers. Introducing new ISA is now worthwhile having the potential to address the challenges of modern technologies and to exploit various integration possibilities [1]. In this context, *TRIPS (Tera-op, Reliable, Intelligently adaptive Processing System)* architectures are a very promising recent proposal facilitating higher exploitation of data, instruction-level, and thread-level parallelisms [10].

TRIPS is an instantiation of the *EDGE (Explicit Data Graph Execution)* ISA concept. EDGE is a new class of ISAs that views an instruction stream as blocks of instructions for a single task using isolated data. The main feature of an EDGE architecture refers to direct instruction communication which enables a dataflow-like execution. Unlike RISC and CISC instruction sets, EDGE explicitly encodes dependences into individual instructions. The hardware is not required to rediscover data dependences dynamically at runtime because the compile-time dependence graph is expressed through the ISA. Higher exposed concurrency and power-efficient execution are therefore facilitated by an EDGE architecture [1]. EDGE overcomes major drawback issues of the RISC and CISC architectures such as the usage of inefficient and power-consuming structures.

Offering increased flexibility, TRIPS supports a static placement of instructions (driven by compiler) and dynamic issue (hardware-determined) execution model. Graphs of predicated hyperblocks are compiled and represented internally as a dataflow graph. Communication between hyperblocks is possible via a set of input and output registers.

The TRIPS architecture aims to increase concurrency, to achieve power-efficient high performance and to diminish communication delays. Concurrency is increased by using an array of arithmetic logic units (ALUs) executed concurrently. ALUs provide scalable issue width as well as scalable instruction window size [1]. The TRIPS architecture minimizes execution delays by using compile-time instruction placement. Computation patterns are efficiently supported by the dataflow-like execution model of TRIPS.

The block-atomic execution engaged in TRIPS works as follows [1]:

- Instructions are grouped by the compiler into groups of instructions called hyperblocks (each hyperblock contains up to 128 instructions) and mapped to an array of execution units;
- Each hyperblock is fetched, executed, and committed *atomically*; instructions are fetched in parallel and loaded into the instruction buffers at each ALU;
- Instructions are efficiently executed by the hardware using a fine-grained dataflow model.

TRIPS can effectively support parallelism (instruction-level, data-level, and thread-level parallelism). As long as the software can discover parallelism, the TRIPS architecture will effectively exploit it. Being easy to scale up and down in performance, TRIPS overcomes the scheduling problems of traditional designs as well as the explicit unit exposure of VLIW (Very Long Instruction Word) designs.

6 Running Agapia programs on TRIPS architectures

We end our trip from brain models to multi-core/many-core computers with some remarks on the possibility of compiling and running Agapia programs on TRIPS architectures.

To illustrate the approach, we consider a simple example involving perfect numbers. A number is perfect if it is equal to the sum of its proper divisors. Before showing Agapia programs for perfect numbers, we describe two typical running scenarios for this task (one for a perfect number, the other for an imperfect one) in Fig. 6. The input-output relation is: *if the input number in the upper-left corner is n, then the output number in the lower-right corner is 0 iff n is perfect.*

The scenarios in Fig. 6 use cells whose behaviors are captured by the modules in Table 1.

Our first Agapia program **Perfect1** corresponds to the construction of the scenarios by rows:

$$(X \ \# \ Y \ \# \ Z) \ \% \ while\_t(x > 0)\{U \ \# \ V \ \# \ W\}$$

The type of the program is **Perfect1** : $\langle nil|sn; nil; nil\rangle \rightarrow \langle nil|sn; sn; sn\rangle$. Actually, the result is a program similar with a usual imperative program. There are some "transactions," each transaction specifying a macro-step in the whole system. The interaction part is simple and it reduces to the interaction of the components in a macro-step.

Our second Agapia program **Perfect2** corresponds to the construction of the scenarios by columns:

$$\begin{aligned}&(X \ \% \ while\_t(x > 0)\{U\} \ \% \ U1)\\ &\# \ (Y \ \% \ while\_t(tx > -1)\{V\} \ \% \ V1)\\ &\# \ (Z \ \% \ while\_t(tx > -1)\{W\} \ \% \ W1\end{aligned}$$

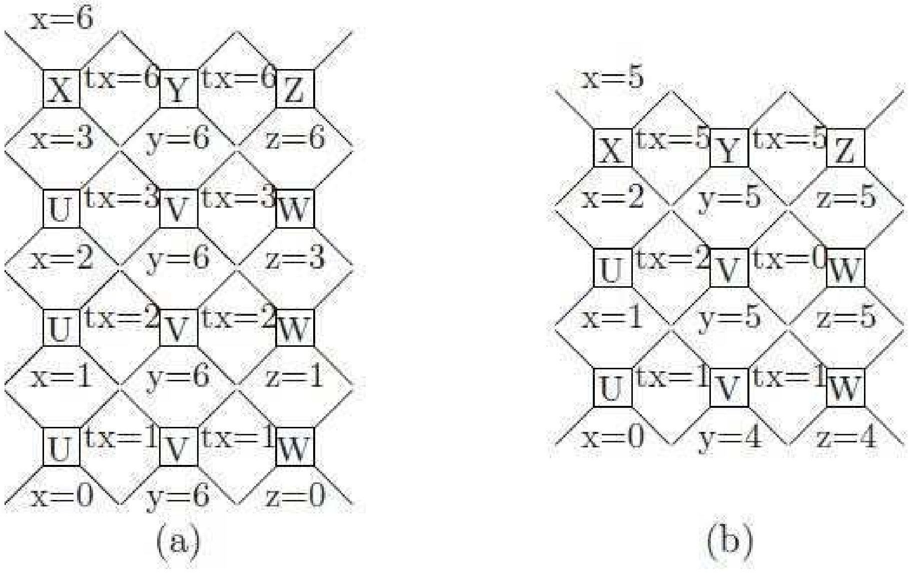


Figure 6: Scenarios for perfect numbers

Its type is **Perfect2** : $\langle nil|sn;nil;nil\rangle \rightarrow \langle nil|nil;nil;sn\rangle$. This variant resembles the dataflow computation paradigm. Each component acts as a stream processing function and the overall result comes from the interaction of these components.

The first program is appropriate for running on classical architectures, while the last one for dataflow architectures. TRIPS architecture is a combination of both. The current prototype uses sequences of up to 128 instructions to feed its matrix of ALUs. Agapia is very flexible and expressive, for instance the above two programs are just the extreme cases of a rich variety of possibilities. More precisely, one could unroll the first program, or restrict the number of steps in each component in the second program to get programs which fit better with the TRIPS architecture. Such transformations might be performed automatically to help the user to focus on the logic of the program and not on the target computer running his/her program.

Compiling Agapia programs for TRIPS architecture is without a doubt a very challenging direction. While our intuition strongly supports such an attempt, the painful procedure of writing a compiler and running programs is actually needed to clarify how well Agapia language and TRIPS architecture fit together.

7 Conclusions and future work

Most programs for AI tasks are inefficient on traditional computers with their Von-Neumann architecture and using imperative programming style. The proposed dataflow machines from eighties-nineties, specific for AI tasks, never had a major impact on the market. What we see in the recently proposed TRIPS architectures is a combination of dataflow and Von-Neumann styles, particularly using speculative execution of long blocks of instructions on the computers arrays of ALUs.

```
module X{listen nil;}{read x:sn;}
    {tx:tn; tx=x; x=x/2;}{speak tx;}{write x;}
module Y{listen tx:tn;}{read nil;}
    {y:sn; y=tx;}{speak tx;}{write y;}
module Z{listen tx:tn;}{read nil;}
    {z:sn; z=tx;}{speak nil;}{write z;}
module U{listen nil;}{read x:sn;}
    {tx:tn; tx=x; x=x-1;}{speak tx;}{write x;}
module V{listen tx:tn;}{read y:sn;}
    {if(y%tx != 0) tx=0;}{speak tx;}{write y;}
module W{listen tx:tn;}{read z:sn}
    {z=z-tx;}{speak nil;}{write z;}
module U1{listen nil;}{read x:sn;}
    {tx:tn; tx=-1;}{speak tx;}{write nil;}
module V1{listen tx:tn;}{read y:sn;}
    {null;}{speak tx;}{write nil;}
module W1{listen tx:tn;}{read z:sn}
    {null;}{speak nil;}{write z;}
```

Table 1: Modules for “perfect numbers” programs

The speculation on possible executions of the paths in a program, used to increase the processing speed, looks somehow similar to the prediction process in the HTM models of the brain. A comparison between these two computing models may be worthwhile for both fields. For instance, while the computer prediction is in the narrow window of the user program demand, the HTM models of the brain are more open, interactive, agent-like - here the prediction is mixed with possible actions of the human being who can change the course of the forthcoming input data. This may explain why humans are good on interactive tasks, while current computers with their predefined program-captured behavior are not.

We plan to develop the ideas from this paper in both directions: (1) to get a rich set of Agapia programs for HTMs models, particularly for those used by Numenta platform [9]; and (2) to explore the possibility of getting a running environment for Agapia programs on TRIPS computers [10].

Acknowledgments. Part of Camelia Chira’s work on this paper was carried out during a visit to the Department of Computer Science of the University of Illinois at Urbana-Champaign, which was supported by a grant from CNCSIS, Romania.

Gheorghe Stefanescu would like to thank Mircea Stan for pointers to TRIPS architectures. His work was partially supported by UIUC-NASA contract NNL08AA23C and by Romaninan PN-II GlobalComp grant.

References

[1] Burger, D., Keckler, S.W., McKinley, K.S., et al.: Scaling to the End of Silicon with EDGE Architectures. *IEEE Computer*, 37(7)(2004), 44-55.

[2] Chira, C.: "Multi-Agent Systems for Distributed Collaborative Design." Science Book Publishing House, Cluj 2007.

[3] Chira, C., Stefanescu, G.: "Interactive machines." Draft, November 2008.

[4] Dragoi, C., Stefanescu, G.: AGAPIA v0.1: A programming language for interactive systems and its typing systems. "Proc. FInCo/ETAPS 2007," *Electronic Notes in Theoretical Computer Science*, 203(3)(2008), 69-94.

[5] Goldin, D., Smolka, S., Wegner, P. (Eds.): "Interactive Computation: The New Paradigm." Springer, 2006.

[6] Hawkins, J. (with Blakeslee, S.). "On Intelligence." Times Books 2004.

[7] Popa, A., Sofronia, A., Stefanescu, G.: High-level structured interactive programs with registers and voices. *Journal of Universal Computer Science*, 13(11)(2007), 1722-1754.

[8] Stefanescu, G.: Interactive systems with registers and voices. *Fundamenta Informaticae*, 73(2006), 285-306.

[9] URL Numenta: `http://www.numenta.com/`

[10] URL TRIPS: `http://www.cs.utexas.edu/~trips/`

Gheorghe Stefanescu
Department of Computer Science, University of Illinois at Urbana-Champaign
N. Goodwin, Urbana, IL 61801, USA
E-mail: stefanes@cs.uiuc.edu

Camelia Chira
Department of Computer Science, Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca, Romania 400084
E-mail: fcchira@cs.ubbcluj.ro

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 211-223

Playing with Word Meanings

Dan Tufiş

Abstract: With the wide-world expansion of the social web, subjectivity analysis became lately one of the main research focus in the area of intelligent information retrieval. Being able to find out what people feel about a specific topic, be it a marketed product, a public person or a political issue, represents a very interesting application for a large class of actors, from the everyday product and service consumers, to the marketing and political analysts or decision-makers. In this paper we will diverge from the standard goal of subjectivity analysis i.e. computing the polarity of an opinionated sentence. Instead, we will try to detect those sentences of a given text that might have different connotations when put in different contexts. We describe a system called CONAN, initially designed to be a subjectivity analyzer, but recently extended towards detecting possible connotations shifts.

1 Introduction

With the wide-world expansion of the social web, subjectivity analysis became lately one of the main research focus in the area of information extraction from textual data. There is al-ready an abundant literature on this topic and here we mention only a few influential papers: (Hatzivassiloglou & McKeown, 1997), (Kamps & Marx, 2002), (Turney & Littman, 2002) (Yu & Hatzivassiloglou, 2003), (Kim & Hovy. 2004), (Grefenstette et al. 2004), (Wiebe et al. 2005), (Andreevskaia & Bergler, 2006), (Polanyi & Zaenen, 2006), (Tetsuya & Yu, 2007), (Mihalcea et al., 2007). Being able to find out what people feel about a specific topic, be it a marketed product, a public person or a political issue, represents a very interesting application for a large class of actors, from the everyday product and service consumers, to the marketing and political analysts or decision-makers.

There are various ways to model the processes of opinion mining and opinion classifications, and different granularities at which these models are defined (e.g. documents vs. sentences). For instance, in reviews classification one would try to assess the overall sentiment of an opinion holder with respect to a product (positive, negative or possibly neutral). However, the document level sentiment classification is too coarse for most applications as one text might express different opinions on different topics. Therefore the most advanced opinion miners are considering the paragraph and/or sentence level. At the paragraph/sentence level, the typical tasks include identifying the opinionated sentences and the opinion holder, deciding whether these opinions are related or not to a topic of interest and classifications according to their polarity (positive, negative, undecided) and force. At this level, however, the attitude assessment for a currently processed sentence might be strongly influenced by the discourse structure itself (Polanyi & Zaenen, 2006). Irony is a typical example of that case (e.g. "The river excursion was a disaster. John, who had this brilliant idea, forgot to tell us about the wonderful mosquitoes"). Thus, detecting discourse relations, for instance elaboration in the above example, might be a prerequisite, especially in processing narratives, for an accurate assessment of the subjectivity polarity of each sentence.

Irrespective of the methods and algorithms (which are still in their infancy) used in subjectivity analysis, they exploit the pre-classified words and phrases as opinion or sentiment bearing lexical units. Such lexical units (also called senti-words, polar-words) are manually specified, extracted from corpora or marked-up in the lexicons such as General Inquirer (Stone, et al., 1966), WordAffect (Valitutti,et al., 2004) or SentiWordNet (Esuli & Sebastiani, 2006) etc.

While opinionated status of a sentence is less controversial, its polarity might be rather problematic. The issue is generated by the fact that words are polysemous and the polarity of many senti-words depends on context (sometimes on local context, sometimes on global context). Apparently, bringing into discussion the notion of word sense (as SentiWordNet does) solves the problem but this is not so. As argued in (Tufiş, 2008), it is necessary to make a distinction between words intrinsically bearing a specific subjectivity/polarity and the words the polarity of which should be relationally considered. The latter case refers to the head-modifier relations; compare the different polarities of the modifier long in the two contexts: "the response time is long" vs. "the engine life is long".

Research in this area has been for some time monolingual, focused only on English, which is "mainly explained by the availability of resources for subjectivity analysis, such as lexicons and manually labeled corpora" (Mihalcea et al., 2007). Yet, in the recent years, there are more and more languages for which required resources are developed, essentially by exploiting parallel data and multilingual lexicons. With more than 40 monolingual wordnets[1] , most of them aligned to the Princeton WordNet (Fellbaum, 1998), the recent release of Sen-tiWordNet, and several public domain language independent tools for opinion mining and sentiment analysis, the multilingual research in opinion mining and sentiment analysis has been boosted and more and more sophisticated multilingual applications are expected in the immediate future. Although the mark-up in SentiWordNet is sometimes counter-intuitive (and most likely erroneous), it is currently one of the most useful resource for sentiment analysis.

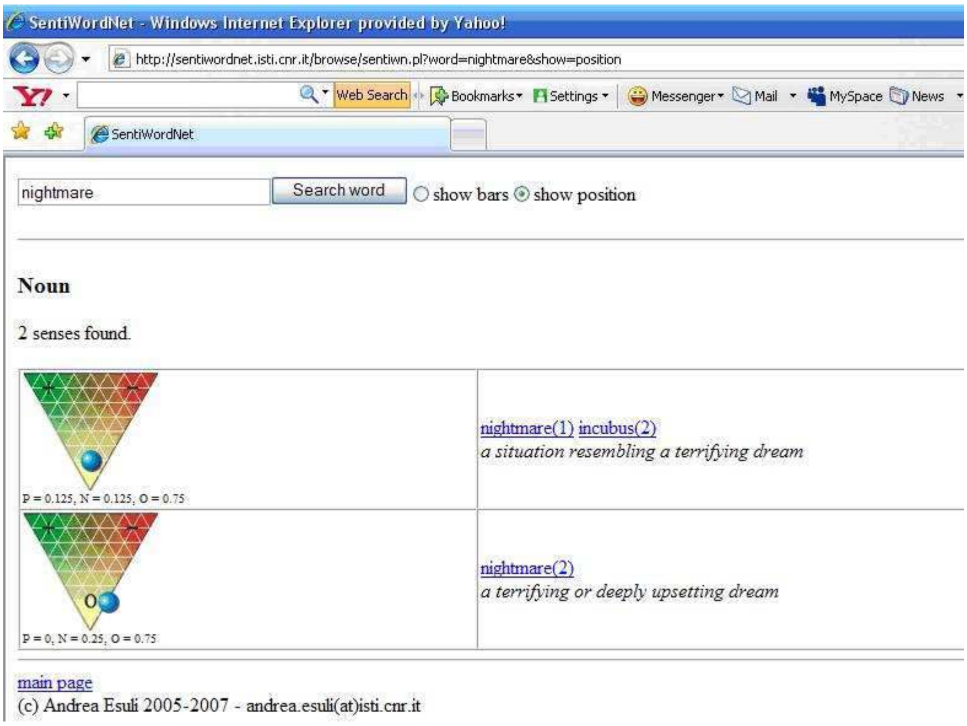
2 SentiWordNet

Among the latest enhancements of the PWN was the development of the SentiWordNet (Essuli & Sebastiani, 2006) an explicit annotation of the all the synsets with subjectivity mark-up. In SentiWordNet, each synset is associated with a triple $< P : \alpha, N : \beta O : \gamma >$ where P denotes its Positive subjectivity, N represents the Negative subjectivity and O stands for Objectivity. The values α, β and γ are sub-unitary numbers summing up to 1 and representing the degrees of positive, negative and objective prior sentiment annotation of the synset in case.

The values α, β and γ associated with each synset of the WordNet2.0 were computed by using machine leaning methods. The authors used two learners Rocchio (from Andrew McCallum's Bow package, available at http://www-2.cs.cmu.edu/˜mccallum/bow/) and SVM (version 6.01 of Thorsten Joachims' SVMlight, available at http://svmlight.joachims.org/) trained on four different data sets. They obtained thus 8 ternary classifiers and used a committee combination procedure to decide on the P, N and O values for each synset (see (Essuli & Sebastiani, 2006) for the details on the combination method and an evaluation of the results).

[1] http://www.globalwordnet.org/

The SentiWordNet graphical interface, exemplified in Figure 1, is available at http://sentiwordnet.isti.cnr.it/browse/. The Figure 1 shows that the subjectivity mark-up de-pends on the word senses. The sense 2 of the word *nightmare* (which denotes a *cognition noun*, subsumed by the term *psychological feature*) has a higher degree of negative subjectivity than sense 1 (which denotes a *state noun*, subsumed by the meaning of the synset < condition: 1, status: 2 >



SentiWordNet - Windows Internet Explorer provided by Yahoo!

http://sentiwordnet.isti.cnr.it/browse/sentiwn.pl?word=nightmare&show=position

nightmare | Search word | ○ show bars ● show position

Noun

2 senses found.

P = 0.125, N = 0.125, O = 0.75	nightmare(1) incubus(2) *a situation resembling a terrifying dream*
P = 0, N = 0.25, O = 0.75	nightmare(2) *a terrifying or deeply upsetting dream*

main page

(c) Andrea Esuli 2005-2007 - andrea.esuli(at)isti.cnr.it

Figure 1: SentiWordNet interface

The SentiWordNet is freely downloadable (for WordNet 2.0) from the address http://sentiwordnet.isti.cnr.it/download_1.0/ as a text file, containing on each line the follow-ing information (see Figure 2): the part of speech (POS) for the current synset, the synset off-set in the Princeton WordNet database, the Positive Score (PosScore), the Negative Score (NegScore) and the synset constituents. The Objective score can be obtained from the equation: ObjScore+PosScore+NegScore=1.

```
Pos offset  PosScoreNegScoreSynset Term
. . .
a 1006645 0.25    0.375   good#a#15 well#a#2
. . .
```

Figure 2: A Sample entry in the SentiWordnet distribution file

The SentiWordnet has been initially developed for English but the subjectivity information can be imported (via the translation equivalence relations) into any other language's wordnet which is aligned with Princeton WordNet or use it as an interlingual index. For instance in RoWordNet, the synset that corresponds to the one in Figure 2 is (bun:13.1)[2].

In (Tufiş 2008a) we presented the conclusions of the evaluation for the subjectivity mark-up imported in Romanian wordnet from the Princeton WordNet and showed that the imported subjectivity mark-up was almost always valid in Romanian. We used for the com-parison the SemCor bilingual corpus (EN-RO).

A wordnet with subjectivity annotation will be referred to as a *sentiwordnet.*

As mentioned before, although most of the synset subjectivity mark-ups in SentiWordNet are quite intuitive, there are many highly questionable subjectivity scores (as the one in Figure 2). Jaap Kamps and Maarten Marx (2002) describe a different but very interesting way of subjectivity labeling of WordNet lexical items. Their work is based on the pioneering 1957 paper *The Measurement of Meaning* by Charles Osgood, George J. Succi, and Percy H. Tan-nenbaum. Although, Kamp and Max (2002) assume a bag of words approach and ignore the sense-discrimination role in tagging the subjectivity of WordNet literals (only adjectives con-sidered) we found a relatively simple way to extend their method to take into account the word sense distinctions (Tufiş & Ştefąnescu, forthcoming). We plan to compare and maybe combine the adjectival synset annotations obtained by this method with the ones available in SentiWordNet, hoping to improve the current subjectivity mark-up. Although the new annotations would possibly change the results returned by the system to be described in the next section, the system itself would not require any significant modifications.

3 Analysis of potentially unwanted connotations

Many commercials make clever use of the language ambiguity (e.g. puns, surprising word associations, images pushing-up a desired interpretation context etc.) in promoting various products or services. Many of these short sentences, when used in regular texts, might have their connotations obliterated by the context and unnoticed by the standard reader. This observation is also valid the other way around: specific sentences, conceived in the context of a given text, when taken out of their initial context and placed in a conveniently chosen new context may convey a completely new (potentially unwanted) message/attitude. It is easy to find, especially in argumentative texts, examples of sentences which taken out of their context and maliciously used could have an adverse interpretation compared to the intended one.

The subjectivity and sentiment analysis methods are usually concerned with detecting whether a sentence is subjective and in case it is, establishing its polarity. This can be easily done with a sentiwordnet once every word in the sentence is sense disambiguated and the scope of each *valence shifter* (see below) has been established. Beside this, our application can be used for another potentially interesting problem: assessing whether a given sentence, taken out of context, may have different subjective interpretations. We estimate, on a [0,1] scale[3], the potential of a sentence being objective (O), positively

[2] In RoWordNet, unlike in Princeton WordNet, we preserved the sense-subsense distinctions given in the Reference Dictionary of Romanian (DEX). Therefore, the notation *bun:13.1* reads as the **first subsense of the 13th sense of 'bun'** (see for details (Tufiş et al. 2004)

[3] The scale for the new subjectivity mark-up, mentioned in the previous section, is different [-1, 1];

subjective (P) or negatively subjective (N), based on the senti-words in the respective sentence. Usually, these scores are uneven with one of them prevailing. We found that sentences which may have comparable subjective (positive or negative) scores are easier to use in a denotation/connotation shift game.

4 CONAN (CONnotation ANalyzer)

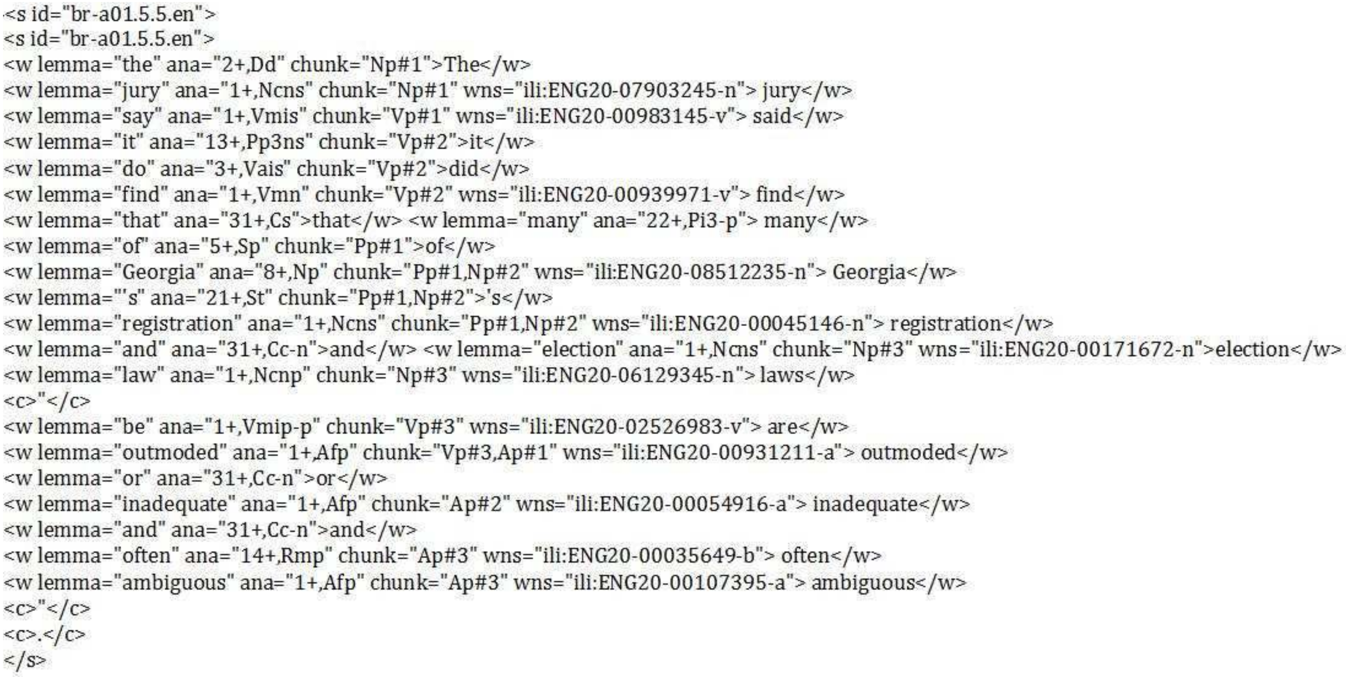
The CONAN system has been developed in a language independent way, and it should work for various languages, provided the analyzed texts are appropriately pre-processed and there are sentiwordnets available for the considered languages.

The necessary text preprocessing, required by CONAN includes: tokenization, tagging, lemmatization and chunking and, optionally, dependency linking. These fundamental operations for any meaningful natural language processing application have been largely described in previous publications and recently have been turned into public web-services (Tufiş et al., 2008) on our web server (http://nlp.racai.ro). Currently our linguistic web-services platform (which is based on standard web technology: SOAP/WSDL/UDDI) ensures processing for Romanian and English. In case CONAN is expected to work as a subjectivity scorer, the WSD processing is also required. In [Tufiş et al., 2004a] and [Ion & Tufiş, 2007] we described two different WSD systems, the first one for parallel corpora and the second one for monolingual corpora. However, in this paper we are interested in using CONAN as a means for detecting sentences which *potentially might have a different meaning* when put in a different context. We call this function of the system *connotation detection*. As a connotation detector, CONAN ignores the sense a senti-word may have in a given context. What it counts is whether the senti-word has senses of different subjectivity strength or even different polarity.

After the text is preprocessed as required, the second phase identifies all senti-words, i.e. those words which, in the associated sentiwordnet (in our case, either the Romanian or English one), have at least one possible subjective interpretation (that is, their objectivity score is less than 1). There has been mentioned by various authors that the bag-of-words (BoW) approaches to subjectivity analysis is not appropriate since the subjectivity priors (the lexicon mark-up subjectivity) may be changed in context by the so-called valence shifters (Polanyi & Zaenen, 2006): intensifiers, diminishers and negations. The first two operators increase and respectively decrease the subjectivity scores (both the negative and the positive ones) while the latter complements the subjective values. As the valence shifters do not necessary act on the senti-word in their immediate proximity, the chunking pre-processing step mentioned earlier is necessary for taking care of delimiting the scope of the operators' action. For instance in the sentence "He is NOT VERY smart", the word in italics (smart) is a (positive) senti-word, while the upper case words are valence shifters: NOT is a negation and VERY is an intensifier. The intensifier acts on the senti-word, while the negation act on the result of the intensifier: NOT(VERY(smart)). As a consequence, the sentence above has a negative subjectivity score. In (Tufiş, 2008) we showed that most wrong subjectivity mark-up existing in SentiWordNet can be explained due to a BoW approach to sense definitions analysis. The majority of synsets with wrong computed subjectivity markup have in their definitions valence shifters which apparently were

This difference might require some minor modification of the present CONAN's code.

ignored. This may explain why the sense of the word *happy* with the same meaning as *glad* (1st sense) is considered a subjective-negative adjective with a quite big score of 0.75.



```
<s id="br-a01.5.5.en">
<s id="br-a01.5.5.en">
<w lemma="the" ana="2+,Dd" chunk="Np#1">The</w>
<w lemma="jury" ana="1+,Ncns" chunk="Np#1" wns="ili:ENG20-07903245-n"> jury</w>
<w lemma="say" ana="1+,Vmis" chunk="Vp#1" wns="ili:ENG20-00983145-v"> said</w>
<w lemma="it" ana="13+,Pp3ns" chunk="Vp#2">it</w>
<w lemma="do" ana="3+,Vais" chunk="Vp#2">did</w>
<w lemma="find" ana="1+,Vmn" chunk="Vp#2" wns="ili:ENG20-00939971-v"> find</w>
<w lemma="that" ana="31+,Cs">that</w> <w lemma="many" ana="22+,Pi3-p"> many</w>
<w lemma="of" ana="5+,Sp" chunk="Pp#1">of</w>
<w lemma="Georgia" ana="8+,Np" chunk="Pp#1,Np#2" wns="ili:ENG20-08512235-n"> Georgia</w>
<w lemma="'s" ana="21+,St" chunk="Pp#1,Np#2">'s</w>
<w lemma="registration" ana="1+,Ncns" chunk="Pp#1,Np#2" wns="ili:ENG20-00045146-n"> registration</w>
<w lemma="and" ana="31+,Cc-n">and</w> <w lemma="election" ana="1+,Ncns" chunk="Np#3" wns="ili:ENG20-00171672-n">election</w>
<w lemma="law" ana="1+,Ncnp" chunk="Np#3" wns="ili:ENG20-06129345-n"> laws</w>
<c>"</c>
<w lemma="be" ana="1+,Vmip-p" chunk="Vp#3" wns="ili:ENG20-02526983-v"> are</w>
<w lemma="outmoded" ana="1+,Afp" chunk="Vp#3,Ap#1" wns="ili:ENG20-00931211-a"> outmoded</w>
<w lemma="or" ana="31+,Cc-n">or</w>
<w lemma="inadequate" ana="1+,Afp" chunk="Ap#2" wns="ili:ENG20-00054916-a"> inadequate</w>
<w lemma="and" ana="31+,Cc-n">and</w>
<w lemma="often" ana="14+,Rmp" chunk="Ap#3" wns="ili:ENG20-00035649-b"> often</w>
<w lemma="ambiguous" ana="1+,Afp" chunk="Ap#3" wns="ili:ENG20-00107395-a"> ambiguous</w>
<c>"</c>
<c>.</c>
</s>
```

Figure 3: XCES encoding of a sentence contained in an input file for CONAN.

CONAN takes input either from a file or from the keyboard. In case of input from a file, CONAN expects the text to be already preprocessed and encoded observing the encoding schema used by our linguistic web services platform (XCES compliant). In Figure 3, we ex-emplify the encoding of a sentence from the SEMCOR corpus processed by the RACAI web service platform. In case of keyboard input, the software detects whether the text (one or more sentences) is already in the XCES format. If not, a predefined workflow (based on by RACAI's linguistic web services) is invoked with the input text (assumed to be raw text) The subsequent processing is common for both input channels.

Based on the information available in the preprocessed form of the input, CONAN shallow parses sentences, one by one, producing a tree representation for each sentence. This structure is used to establish the scope of the valence shifters. In the current implementation a valence shifter (adjectives or adverbs which are constituents of adjective/adverbs chunks, see Figure 3) acts on the senti-words in the adjacent (either preceding or following) modified chunk (noun phrase, prepositional phrase or verb-phrase). The parsing structure is used for computing subjectivity scores for each chunk of the sentence and then to compute the overall score for the sentence. There are various interpretation modes of a sentence: the most objective reading, or the most subjective reading (either positive or negative). The interpretation mode is specified by the user when she/he loads (either from a file or from the keyboard) the text to be analyzed. This interpretation mode is used for all sentences in the input text.

For each sentence of the input text CONAN computes the score according to the user selected interpretation mode. To do that, the program detects all the senti-words in the current sentence and take into account the senses with the maximum scores with respect to the interpretation mode. The algorithm recursively calculates the interpretation scores for every node in a tree (taking into account the subjectivity operators and their scopes) by computing the average of its child nodes scores. Starting from the leaves, the scores propagate until the sentences scores are computed.

Figure 4 shows screenshots of the application with input taken from a file (exemplifying the sentence in Figure 1). The lower part (named CONAN) represents the main system inter-face and has three panels: the left panel displays the analysis of the sentences in the input file, according to the interpretation mode chosen by the user. The sentences are displayed ordered by the *selected polarity interpretation* scores (see below) or the *objectivity scores.* These scores are displayed in the middle panel. The score is preceded by the position of the current sentence in the original input file. The right panel is the area where the system displays the wordnet sense IDs and definitions for a user-selected word from the tree shown in the left panel. Selecting the Analysis tab from the CONAN control panel allows the user to locally change the interpretation mode for a selected sentence and/or to ask for words substitution in order to convey a (specified) different level of subjectivity. The user can ask for another two types of forced interpretations which will control the choice of the replacement words: polarity or objectivity oriented and polarity or objectivity opposable. For these complex types of interpretations, besides the sentence analysis and scoring, CONAN also makes word replace-ment suggestions.

To formalize, let's assume that the word w has n senses $s_1, s_2, \dots, s_n$, each of them listed in different synsets $S_1, S_2, \dots, S_n$. As we mentioned before, each synset is associated with a subjectivity mark-up represented by a triple $< P : \alpha N : \beta O : \gamma >$. Depending on the selected *objective/subjective interpretation I*, the value of only one field in this triple is considered (either α, β or γ). Given the word w and a selected interpretation I, let us assume that the highest score corresponds to the sense s_I listed in the synset S_I. For the meaning represented by the synset S_I, the word w may have several synonyms: $w^1_{S_I}, w^2_{S_I}, \dots, w^k_{S_I}$. Each of the k literals in this synset, may have also other meanings(corresponding to other senses, listed in different synsets $S^1_I, S^2_I, \dots, S^k_I$).

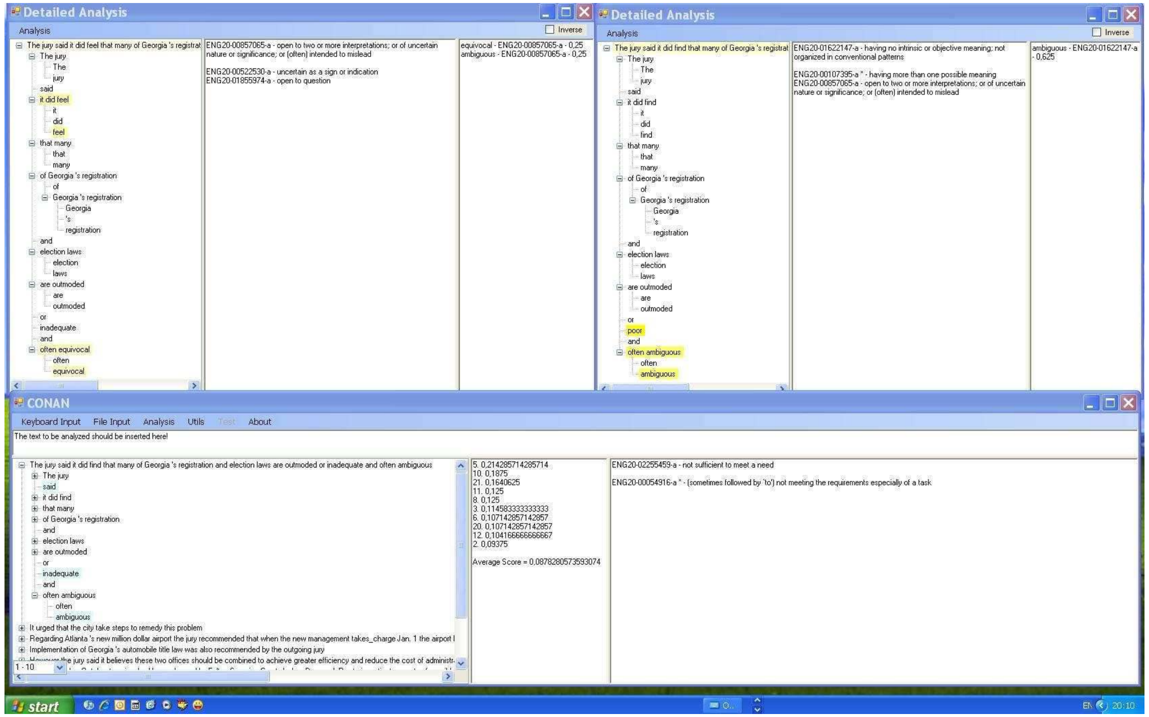
Lower panel: main screen displays the connotation analysis; Upper left panel: the most positive paraphrasing; Upper right panel: the most negative paraphrasing.

The algorithm searches for the literal in SI having one of his senses occurring in a synset that maximizes the expression below:

$$max(score(S_I), score(S^1_I), score(S^2_I), \dots, score(S^k_I)) \tag{1}$$

It may be the case that the literal with the property described by (1) is the very word w and in this situation no replacement will be performed. One should also notice that, in most of the cases, the word replacement, based on the equation (1), may significantly change the meaning of the current sentence. This is practically what the user that employs this option aims for. He/She wants some his/her text be perceived by the readers in a certain interpretation. Once the replacement word has been identified in the sentiwordnet, it must be generated in the inflected form required by its occurrence in the sentence.

This generation process is performed by using the contextual grammatical information (number, case, tense, etc) associated with the original word (contained in its morpho-syntactic description -MSD- produced by the tagger). The new lemma and the MSD of the



```
⊟ The jury said it did feel that many of Georgia 's registrat
   ⊟ The jury
      The
      jury
   said
   ⊟ it did feel
      it
      did
      feel
   ⊟ that many
      that
      many
   ⊟ of Georgia 's registration
      of
      ⊟ Georgia 's registration
         Georgia
         's
         registration
   and
   ⊟ election laws
      election
      laws
   ⊟ are outmoded
      are
      outmoded
   or
   inadequate
   and
   ⊟ often equivocal
      often
      equivocal
```

ENG20-00857065-a - open to two or more interpretations; or of uncertain nature or significance; or (often) intended to mislead

ENG20-00522530-a - uncertain as a sign or indication
ENG20-01855974-a - open to question

equivocal - ENG20-00857065-a - 0,25
ambiguous - ENG20-00857065-a - 0,25

```
⊟ The jury said it did find that many of Georgia 's registrat
   ⊟ The jury
      The
      jury
   said
   ⊟ it did find
      it
      did
      find
   ⊟ that many
      that
      many
   ⊟ of Georgia 's registration
      of
      ⊟ Georgia 's registration
         Georgia
         's
         registration
   and
   ⊟ election laws
      election
      laws
   ⊟ are outmoded
      are
      outmoded
   or
   poor
   and
   ⊟ often ambiguous
      often
      ambiguous
```

ENG20-01622147-a - having no intrinsic or objective meaning; not organized in conventional patterns

ENG20-00107395-a * - having more than one possible meaning
ENG20-00857065-a - open to two or more interpretations; or of uncertain nature or significance; or (often) intended to mislead

ambiguous - ENG20-01622147-a - 0,625

The text to be analyzed should be inserted here!

```
⊟ The jury said it did find that many of Georgia 's registration and election laws are outmoded or inadequate and often ambiguous
   ⊞ The jury
   said
   ⊞ it did find
   ⊞ that many
   ⊞ of Georgia 's registration
   and
   ⊞ election laws
   ⊞ are outmoded
   or
   inadequate
   and
   ⊟ often ambiguous
      often
      ambiguous
⊞ It urged that the city take steps to remedy this problem
⊞ Regarding Atlanta 's new million dollar airport the jury recommended that when the new management takes_charge Jan. 1 the airport l
⊞ Implementation of Georgia 's automobile title law was also recommended by the outgoing jury
⊞ However the jury said it believes these two offices should be combined to achieve greater efficiency and reduce the cost of administr
```

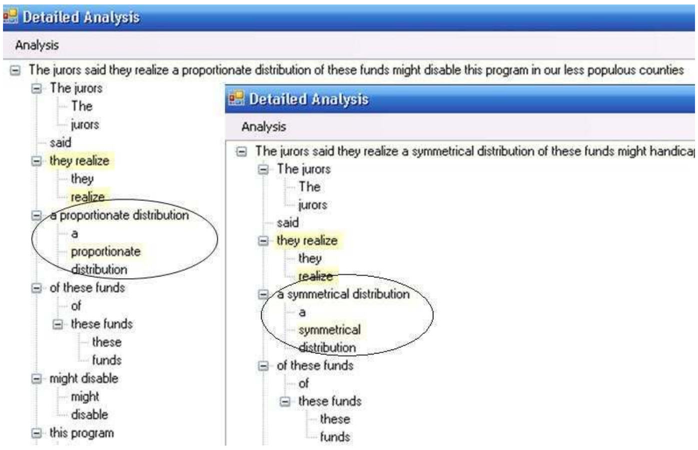
```
5. 0,214285714285714
10. 0,1875
21. 0,1640625
11. 0,125
8. 0,125
3. 0,114583333333333
6. 0,107142857142857
20. 0,107142857142857
12. 0,104166666666667
2. 0,09375

Average Score = 0,0878280573593074
```

ENG20-02255459-a - not sufficient to meet a need

ENG20-00054916-a * - (sometimes followed by `to') not meeting the requirements especially of a task

Figure 4: CONAN in action.



```
⊟ The jurors said they realize a proportionate distribution of these funds might disable this program in our less populous counties
  ⊟ The jurors
      The
      jurors
    said
  ⊟ they realize
      they
      realize
  ⊟ a proportionate distribution
      a
      proportionate
      distribution
  ⊟ of these funds
      of
    ⊟ these funds
        these
        funds
  ⊟ might disable
      might
      disable
  ⊟ this program
```

```
⊟ The jurors said they realize a symmetrical distribution of these funds might handica
  ⊟ The jurors
      The
      jurors
    said
  ⊟ they realize
      they
      realize
  ⊟ a symmetrical distribution
      a
      symmetrical
      distribution
  ⊟ of these funds
      of
    ⊟ these funds
        these
        funds
```

Figure 5: Word changed due to a forced subjective positive interpretation.

previous word are sufficient for producing the adequate inflected form of the replacement word. While for English, this generation phase is trivial, for a highly inflectional language, the inflectional form generation is a must in order to preserve the grammaticality of the output text.

Figure 5 presents an example in which a word had been changed due to a forced *subjective positive* interpretation.

The second complex type of interpretation, *polarity or objectivity opposable interpretation* is dealing with the minimizing the interpretation possibilities for the current sentence. The difference between this option and the previous one resides in the way the selection for the appropriate literals is done. The algorithm selects the literal having the meaning with the lowest (not highest) score, in the inverse (not current) interpretation, among all the other meanings. In other words, the winning literal must satisfy the equation (2):

$$min(score(S_I), score(S_I^1), score(S_I^2), \ldots, score(S_I^k)) \tag{2}$$

The rationale is that we want to select synonyms for the current words that can avoid interpretations of a certain polarity (better than the current words do it).

In addition, by the use of shades of color for the nodes (words) of the dependence parse, the user may easily observe the words that play a role in the possible meaning shifts of the current sentence. The more intense the color of a node, the higher its subjectivity score. A simple click on a node displays its score.

Another thing CONAN can do is to compute the *interpretative score* for a sentence or for a text as the average of the sentences' interpretative scores.

We define the interpretability score (IS) as a quantitative measure of the potential for connotation shift of a sentence. It is computed as a normalized sum of the interpretability scores of the senti-words (sw) of the considered sentence as described in the equations below:

$$IS(sentence_k) = \frac{0.5 * (maxP(sentence_k) + maxN(sentence_k))}{1 + |maxP(sentence_k) - maxN(sentence_k)|} \tag{3}$$

with |senti-words| representing the number of the senti-words in the current sentence

$$IS(sw_k) = \frac{0.5 * (maxP(sw_k) + maxN(sw_k))}{1 + |maxP(sw_k) - maxN(sw_k)|} \tag{4}$$

with $maxP(sw_k)$ and $maxN(sw_k)$ representing the highest positive and negative scores among the senses of (sw_k) senti-word.

Evidently, the interpretative score can be computed only after the interpretation scores for positivity and negativity are calculated. The rationale for this empirical formula resides in the fact that when a senti-word has one sense highly positive and another one highly negative and these values are comparable, the respective word is a major constituent for a possible connotation shift of the sentence in which it appeared. The interpretability score of a senti-word is maximum (1) when it has one exclusively positive sense ($P(sw_k) = 1$) and another sense which is exclusively negative ($N(sw_k) = 1$). For the current SentiWordNet annotations, the senti-words with the highest interpretability score ($IS = 0,875$) are pretty, immoral and gross. The valence shifters (intensifiers, diminishers and negations) are specified in three external text files (user editable) which are read-in each time CONAN is launched. Currently all the valence shifters are uniformly dealt with, irrespective of the arguments they take: the intensifiers and diminishers increase or decrease with 20% the score of their argument (senti-word or senti-group-phrase) while the negations switch the P/N scores of their arguments. Therefore, the valence shifter files are simple lists of words. A more elaborated approach (under development) will specify for each valence shifter, its grammar category its sense number (if necessary) and preferred argument-type as well as an argument-sensitive valence shifting function.

Concerning the valence shifters, it is interesting to note that, in general, translation equivalence preserves their type distinctions. However this is not always true. For instance, in Romanian destul (either adjective or adverb), when followed by the preposition de, is arguably a diminisher. In English, its translation equivalent enough acts more like an intensifier than as a diminisher.

5 Conclusions

Most of our experiments were performed on the SEMCOR corpus[4] , which was translated in Romanian and aligned at both sentence and word level. Our working version has 8146 sentences, in both languages, the analysis of which, independent of the analysis type, takes only a couple of minutes. We analyzed the texts in both languages and observed that the IS values for aligned sentences slightly differ. One possible explanation is that due to much larger coverage (especially in terms of adjectives and adverbs) of the Princeton WordNet as compared to the Romanian Wordnet the numbers of the identified senti-words in the aligned sentences were frequently different. Additionally, the literals in Romanian WordNet have currently less senses than their English counterparts. We also noticed several strange subjectivity annotations (especially for the adjectives) in the SentiWordnet distribution which strongly contradicts the common sense.

Future research will be dedicated to a new method for assigning subjectivity scores to the adjectives, to further extend the Romanian Wordnet and a thorough cross-lingual evaluation of the bilingual SEMCOR processing. One very interesting research avenue for

[4] http://www.cs.unt.edu/ rada/downloads.html

dealing with prior mark-up for senti-words as well as computing their contextual subjective scores is represented by recent work of Lotfi Zadeh (2008) on precisiated natural language and his paradigm on computation with words.

References

[1] Alina Andreevskaia, Sabine Bergler (2006). Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings EACL-06 the 11rd Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, pp. 209-216.

[2] Andrea A. Esuli, F. Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. Proceedings of the 5th Conference on Language Resources and Evaluation LREC-06, Genoa, Italy, pp. 417-422.

[3] Christiane Fellbaum, editor. 1998. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA.

[4] Radu Ion, Dan Tufiş: "Meaning Affinity Models". In Eneko Agirre, Lluìs Màrquez and Richard Wicentowski (eds.): "Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)", Prague, Czech Republic, Association for Computational Linguistics, June 2007, pp. 282-287.

[5] Mihalcea Rada, C. Banea, Wiebe Janyce. 2007. Learning Multilingual Subjective Language via Cross-Lingual Projections. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics,.Prague, Czech Republic, June, pp. 976-983.

[6] Livia Polanyi, Annie Zaenen 2006. Contextual Valence Shifters. In James G. Shanahan, Yan Qu, and Janyce Wiebe, editors, Computing Attitude and Affect in Text: Theory and Appli-cation. Springer Verlag.

[7] Dan Tufiş, Radu Ion, Nancy Ide: Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets (2004a). In proceedings of the 20th International Conference on Computational Linguistics, COLING2004, Geneva, 2004, pp. 1312-1318, ISBN 1-9324432-48-5.

[8] Dan Tufiş, Eduard Barbu, Verginica Barbu-Mititelu, Radu Ion and Luigi Bozianu (2004b). The *Romanian Wordnet. In Dan Tufi (ed.), Romanian Journal on Information Science and Technology*. Special Issue on BalkaNet, volume 7, pp. 105-122. Romanian Academy, April 2004. ISSN 1453-8245.

[9] Dan Tufiş, Radu Ion, Luigi Bozianu, Alexandru Ceauşu, Dan Ştefănescu (2008). Romanian WordNet: Current State, New Applications and Prospects. In Attila Tanacs, Dora Csendes, Veronika Vincze, Christiane Fellbaum, Piek Vossen (eds.): *Proceedings of 4th Global WordNet Conference, GWC-2008*, University of Szeged, Hungary, January 22-25, pp. 441-452

[10] D. Tufiş, R. Ion, A. Ceauşu, D. Ştefănescu 2008. RACAI's Linguistic Web Services. In Proceedings of 6th Conference on Language Resources and Evaluation LREC-08, Marrakech, Marocco.

[11] D. Tufiş 2008a. Subjectivity mark-up in WordNet: does it work cross-lingually? A case study on Romanian Wordnet. Invited talk on the Panel "Wordnet Relations" at the Global Word-Net Conference, January 22-25, 2008.

[12] D. Tufiş 2008b. Mind Your Words! You Might Convey What You Wouldn't Like To. Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844Vol. III (2008), Suppl. issue: Proceedings of ICCCC 2008, pp. 139-144

[13] D. Tufiş, D. Ştefănescu (forthcoming). Words with Attitude Revisited. RACAI Research Report, November 2008 (to be published in Romanian Journal on Science and Technology of Information, Romanian Academy)

[14] D. Fox, W. Burgard, H. Kruppa and S. Thrun, A probabilistic approach to collaborative multi-robot localization. Autonomous Robots 8(3), 2000.

[15] Gregory Grefenstette, Yan Qu, David A. Evans, and James G. Shanahan. 2004. Validating the Coverage of Lexical Resources for Affect Analysis and Automatically Classifying New Words along Semantic Axes. In Yan Qu, James Shanahan, and Janyce Wiebe, editors, *Exploring Attitude and Affect in Text: Theories and Applications*, AAAI-2004 Spring Symposium Series, pages 71-78.

[16] Vasileios Hatzivassiloglou and Kathleen B. McKeown. 1997. Predicting the Semantic Orientation of Adjectives. In *35th* ACL, pages 174-181.

[17] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD-04*, pages 168-177.

[18] Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. 2004. UsingWord-Net to measure semantic orientation of adjectives. In *LREC 2004*, volume IV, pages 1115-1118.

[19] Jaap Kamps and Maarten Marx, 2002. Words with attitude, In Proceedings of the 1st International WordNet Conference, Mysore, India, pp. 332-341.

[20] Soo-Min Kim and Edward Hovy. 2004. Determining the sentiment of opinions. In *COLING-2004*, pages 1367-1373, Geneva, Switzerland.

[21] Adrienne Lehrer. 1974. *Semantic Fields and Lexical Structure.* North Holland, Amsterdam and NewYork.

[22] Eleanor Rosch. 1978. Principles of Categorization. In Eleanor Rosch and Barbara B. Lloyd, editors, *Cognition and Categorization*, pages 28-49. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[23] P. J. Stone, D. C. Dumphy, M. S. Smith and D. M. Ogilvie. 1966. *The General Inquirer: a computer approach to content analysis.* M.I.T. studies in comparative politics. M.I.T. Press, Cambridge,MA.

[24] Pero Subasic and Alison Huettner. 2001. Affect Analysis of Text Using Fuzzy Typing. *IEEE-FS*, 9:483-496.

[25] Peter Turney and Michael Littman. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report ERC-1094 (NRC 44929), National Re-search Council of Canada.

[26] Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. 2004. Developing Affective Lex-ical Resources. *PsychNology Journal*, 2(1):61-83.

[27] Hong Yu and Vassileios Hatzivassiloglou. 2003. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-03).*

[28] Lotfi A. Zadeh. 1975. Calculus of Fuzzy Restrictions. In L.A. Zadeh, K.-S. Fu, K. Tanaka, and M. Shimura, editors, *Fuzzy Sets and their Applications to cognitive and decision processes*, pages 1-40. Academic Press Inc., New-York.

[29] Lotfi A. Zadeh. 1987. PRUF - a Meaning Representation Language for Natural Languages. In R.R. Yager, S. Ovchinnikov, R.M. Tong, and H. T. Nguyen, editors, *Fuzzy Sets and Ap-plications: Selected Papers by L. A. Zadeh*, pages 499-568. John Wiley & Sons.

[30] Lotfi Zadeh. 2008. A New Frontier in Computation - Computation Described in Natural Lan-guage. Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. III (2008), Suppl. issue: Proceedings of ICCCC 2008, pp. 26-27

[31] Wiebe Janyce, Wilson Theresa, Cardie Claire (2005). Annotating Expressions and Emotions in Language. In Language, Resources and Evaluation, vol. 39, No. 2/3, pp. 164-210

[32] Miyoshi Tetsuya, Nakagami Yu (2007). Sentiment classification of customer reviews on electric products. IEEE International Conference on Systems, Man and Cybernetics, pp. 2028-2033

[33] Charles E. Osgood, George J. Succi and Percy H. Tannenbaum. 1957. The Measurement of Meaning. University of Illinois Press, Urbana IL.

Dan Tufiş
Research Institute for Artificial Intelligence
Bucharest, Romania
E-mail: tufis@racai.ro

Zadeh L.A., Tufiş D., Filip F.G., Dziţac I.(Eds.),
From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence,
Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008, pp. 224-225

A New Frontier in Computation - Computation with Information Described in Natural Language

Lotfi A. Zadeh

Department of EECS, University of California, Berkeley, CA 94720-1776;
Telephone: 510-642-4959; Fax: 510-642-1712; E-Mail: zadeh@eecs.berkeley.edu.
Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Omron Grant, Tekes Grant, Chevron Texaco Grant and the BISC Program of UC Berkeley.

Extended Abstract

What is meant by Computation with Information Described in Natural Language, or NL-Computation, for short? Does NL-Computation constitute a new frontier in computation? Do existing bivalent-logic-based approaches to natural language processing provide a basis for NL-Computation? What are the basic concepts and ideas which underlie NL-Computation? These are some of the issues which are addressed in the following.

What is computation with information described in natural language? Here are simple examples. I am planning to drive from Berkeley to Santa Barbara, with stopover for lunch in Monterey. It is about 10 am. It will probably take me about two hours to get to Monterey and about an hour to have lunch. From Monterey, it will probably take me about five hours to get to Santa Barbara. What is the probability that I will arrive in Santa Barbara before about six pm? Another simple example: A box contains about twenty balls of various sizes. Most are large. What is the number of small balls? What is the probability that a ball drawn at random is neither small nor large? Another example: A function, f, from reals to reals is described as: If X is small then Y is small; if X is medium then Y is large; if X is large then Y is small. What is the maximum of f? Another example: Usually the temperature is not very low, and usually the temperature is not very high. What is the average temperature? Another example: Usually most United Airlines flights from San Francisco leave on time. What is the probability that my flight will be delayed?

Computation with information described in natural language is closely related to Computing with Words. NL-Computation is of intrinsic importance because much of human knowledge is described in natural language. This is particularly true in such fields as economics, data mining, systems engineering, risk assessment and emergency management. It is safe to predict that as we move further into the age of machine intelligence and mechanized decision-making, NL-Computation will grow in visibility and importance.

Computation with information described in natural language cannot be dealt with through the use of machinery of natural language processing. The problem is semantic imprecision of natural languages. More specifically, a natural language is basically a system for describing perceptions. Perceptions are intrinsically imprecise, reflecting the bounded ability of sensory organs, and ultimately the brain, to resolve detail and store information. Semantic imprecision of natural languages is a concomitant of imprecision of perceptions.

Our approach to NL-Computation centers on what is referred to as generalized-constraint-based computation, or GC-Computation for short. A fundamental thesis which underlies NL-Computation is that information may be interpreted as a generalized constraint. A generalized constraint is expressed as X isr R, where X is the constrained variable, R is a constraining relation and r is an indexical variable which defines the way in which R constrains X. The principal constraints are possibilistic, veristic, probabilistic, usuality, random set, fuzzy graph and group. Generalized constraints may be combined, qualified, propagated, and counter propagated, generating what is called the Generalized Constraint Language, GCL. The key underlying idea is that information conveyed by a proposition may be represented as a generalized constraint, that is, as an element of GCL.

In our approach, NL-Computation involves three modules: (a) Precisiation module; (b) Protoform module; and (c) Computation module. The meaning of an element of a natural language, NL, is precisiated through translation into GCL and is expressed as a generalized constraint. An object of precisiation, p, is referred to as precisiend, and the result of precisiation, p^*, is called a precisiand. Usually, a precisiend is a proposition, a system of propositions or a concept. A precisiend may have many precisiands. Definition is a form of precisiation. A precisiand may be viewed as a model of meaning. The degree to which the intension (attribute-based meaning) of p* approximates to that of p is referred to as cointension. A precisiand, p^*, is cointensive if its cointension with p is high, that is, if p^* is a good model of meaning of p.

The Protoform module serves as an interface between Precisiation and Computation modules. Basically, its function is that of abstraction and summarization.

The Computation module serves to deduce an answer to a query, q. The first step is precisiation of q, with precisiated query, q^*, expressed as a function of n variables $u_1, \ldots, u_n$. The second step involves precisiation of query-relevant information, leading to a precisiand which is expressed as a generalized constraint on $u_1, \ldots, u_n$. The third step involves an application of the extension principle, which has the effect of propagating the generalized constraint on $u_1, \ldots, u_n$ to a generalized constraint on the precisiated query, q^*. Finally, the constrained q^* is interpreted as the answer to the query and is retranslated into natural language.

The generalized-constraint-based computational approach to NL-Computation opens the door to a wide-ranging enlargement of the role of natural languages in scientific theories. Particularly important application areas are decision-making with information described in natural language, economics, systems engineering, risk assessment, qualitative systems analysis, search, question-answering and theories of evidence.

Lotfi A. Zadeh
Professor in the Graduate School and Director
Berkeley Initiative in Soft Computing (BISC), Computer Science Division
Department of EECS, University of California
Berkeley, CA 94720-l776; Telephone: 5l0-642-4959; Fax: 5l0-642-l7l2
E-mail: zadeh@eecs.berkeley.edu
http://www.cs.berkeley.edu/ zadeh/

Author index

Ahmed S., 129

Bălaş M., 11, 22
Bălaş V., 11, 22
Buciu I., 32

Castañeda C., 172
Chira C., 51, 196
Cordova F.M., 63

Dumitrescu D., 51
Dziţac I., 84
Dziţac S., 84

Felea I., 84
Fodor J., 185

Lefranc G., 100

Naforniţă I., 32
Nikolova N., 129
Nuñez R., 172

Ortega A., 172

Pintea C-M., 51

Quezada L.E., 63

Radojevic D.R., 150
Rosal E., (del), 172
Rudas I.J., 185

Stefanescu G., 196

Tenekedjiev K., 129
Toneva D., 129
Tufiş D., 211

Vesselenyi T., 84

Zadeh L.A., 224