I T H E A

International Journal INFORMATION THEORIES & APPLICATIONS

2013 Volume 20 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
Volume 20 / 2013, Number 3

International Journal "INFORMATION THEORIES & APPLICATIONS" (IJ ITA)
is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society

IJ ITA welcomes scientific papers connected with any information theory or its application.
IJ ITA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITA are given on *www.ithea.org*.

Responsibility for papers published in IJ ITA belongs to authors.

General Sponsor of IJ ITA is the **Consortium FOI Bulgaria** (*www.foibg.com*).

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol. 20, Number 3, 2013

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
Institute of Mathematics and Informatics, BAS, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politécnica de Madrid, Spain,
Hasselt University, Belgium,
St. Petersburg Institute of Informatics, RAS, Russia,
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia

Printed in Bulgaria
Publisher ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: **Ina Markova**

ISSN 1310-0513 (printed) **ISSN 1313-0463 (online)** **ISSN 1313-0498 (CD/DVD)**

RESERVOIR FORECASTING NEURO-FUZZY NETWORK AND ITS LEARNING

Yevgeniy Bodyanskiy, Oleksii Tyshchenko, Iryna Pliss

***Abstract**: The idea of Reservoir Computing has become so popular recently due to its computational efficiency and the fact that just only a linear output layer must be taught. In this paper a reservoir is proposed to be built using neo-fuzzy neurons which means that Reservoir Computing and paradigms and fuzzy logic are combined.*

***Keywords**: reservoir computing, hybrid systems, neo-fuzzy neuron, online learning procedure, prediction.*

***ACM Classification Keywords**: I.2.6 Learning – Connectionism and neural nets.*

Introduction

Training a recurrent neural network is not an easy task to do, it has been offered lately to construct a custom recurrent topology to train a single linear output layer. Nowadays it can be easily performed with the help of Reservoir Computing [Jaeger, 2001a; Jaeger, 2001b]. Reservoirs are randomly created, and the exact weight distribution and sparsity have very limited influence on the reservoir's performance.

Reservoir Computing usually consists of Echo State Networks [Alexandre, 2009], Liquid State Machines and the Backpropagation Decorrelation learning rule. The most important part of tuning weights in traditional learning procedures of recurrent neural networks is the output layer weight tuning.

Reservoir neural networks were designed to have the same computational power as traditional recurrent neural networks except the fact that there's no need to train internal weights.

The reservoir is a recurrent neural network which has n input units, h internal units of the hidden layer and m output units. Input elements at a time point k form a vector $\mathrm{x}(\mathrm{k}) = \left(x_1(\mathrm{k})\ldots x_n(\mathrm{k})\right)^T$, internal units form a vector $\tilde{\mathrm{s}}(\mathrm{k}) = \left(\tilde{\mathrm{s}}_1(k)\ldots\tilde{\mathrm{s}}_h(k)\right)^T$ and finally output units form a vector $y(\mathrm{k}) = \left(y_1(\mathrm{k})\ldots y_m(\mathrm{k})\right)^T$.

Real-valued connection weights are collected in a $(h\times n)$ weight matrix W_{in} for the input weights, in an $(h\times h)$ matrix W_{res} for the internal connections, in an $m\times(h+n+m)$- matrix W_{out} for the connections to the output units, and in a $(n\times m)$-matrix W_{back} for the connections that feed back from the output to the internal units. The weight matrix W_{in} is created randomly. This matrix can be either full or sparsed. The weight matrix W_{res} is usually created according to Gaussian distribution. It can be easily explained by the fact that a network creates a reservoir full of different nonlinear current and preceding values (a so-called "echo"). Connections directly from the input to the output units and connections between output units are allowed. A reservoir should have a suitable perceptibility. The simplest way to make it is to tune a spectral radius (the biggest eigenvalue) of weight matrix W_{res}.

The activation of internal units is updated according to expression

$$\tilde{\mathrm{s}}(k+1) = \mathrm{f}\left(\mathrm{W}_{\mathrm{in}}x(\mathrm{k}+1)+\mathrm{W}_{res}\tilde{\mathrm{s}}(\mathrm{k})+\mathrm{W}_{\mathrm{back}}\mathrm{y}(\mathrm{k})\right), \quad (1)$$

where $\tilde{s}(k+1)$ is a vector of reservoir states at a time point k, where $f = (f_1 \dots f_P)$ – the internal unit's output functions (typically sigmoid functions).

The output is computed according to equation

$$y(k+1) = f_{out}\left(W_{out}\begin{bmatrix}\tilde{s}(k+1)\\ x(k+1)\end{bmatrix}\right), \quad (2)$$

where $f_{out} = (f_{out1} \dots f_{outT})$ – the output unit's output functions.

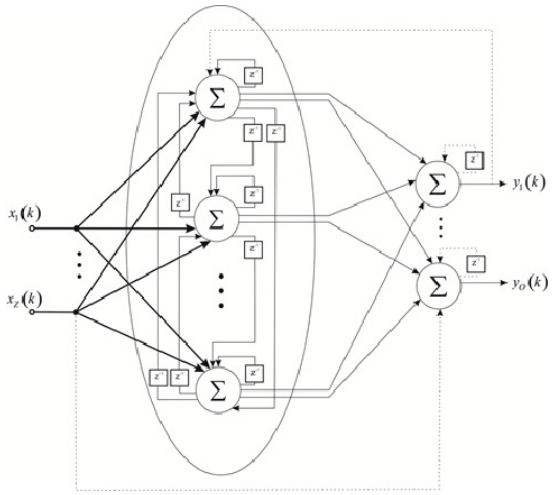


Figure 1. Representation of a reservoir network. Large circles are the neurons and small ones represent the input data. The hidden layer is the reservoir. Dashed connections are optional

In the Echo-State Networks literature, the reservoirs are rescaled using measures based on stability bounds. Several of these measures have been presented in [Jaeger, 2001b]. Jaeger has proposed the spectral radius to be slightly lower than one (because the reservoir is then guaranteed to have the echo state property).

The most interesting thing about reservoirs is that all weight matrices to the reservoir are initialized randomly, while all connections to the output are trained. When using the system after training with the "reservoir-output" connections, the computed output by is fed back into the reservoir.

Although many different neuron types have already been used in reservoirs but there is no yet clear understanding which node types are optimal for given applications. The necessary and sufficient conditions for a recurrent network to have echo states are based on the spectral radius and the largest singular value of the connection matrix. The echo state property means that the current state of the network is uniquely determined by the network input up to now. The network is thus state forgetting.

Network Architecture

In this paper a new architecture of a forecasting neural network is proposed which is built with the help of neo-fuzzy neurons (NFN) [Horio, 2001] and a layer of delay elements. The proposed network was called a reservoir forecasting neuro-fuzzy network in [Bodyanskiy, 2009; Bodyanskiy, 2010]. The scheme of the proposed network is on fig. 2.

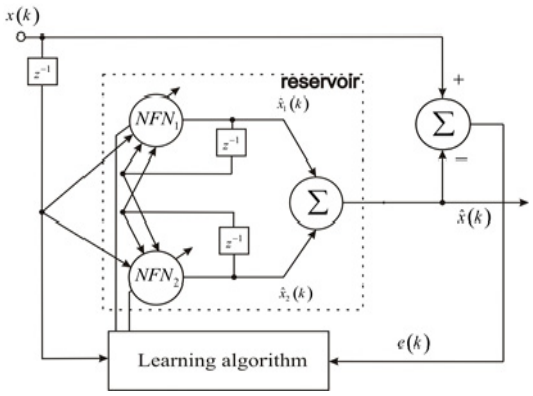


Figure 2. The architecture of the reservoir forecasting neuro-fuzzy network

$x(k)$ is an input value of the network, $\hat{x}(k)$ is a network output, $k = 1, 2, \ldots, N, \ldots$ - current discrete time.

Neo-fuzzy neurons were proposed by T. Yamakawa and co-authors [Yamakawa, 1992]. These constructions are neuronal models with nonlinear synapses. The output of the nonlinear synapse neuron is obtained by sum of the outputs of the synapses represented by nonlinear functions. They can approximate a nonlinear input-output relationship by one neuron, and there is no local minimum problem in learning [Uchino, 1997; Miki, 1999]. Fig. 3 shows a structure of the conventional NFN.

An input signal $x(k)$ is fed into the NFN layer. It should be mentioned that this construction has back connections which come through the layer of delay elements from NFN outputs back to their inputs [Tyshchenko, 2012].

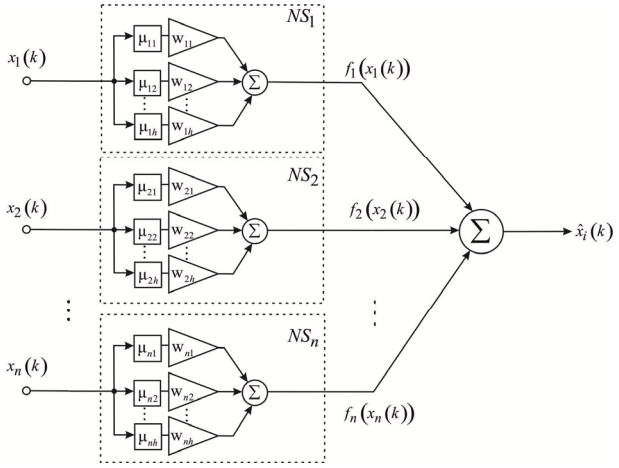


Figure 3. The structure of the conventional neo-fuzzy neuron

A network output is calculated in the form

$$\hat{x}(k) = \hat{x}_1(k) + \hat{x}_2(k) = \sum_{l=1}^{2} f^{[l]}(x(k-1)) + f^{[l]}(\hat{x}_1(k-2)) + f^{[l]}(\hat{x}_2(k-2)) =$$
$$= \sum_{l=1}^{2}\sum_{i=1}^{3}\sum_{j=1}^{h} \mu_{ij}^{[l]}\left(x(k-1)\right) w_{ij}^{[l]}(k) + \mu_{ij}^{[l]}\left(\hat{x}_1(k-2)\right) w_{ij}^{[l]}(k) + \mu_{ij}^{[l]}\left(\hat{x}_2(k-2)\right) w_{ij}^{[l]}(k), \quad (3)$$

where $\hat{x}_1, \hat{x}_2$ – outputs of NFN_1 and NFN_2 correspondingly;

$\mu_{ij}^{[l]}$ – membership functions of NFN_l;

$w_{ij}^{[l]}$ – synaptic weights of NFN_l;

h - a number of membership functions in a nonlinear synapse.

The architecture of the special NFN used in our case is presented on fig. 4. Here

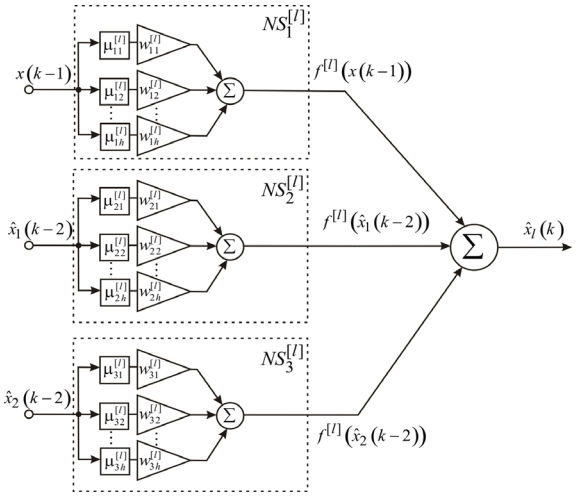


Figure 4. The special neo-fuzzy neuron's architecture for the reservoir forecasting network

$$\hat{x}_l(k) = f^{[l]}(x(k-1)) + f^{[l]}(\hat{x}_1(k-2)) + f^{[l]}(\hat{x}_2(k-2)), \tag{4}$$

$$f^{[l]}(x(k-1)) = \sum_{i=1}^{3}\sum_{j=1}^{h} \mu_{ij}^{[l]}\left(x(k-1)\right) w_{ij}^{[l]}\left(k\right), \tag{5}$$

$$f^{[l]}(\hat{x}_1(k-2)) = \sum_{i=1}^{3}\sum_{j=1}^{h} \mu_{ij}^{[l]}\left(\hat{x}_1(k-2)\right) w_{ij}^{[l]}\left(k\right), \tag{6}$$

$$f^{[l]}(\hat{x}_2(k-2)) = \sum_{i=1}^{3}\sum_{j=1}^{h} \mu_{ij}^{[l]}\left(\hat{x}_2(k-2)\right) w_{ij}^{[l]}\left(k\right). \tag{7}$$

Membership functions usually form a set of functions similar to the function array shown on fig. 5.

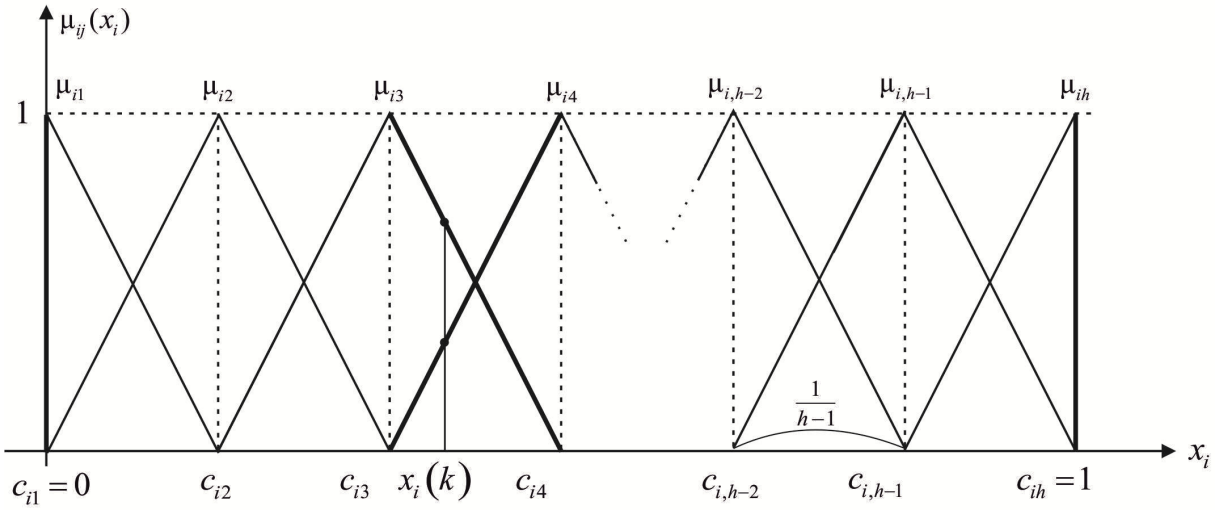


Figure 5. Triangular membership functions

Membership functions μ_{ij} of nonlinear synapses provide Ruspini (unity) partitioning which means that

$$\sum_{j=1}^{m} \mu_{ij}\left(x_i\right) = 1. \tag{8}$$

We will consider μ_{ij} to be a triangular membership function which is defined as

$$\mu_{ij}(x_i) = \begin{cases} \dfrac{c_{ij} - x_i}{c_{ij} - c_{i,j-1}}, & x_i \in (c_{i,j-1}; c_{ij}], \\ \dfrac{x_i - c_{ij}}{c_{i,j+1} - c_{ij}}, & x_i \in (c_{ij}; c_{i,j+1}], \\ 0, & otherwise \end{cases} \tag{9}$$

for $i = 1\ldots 3$, $j = 1, 2, \ldots, h$.

In our case membership functions are evenly distributed in the range $[0,1]$.

It should be noticed that triangular membership functions provide piecewise-linear approximation which can lead to the deterioration of results accuracy. To minimize this effect one may increase the amount of membership functions but this will increase the amount of synaptic weights and make a system architecture much more complicated as well as its learning algorithm. Cubic splines should be used as membership functions to get rid of the above-mentioned situations which can be written in the form:

$$\mu_{ij}\left(x_i\right) = \begin{cases} 0.25\left(2 + 3\dfrac{2x_i - c_{ij} - c_{i,j-1}}{c_{ij} - c_{i,j-1}} - \left(\dfrac{2x_i - c_{ij} - c_{i,j-1}}{c_{ij} - c_{i,j-1}}\right)^3\right), & x \in \left[c_{i,j-1}, c_{ij}\right], \\ 0.25\left(2 - 3\dfrac{2x_i - c_{i,j+1} - c_{ij}}{c_{i,j+1} - c_{ij}} + \left(\dfrac{2x_i - c_{i,j+1} - c_{ij}}{c_{i,j+1} - c_{ij}}\right)^3\right), & x \in \left(c_{ij}, c_{i,j+1}\right], \\ 0, \quad otherwise. \end{cases} \tag{10}$$

Cubic splines satisfy Ruspini partition too and improve approximation characteristics of the fuzzy inference process. On the other hand, the usage of cubic splines provides smooth polynomial approximation and allows to model nonstationary signals with high accuracy results (fig. 6).

Taking into consideration a $\left(\left(n+1\right)nhg\right)\times 1$-vector of membership functions values

$$\mu\left(k\right) = \left(\left(\mu_{11}^{[11]}\left(x_1\left(k-1\right)\right);\ldots;\mu_{21}^{[11]}\left(\hat{x}_1^1\left(k-2\right)\right);\ldots;\mu_{n+1,h}^{[11]}\left(\hat{x}_n^1\left(k-2\right)\right);\right.$$
$$\left.\left(\mu_{11}^{[gn]}\left(x_g\left(k-1\right)\right);\ldots;\mu_{21}^{[gn]}\left(\hat{x}_1^g\left(k-2\right)\right);\ldots;\mu_{n+1,h}^{[gn]}\left(\hat{x}_n^g\left(k-2\right)\right)\right)^T$$

and a synaptic weights vector of the same dimensionalities

$$w\left(k\right) = \left(\left(w_{11}^{[11]}\left(x_1\left(k-1\right)\right);\ldots;w_{21}^{[11]}\left(\hat{x}_1^1\left(k-2\right)\right);\ldots;w_{n+1,h}^{[11]}\left(\hat{x}_n^1\left(k-2\right)\right);\right.$$
$$\left.\left(w_{11}^{[gn]}\left(x_g\left(k-1\right)\right);\ldots;w_{21}^{[gn]}\left(\hat{x}_1^g\left(k-2\right)\right);\ldots;w_{n+1,h}^{[gn]}\left(\hat{x}_n^g\left(k-2\right)\right)\right)^T$$

the network output can be presented in a vector form

$$\hat{x} = w^T \mu(k). \tag{11}$$

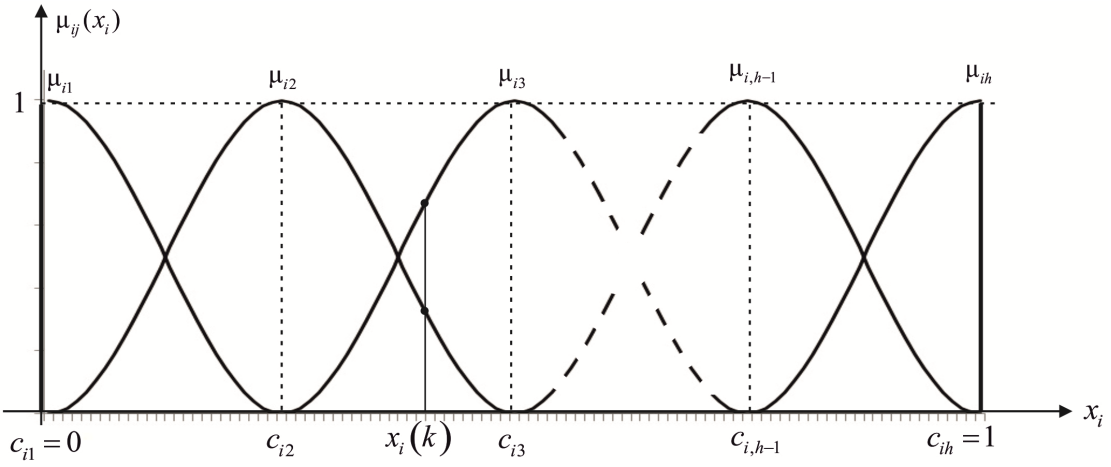


Figure 6. Cubic spline membership functions.

Learning Procedure

To find optimal values of synaptic weights the conventional learning criterion

$$E = \frac{1}{2}\sum_{k=1}^{N} \left\| y(k) - \hat{y}(k) \right\|^2 \alpha^{N-k} \tag{12}$$

and an exponentially weighted recurrent least squares optimization technique are used:

$$\begin{cases} w(k+1) = w(k) + \dfrac{P(k)(x(k+1) - w^T(k)\mu(k+1))}{1 + \mu^T(k+1)P(k)\mu(k+1)}\mu(k+1), \\ P(k+1) = \dfrac{1}{\alpha}\left(P(k) - \dfrac{P(k)\mu(k+1)\mu^T(k+1)P(k)}{\alpha + \mu^T(k+1)P(k)\mu(k+1)} \right), \ 0 < \alpha < 1. \end{cases} \tag{13}$$

Conclusion

The proposed architecture forms a "reservoir" which is built using neo-fuzzy neurons. This neuro-fuzzy network is designed to deal with time series of considerably nonstationary characteristics. The network has such advantages as numerical simplicity and high processing speed while comparing it to traditional forecasting neural networks and neuro-fuzzy systems.

Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Bibliography

[Alexandre, 2009] L.A. Alexandre, M.J. Embrechts. Reservoir size, spectral radius and connectivity in static classification problems. Ed. C. Alippi: ICANN 2009, Part I, LNCS 5768, Springer-Verlag, Berlin Heidelberg, 2009, P. 1015-1024.

[Bodyanskiy, 2009] Ye. Bodyanskiy, O. Tyshchenko Neo-fuzzy forecasting echo state network. Advanced Computer Systems and Networks: Design and Application: Proc. 4th Int. Conf. ACSN-2009, Lviv: NVF «Ukrainski Tehnologii», 2009, P. 95-96.

[Bodyanskiy, 2010] Ye. Bodyanskiy, O. Tyshchenko. The reservoir predictive neuro-fuzzy network. Proc. Int. Conf. on Intellectual Systems for Decision Making and Problems of Computational Intelligence, Kherson:KhNTU, Vol.1, 2010, P. 279-282 (in Russian).

[Horio, 2001] K. Horio, T. Yamakawa. Modified counterpropagation employing neo fuzzy neurons and its application to system modelling. Proc. Int. Conf. on Info-tech and Info-net (ICII 2001), IEEE Press, 2001, Vol.4, P. 50–55.

[Jaeger, 2001a] H. Jaeger. The echo-state approach to analysing and training recurrent neural networks. Technical report, German National Research Centre for Information Technology, 2001, 45p.

[Jaeger, 2001b] H. Jaeger. Short term memory in echo state networks. Technical report, German National Research Centre for Information Technology, 2001, 36p.

[Miki, 1999] T. Miki, T. Yamakawa. Analog implementation of neo-fuzzy neuron and its on-board learning. Computational Intelligence and Applications, ed. N. E. Mastorakis, Piraeus: WSES Press, 1999, P. 144-149.

[Tyshchenko, 2012] O. Tyshchenko, I. Pliss. The forecasting neuro-neo-fuzzy network based on reservoir computing. Control Systems, Navigation and Connection Systems, No. 1(21), Vol. 2, Kyiv, 2012, P. 123-126 (in Russian).

[Uchino, 1997] E. Uchino, T. Yamakawa. Soft computing based signal prediction, restoration and filtering. Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, ed. Da Ruan, Boston: Kluwer Academic Publisher, 1997, P. 331-349.

[Yamakawa, 1992] T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. Proc. 2-nd Int.Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92", Iizuka, Japan, 1992, P. 477-483.

Authors' Information

Yevgeniy Bodyanskiy *– Doctor of Technical Sciences, Professor of Artificial Intelligence Department and Scientific Head of the Control Systems Research Laboratory; Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, 61166, Ukraine. Tel: +380577021890,e-mail:* bodya@kture.kharkov.ua.

Major Fields of Scientific Research: Hybrid Systems of Computational Intelligence.

Iryna Pliss - *Ph.D., Leading researcher, Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenin Av., 14, Kharkiv, 61166, Ukraine, e-mail:* pliss@kture.kharkov.ua.

Major Fields of Scientific Research: Hybrid Neuro-Fuzzy Systems.

Oleksii Tyshchenko *– Junior Researcher, Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine, 61166. Tel: +380577021890, e-mail:lehatish@gmail.com.*

Major Fields of Scientific Research: Hybrid Neuro-Fuzzy Systems.

PRION CRYSTALIZATION MODEL AND ITS APPLICATION TO RECOGNITION PATTERN

Paula Cordero, Rafael Lahoz-Beltra and Juan Castellanos

***Abstract**: This paper introduces APA ("Artificial Prion Assembly"): a pattern recognition system based on artificial prion crystalization. Specifically, the system exhibits the capability to classify patterns according to the resulting prion self- assembly simulated with cellular automata. Our approach is inspired in the biological process of proteins aggregation, known as prions, which are assembled as amyloid fibers related with neurodegenerative disorders.*

***Keywords**: pattern recognition, prion protein aggregation, crystal growth, simulation models.*

***ACM Classification Keywords**: F.1. Computation by abstract devices, F1.1. Models of Computation.*

Introduction

Molecular self-assembly is one of the most relevant biological mechanisms related with the self-organization and biological functions exhibited within cells [Lahoz-Beltra, 1997]. Cellular automata are able to capture the main features of biomolecules that form part of cellular structures and organelles. A breakthrough in the computer modeling and simulation of proteins took place when cellular automata modeling was applied to simulate the interaction of proteins during self-assembly [Lahoz-Beltra, 1999]. In a different realm spin glasses and the Hopfield content addressable memory exhibit emergent collective computational abilities. Such capabilities are related with phase transitions between the crystalline state (low temperature and energy) and liquid state (high temperature and entropy). In consequence, crystalization process may be a manifestation of the ability of Nature to process information, e.g storage and pattern recognition.

Cellular automata belong to a family of discrete, connectionist techniques being used to investigate fundamental principles of dynamics, evolution, and self-organization. In general, they constitute exactly computable models for complex phenomena and large-scale correlations that result from very simple short-range interactions. In this paper, a cellular automaton is designed to model the proteins behavior during self-assembly. The cellular approach makes it possible to achieve a theory-based view of morphogenesis detail to link the results directly to a classification model. In this paper we explore how prion protein aggregation or self-assembly could be used to design an artificial pattern recognition system. The proposed computational system has been dubbed as APA ("Artificial Prion Assembly") memory.

Prion protein aggregation and its simulation models

A prion is an infectious agent composed of protein in a misfolded form. Prion diseases are fatal neurodegenerative disorders associated with the polymerization of the cellular form of prion protein (PrPC) into an amyloidogenic β-sheet infectious form (PrPSc) [Fontaine & Brown, 2009].

Prions propagate by transmitting the misfolded protein state. When a prion enters a healthy organism, it induces existing properly folded proteins to convert into the disease-associated prion form; the prion acts as a template to guide the misfolding of more proteins into prion form. These newly formed prions can then go on to convert more proteins themselves; this triggers a chain reaction that produces large amounts of the prion form.

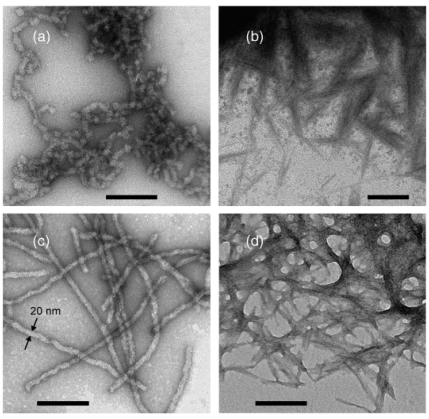


Figure 1. TEM images of prion-seeded Ure2p fibrils (a and b) and unseeded Ure2p fibrils (c and d) before PK treatment (a and c) and after PK treatment (b and d). TEM grids are negatively stained with uranyl acetate. Scale bars represent 200 nm. [Kryndushkin et al, 2011]

Under normal conditions, the high-energy barrier separates PrPC from PrPSc isoform. However, pathogenic mutations, modifications as well as some cofactors, such as glycosaminoglycans, nucleic acids, and lipids, could modulate the conformational conversion process. Abundant nonfibrillar oligomeric intermediates are a common feature of amyloid formation [Bemporad & Chiti, 2013], and these oligomers, rather than the final fibers, have been suggested to be the toxic species in some amyloid diseases.

Evidence suggests that an aggregated form of PrPSc is in fact the key component in the disease [Stahl et al., 1987; Brown et al., 1997] but the precise character of the infectious aggregates is unclear. Therefore, the study of in vitro aggregation of recombinant PrP is instrumental in providing insight into the mechanisms behind conversion from PrPC to PrPSc and aggregate accumulation, as well as to determine the conformation and species that is actually responsible for prion pathogenesis.

Many efforts are dedicated to design aggregation models developed. Aggregation of PrP has been modelled using three kinetic theories: template assisted-aggregation, nucleation-elongation polymerization, and branched-chain polymerization [Fontaine & Brown, 2009]. Each of these theories has been reviewed elsewhere, but in brief summary all employ the idea that a smaller unit of PrP is responsible for further catalysing protein aggregation.

Mainly, models of aggregation processes in prion disease include one-dimensional, fibrillar aggregation-and-fission models, since aggregates grown in vitro are typically seen to be fibrillar. There are several approaches used to model this kind of processes. These models range from stochastically and deterministic approaches through known kinetic models based on differential equations [Greer et al, 2006] to cellular automata based models [Kulkarni et al, 2003].

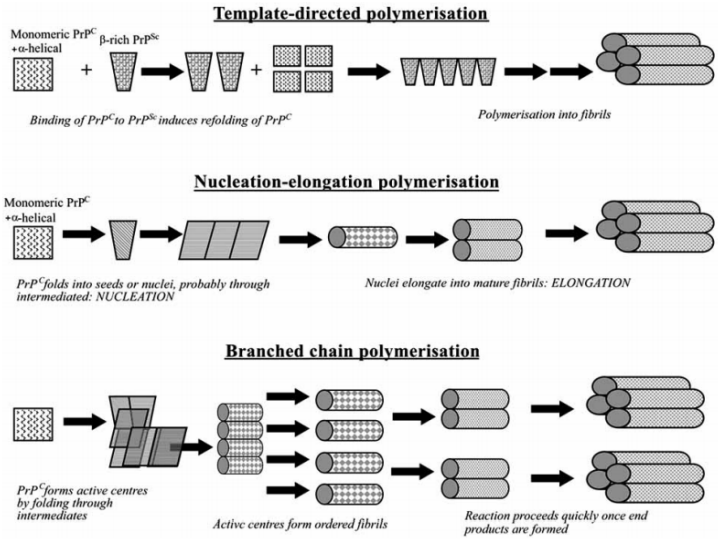


Figure 2. Prion protein aggregation kinetic models. Schematic illustrating the differences between the kinetic theories of PrP aggregation. [Fontaine & Brown, 2009]

Diffusion-Limited Aggregation and Crystal Growth models

Diffusion-limited aggregation (DLA) [Witten & Sander, 1981] is an idealization of the process by which matter irreversibly combines to form dust, soot, dendrites, and other random objects in the case where the rate-limiting

step is diffusion of matter to the aggregate. Diffusion is the movement of particles due to temperature fluctuations and seen in Brownian motion. By the other hand, an aggregate is a collection of particles that are connected together this growing process is called diffusion-limited when the aggregate increases in size by one particle at a time. A particle is appear from a random position far away and is allowed to diffuse. If it touches the seed, it is immobilized instantly and becomes part of the aggregate. This happens since the density of particles is low and thus the particles do not come into contact with each other before reaching the aggregate.

The growth processes [Levi & Kotrla, 1997] are described by nonlinear partial-differential equations and both the analytical and the numerical treatments of these equations are extremely difficult even on current computers. As a result, many of the questions concerning structure formation and transitions between different growth morphologies have not so far been satisfactorily answered. Much effort has especially been devoted to establishing the relationship between cluster morphology and the growth mechanism.

The crystal growth is a phase transition process with sharp border between it and initial feeding phase like a liquid, gas or plasma. The structure element, molecule, of the crystal could be determined as a minimum part of it when a reaction of incorporating itself in the crystal will effect with changing energy of the whole system that will be equivalent to the condensation energy of the corresponding mass of crystal. The molecule is the minimum part of the crystal that behaves as a whole crystal. The principal difference of behavior of molecules in liquids is based on the principle of the long order in the crystals structure. Each molecule has exact position relatively to the other in crystal.

In the every moment of time one molecule on the surface of a crystal have two options: to get out to initial matter or stay incorporated as a part of crystal. The same choice is true for the molecule outside of crystal in direct closeness to its surface. It can be incorporated into body of crystal or stay outside. Most of simulation models implement these phenomena by calculating these probabilities for each position on the surface of the crystal and comparing with the random number to decide what one of the possible events will happen.

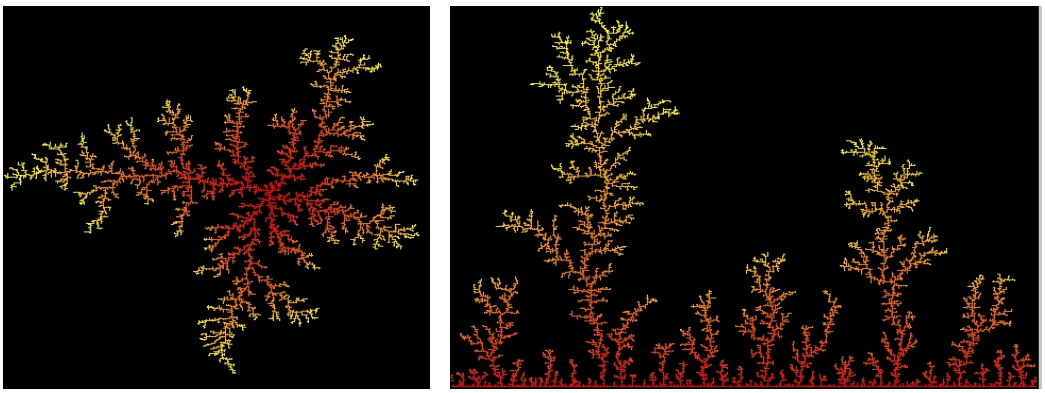


Figure 3. Sample DLA images from java applet: http://www.joakimlinde.se/java/DLA/

Spin glass models and addressable memories

At this point it should be noted that many studies find that some types of addressable memory as Hopfield's neural network [Hopfield, 1982] was inspired by analogies with the physics of magnetism the same as crystallization kinetics. Main characteristic of the Hopfield's model is the recurrence of the network with total connectivity and a symmetric weight matrix; binary valued outputs, which provide a simple prescription for the weights, with no training needed; output settles down to a steady state.

As it was mentioned, there are some relations between the behavior of addressable memories and the physics of magnetic spin systems [Edwards & Anderson, 1975]. In particular, phase transitions represent a competition between minimizing the energy (usually producing an ordered state) and maximizing the entropy (increasing the

disorder). This can be understood in terms of the behavior to the free energy, F=E-TS, which tends to a minimum for a system in thermal equilibrium. At low values of the temperature T, is more important to minimize the energy E. When T is high, then large values of the entropy S will make F smaller. The most interesting aspect is the abrupt, discontinuous nature of the transition between the ordered state and the disordered state occasionally attributed to emergent collective behavior. This behavior depends on the values of the positions and momentum of each of the molecules.

The key to being able to quantitatively describe the free energy and related thermodynamic quantities is to express the total energy of the system in terms of the states of the atoms. In this sense, it is described the Hopfield's model behavior where, in the presence of more than one pattern, the weights aren't optimum for the retrieval of any one pattern, but represent an average or compromise over the set of patterns. Instead of having a single minimum energy state, we will have a local minimum for each pattern. If the initial state of the network isn't too far from the stored state, the system will slide into the nearest local minimum, which will be the desired output state. When it try to minimize the error in a feed-forward net by using the back-propagation algorithm, getting stuck in a local minimum can keep us from finding the global minimum that produces the best set of weights. In this case, local minima are desirable.

Hopfield found in his computer experiments that the ability of the network to retrieve patterns fell dramatically when the number of stored patterns approached 15% of the number of neurons. As the number of patterns becomes large, the weight begins to look like a random variable. The term in the sum that favors a particular pattern is greatly outweighed by all the others, and the associative memory begins to look like a true spin glass.

Analogously to the above approach, the work presented in this paper is based on parallelism between the phenomena of crystallization (in the case of prions) and addressable memories behavior that implement pattern recognition capabilities as it is described below.

Pattern recognition system based on Prion Crystalization

This work presents a pattern recognition system based on a model of prion crystallization. This proposed model represents an unsupervised learning system. The theoretical model is composed of two phases: Phase-I) in which is modelled the winged-helix dimerization process through a probabilistic cellular atuomata approach in order to obtain the nucleation seeds of the prion-crystallization process. Phase-II) in which crystallization process will model prion dynamics from the nucleation centers obtained. The system will be able to classify input patterns by decoding implicit information from the morphology of the crystal-prions, which will be obtained as a result of the application of the two phases of the model.

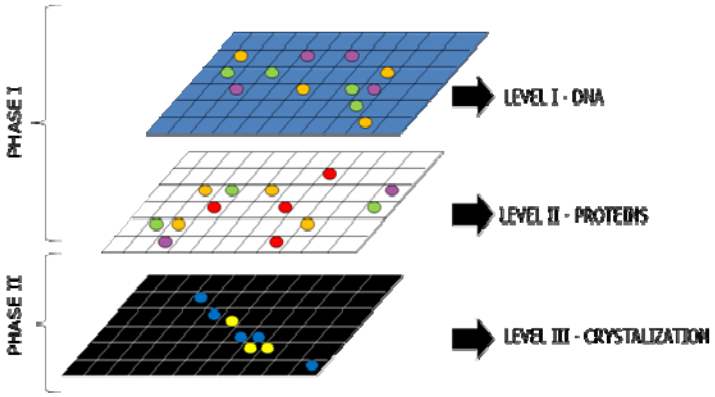


Figure 4. Schema of the Probabilistic Cellular Automata proposed.

As mentioned, the proposed system is inspired by the biochemical process by which certain strands of DNA in vitro affect the process of prion formation. Specifically, this process is described in the work "A DNA promoted amyloid proteinopathy in Escherichia coli" [Fernandez et al, 2011]. This research team goes into detail about this phenomena, core of proposed model in its Phase-I, and states that similarly to the mammalian proteins PrP and a-

synuclein, the winged-helix dimerization (WH1) domain of the bacterial plasmid-encoded RepA protein can assemble into amyloid fibers upon binding to DNA in vitro.

In the proposed model, this phenomenon is simulated by a two-level automaton. At first level (Level-I) each occupied cell represents one of the three different DNA sequences (ADN_1, ADN_2, ADN_3) that can interact with the protein subunits (PrP). At second level (Level-II) proteins are represented. These cells can transit, with some probability defined at probability vector (P_v), from six different states according to the transition rules defined in each case. The formalization of the proposed cellular automata is described below. Our cellular automata is defined by M= (G, G_0, N, Q, δ, T), where:

- G: matrix automata (Q-dimensional);
- G_0: set of initial values of the automata (G) states;
- N: function that assigns each automaton the set of neighbors (Neighborhood function);
- Q: sets of possible states;
- δ: transition function that assigns a new state to an automaton having into account the state of all its neighbors;
- T: set of final states.

The automata consists of a set of six possible states (Q= {S_0, S_1, S_2, S_3, S_4, S_5}), it implements the well-known function of Von Neumann neighborhood [Kennedy & Mendes, 2003] and has a unique final state, S_4. Each of the possible states represents a different physicochemical prion formation state during the process, thus:

- S_0= represents the absence of protein in the cell;
- S_1= represents a stable state of the protein. This state is not affected by the interaction with the DNA grid (Level-I);
- S_2= molecular state of instability. Cells in this state are potentially exposed to interaction with Level-I. The neighborhood function is applied and executed transition rules;
- S_3= represents a functional protein;
- S_4= represents a prion infectious form. Seed of the nucleation center;
- S_5= represents a nonfunctional protein;

The transition rules, TR = {$p_{0,1}$, $p_{1,2}$, $p_{2,3}$, $p_{2,4}$, $p_{2,5}$, $p_{3,0}$, $p_{5,0}$}, are applied in each case with a certain probability. In particular, the rules $p_{0,1}$ and the set {$p_{3,0}$,$p_{5,0}$} represent the transition probability $S_0 \rightarrow S_1$, $S_3 \rightarrow S_0$ y $S_5 \rightarrow S_0$, these probabilities represent the rate of molecular generation and degradation rates respectively. In the case of the rule $p_{1,2}$ represents the transition probability $S_1 \rightarrow S_2$ in which the cell moves from one stable state to a state of molecular instability. Finally, the rule set {$p_{2,3}$,$p_{2,4}$, $p_{2,5}$} implements the transition functions related to the interaction with DNA sequences (Level-I) by applying the selected neighborhood function ($S_2 \rightarrow S_3$ binding to ADN_1, $S_2 \rightarrow S_4$ binding to ADN_2, $S_2 \rightarrow S_5$ binding to ADN_3). This behavior is shown in FIGURE 5.

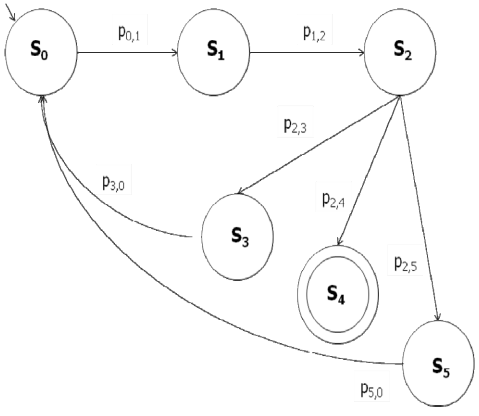


Figure 5. Pion-crystalization Automaton. State-transition detail.

In this paper are shown two simulations of the proposed model. The first one corresponds to the simulation without the presence of input pattern. By this simulation it can appreciate the molecular concentrations of different states according to the underlying model at Phase-I. The second simulation presents the results obtained during the training phase of the system. Note that in this work the results related to Phase-II of the model (algorithm crystallization) are not shown. The results obtained by applying the model of crystallization and the appropriate decoding for final classification of the patterns will be shown in the near future.

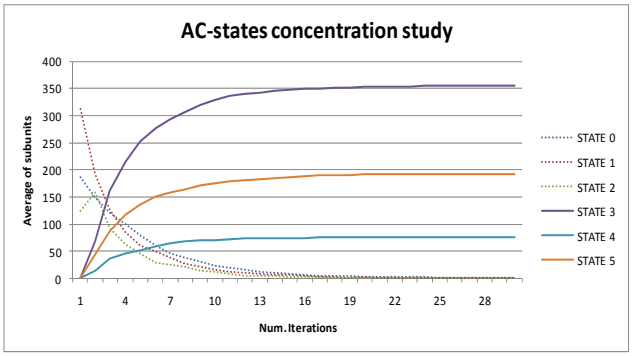


Figure 6. Average of states concentration chart.

At first simulation, the initial population of items, molecules, is randomly distributed on a two-dimensional square lattice with periodic boundary conditions, $N_X \times N_Y = 50 \times 50$. The initialization of the Level-I grid corresponds to three-type DNA initial concentration (C_{DNAn}). These values have been calculated according to the following proportions: $C_{DNA3} > (C_{DNA1} + C_{DNA2})$ and $C_{DNA1} > C_{DNA2}$. In FIGURE 6 results over 10 simulations with same settings can be observed. These average concentrations were obtained after 30 system iterations and with rates of generation and degradation of 0,15 and 0,45 respectively, for each simulation. As you it can see the concentration of state 3 (functional proteins triggered by DNA_1) is higher despite the initial conditions of concentration of DNA sequences, where the proportion of type sequences DNA_3 is majority. This behavior corresponds proportionally to results observed in real (in-vitro) prion formation processes.

Finally, it is shown the results obtained in the training phase of the proposed system, FIGURE 7. This training experiment aims to get the system to recognize alphanumeric characters. Specifically, in the case above, is introduced an input pattern which codifies the character 'A' (FIGURE 7-a), the input pattern is encoded in the Level-I of the model. The DNA sequences of this level are initialized with concentrations according to the codification of the input pattern. As a result of applying the underlying algorithm it is obtained the corresponding Level-II, state level, configuration (FIGURE 7-b) and also the resulted configuration of Level-III, nucleation seed level, (FIGURE 7-c); starting point of crystallization process. As mentioned above, this process will result in encoded input patterns implicitly in the resulting morphology of prion forms. By applying the decoding algorithm designed for the system will be able to classify unknown input patterns.

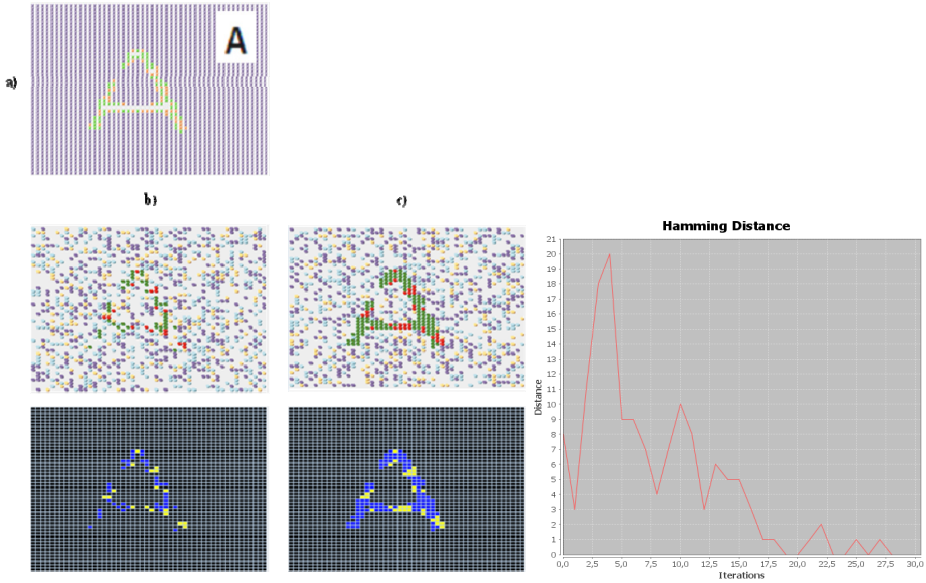


Figure 7. Pion-crystalization Automaton. State transition detail. a)encoded pattern in Level-I b) Level-II and Level-III at 30% simulation c) Level-II and Level-III at hundred 100% simulation d) Hamming distance beyween input pattern and resulted outputs.

Conclusion

This paper introduces APA: a novelty artificial pattern recognition system based on prion crystalization. At present we conducted the modeling and simulation experiments showing the plausibility of a memory based on prion self-assembly. Thus, we studied the main steps and features of the training step. The system is able to memorize patterns into the resulting prion self- assembly. Such memory is implemented as a hierarchical (DNA, proteins, crystal assembly) bioinspired 2D cellular automata. Our approach opens the possibility of designing pattern recognition systems inspired by the phenomenon of crystallization in biology.

Bibliography

[Bemporad & Chiti, 2013] Bemporad, F., & Chiti, F. Pathways of Amyloid Formation. In: Amyloid Fibrils and Prefibrillar Aggregates: Molecular and Biological Properties, 2013.

[Brown et al, 1997] Brown, D. R., Schulz-Schaeffer, W. J., Schmidt, B., & Kretzschmar, H. A. Prion protein-deficient cells show altered response to oxidative stress due to decreased SOD-1 activity. In: Experimental neurology, 1997.

[Edwards & Anderson, 1975] Edwards, S. F., & Anderson, P. W. Theory of spin glasses. In: Journal of Physics F: Metal Physics, 1975.

[Fernández et al, 2010] Fernández Tresguerres, M. E., Moreno Díaz de la Espina, S., Gasset Rosa, F., & Giraldo, R. A DNA-promoted amyloid proteinopathy in Escherichia coli. In: Molecular microbiology, 2010.

[Fontaine & Brown, 2009] Fontaine, S. N., & Brown, D. R. Mechanisms of prion protein aggregation. In: Protein and peptide letters, 2009.

[Greer et al, 2006] Greer, M. L., Pujo-Menjouet, L., & Webb, G. F. A mathematical analysis of the dynamics of prion proliferation. In: Journal of theoretical biology,2006.

[Hopfield, 1982] Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the national academy of sciences, 1982.

[Kennedy & Mendes, 2003] Kennedy, J., & Mendes, R. Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms. In: Soft Computing in Industrial Applications, 2003.

[Kryndushkin et al, 2011] Kryndushkin, D. S., Wickner, R. B., & Tycko, R. The core of Ure2p prion fibrils is formed by the N-terminal segment in a parallel cross-β structure: evidence from solid-state NMR. In: Journal of molecular biology, 2011.

[Kulkarni et al, 2003] Kulkarni, R. V., Slepoy, A., Singh, R. R. P., Cox, D. L., & Pázmándi, F. Theoretical modeling of prion disease incubation. In: Biophysical journal, 2003.

[Lahoz-Beltra, 1997] Lahoz-Beltra, R. Molecular automata assembly: principles and simulation of bacterial membrane construction. In: Biosystems. 1997.

[Lahoz-Beltra, 1999] Lahoz-Beltra R. Molecular automata modeling in structural biology. In: Advances in Structural Biology, 1999.

[Levi & Kotrla, 1997] Levi, A. C., & Kotrla, M. (1997). Theory and simulation of crystal growth. In: Journal of Physics: Condensed Matter, 1997.

[Slepoy et al, 2001] Slepoy, A., Singh, R. R. P., Pazmandi, F., Kulkarni, R. V., & Cox, D. L. Statistical mechanics of prion diseases. In: Physical review letters, 2001.

[Stahl et al, 1987] Stahl, N., Borchelt, D. R., Hsiao, K., & Prusiner, S. B. In: Scrapie prion protein contains a phosphatidylinositol glycolipid. In: Cell, 1987.

[Witten & Sander, 1981] Witten Jr, T. A., & Sander, L. M. (1981). Diffusion-limited aggregation, a kinetic critical phenomenon. In: Physical Review Letters, 1981.

Authors' Information

Paula Cordero – Natural Computing Group, Facultad de Informática,Universidad Politécnica de Madrid, Campus de Montegancedo s.n., 28660 Boadilla del Monte, Madrid, Spain; e-mail: paula.cormo@gmail.com

Major Fields of Scientific Research: complex systems, natural computing, evolutionary computation and the design of bioinspired algorithms.

Rafael Lahoz-Beltra – Department of Applied Mathematics, Faculty of Biological Sciences, Complutense University of Madrid, 28040 Madrid, Spain ; e-mail: lahozraf@ucm.es

Major Fields of Scientific Research: evolutionary computation, embryo development modeling and the design of bioinspired algorithms

Juan Castellanos – Head of Natural Computing Group, Facultad de Informática,Universidad Politécnica de Madrid, Campus de Montegancedo s.n., 28660 Boadilla del Monte, Madrid, Spain; e-mail: jcastellanos@fi.upm.es

Major Fields of Scientific Research: natural computing, formal language and automata theory.

AN ARCHITECTURE FOR REPRESENTING BIOLOGICAL PROCESSES BASED ON NETWORKS OF BIO-INSPIRED PROCESSORS

Sandra Gómez Canaval, Fernando Arroyo and José Ramón Sánchez-Couso

***Abstract**: In this paper we propose the use of Networks of Bio-inspired Processors (NBP) to model some biological phenomena within a computational framework. In particular, we propose the use of an extension of NBP named Network Evolutionary Processors Transducers to simulate chemical transformations of substances. Within a biological process, chemical transformations of substances are basic operations in the change of the state of the cell. Previously, it has been proved that NBP are computationally complete, that is, they are able to solve NP complete problems in linear time, using massively parallel computations. In addition, we propose a multilayer architecture that will allow us to design models of biological processes related to cellular communication as well as their implications in the metabolic pathways. Subsequently, these models can be applied not only to biological-cellular instances but, possibly, also to configure instances of interactive processes in many other fields like population interactions, ecological trophic networks, industrial ecosystems, etc.*

***Keywords**: Networks of biologically-inspired processors, Bioinspired Architectures, Computational Models, Natural Computing.*

***ACM Classification Keywords**: F.1.1 Models of Computation - Unbounded-action devices.*

Introduction

Several new lines of research have been initiated in the last decade within Natural Computing from two working perspectives: on the one hand, the bio-inspired architectures and computational models, and on the other, the computational techniques and user friendly tools, which support the advancements in synthetic and systems biology. A Network of Bio-inspired Processors (NBP) [Castellanos, 2003] [Manea, 2005] [Campos, 2012] is a computational model inspired by cell biology; thus it is based on a rather common architecture for parallel and distributed symbolic processing and is related to the Connection Machine [Hillis, 1979] and the Logic Flow Paradigm [Errico, 1994]. NBP model consists of several processors, each one of which is placed in a node of a virtual graph. Every processor acts on local data in accordance with some predefined rules. After that, the processor both sends and receives the data (which behaves like a mobile agent that can navigate in the network following a given protocol). Then, the processors can communicate the data resulting with the rest of the processors which it is connected in the graph, using a filtering strategy. This strategy may require satisfying some conditions that are imposed by processors, when sending, receiving or both. A processor node is very simple: can be either an evolutionary, splicing or genetic processor depending on the operations that carries out. In addition, an extension of NBP named Network Evolutionary Processors Transducers (NEPT for short) was introduced in [Gómez, 2012] to simulate the work of generalized sequential machines.

Changes of the state of a cell can be modeled by means of rewriting rules like in formal grammars [Pâun, 2000]. Also, the parallel nature of these changes implies a parallel application of the involved rules. In this sense, is

more than proved that NBP is a suitable bio-inspired computing model to represent massively parallel changes of the state of a cell [Castellanos, 2003].

Cellular organization and their biological processes include evolutionary aspects which can be viewed as a complex web of subsystems that transform the matter, such as cellular metabolism, signaling, sexual reproduction, etc. One can consider such a web as a composition of a number of different networks working together. Moreover, each one of these networks will represent the dynamics of a biological process, like the metabolism through metabolic pathways or cellular signaling through intracellular pathways. Therefore, we claim that a web of NBPs and NEPTs is able to represent this dynamic. In addition, we propose a multilayer architecture to design and to build this web of NBPs and NEPTs working in a cooperative way.

Related Work

The main framework analysis for the most part of biological dynamics remains in the theory of ordinary differential equations (ODE). There are a lot of works approximating ODE models to particular biological phenomena and some of them focus in a more general perspective, for instance the works introduced in [Schaff, 1997] and [Novak, 2006]. However, the use of this type of models presents some intrinsic limitations in the evaluation of the kinetic reaction rates, which usually refer to a microscopic level, hardly accessible to reliable measurements. Really, in living organisms, these measurements dramatically alter the context of the investigated processes [Manca, 2009]. On the other hand, in a discrete mathematical setting, specific applications for modeling biological phenomena and bio-inspired paradigms are widely developed. Bio-inspired paradigms abstracted from the information processing that are present in all living cellular systems, such as Membrane Computing based on P Systems [Pâun, 2000] have demonstrated being universal and computationally complete when they work in a massively parallel way. In particular, P Systems have exhibited models representing biological phenomena. However, these models are mainly of a qualitative nature, and do not provide criteria for predicting quantitative aspects of biological processes. For overcoming these limitations, Metabolic P (MP) systems were introduced in [Manca, 2009] and then widely developed along different approaches. Nevertheless, the extension of these models to other types of biological processes is still an open working line, despite of the efforts given in [Gutiérrez, 2008] and [Nguyen, 2009], mainly due to the hierarchical structure of P Systems and the limitations to develop suitable and practical architectures (both software and hardware) to implement them.

NBP were widely studied from its definition [Castellanos, 2001], as well as a series of subsequent works about their computational power and complexity. NBP can be viewed as a device that solves NP-complete problems in linear time with polynomially bounded resources. In addition, a new variant of NBP with the same properties (but without filtering process) and an energetic charge representation was studied in [Alarcón, 2012]. Also a software environment and its architecture were presented in [Ortega, 2012]; this framework continues to be an active line of research as well as the suitability of some hardware architectures. Recently, in [Gómez, 2012], it has been proved that a transducer based on the model denominated NEP transducer (NEPT) can simulate the work of generalized sequential machines, that is, every recursively enumerable language can be the transduction defined by the new transducer of a very simple regular language, computing the set of all words obtained by the shortest computations.

So far, NPB had only been used as solvers of classical mathematical problems. Alike that the other discrete computational models, NBP can be investigated as solvers of other real problems, not only those related to biological interest, but also the possible relevance ecologies, dynamics of social interactions, etc., that are more complex than combinatorial optimization, as well as other classical NP-complete ones. The study of the web structure of a NBP together the definition of its communication process, as well as some aspects concerning computational power, efficiency and descriptive complexity leads to interesting results that apply in new fields.

Contributions

In this work we present two main contributions. The first one is the use of NEPT to model some chemical transformations within cell biological phenomena. In particular, this novelty is applied to simulate chemical transformations of substances within a computational framework. We use a NEPT to transform words representing extracellular signaling molecules in new words representing receptor proteins able to start the signaling activities. These new words define a language that is recognizable by a NBP. The second contribution, and related to the first, is the definition of an architecture that utilizes both NBP and NEPT to represent signaling activities associated with cellular metabolism. Signaling activities start when an extracellular signaling molecule arrives at plasma membrane where are transformed for receptor proteins. This transformation gives as result intracellular signaling proteins responsible to activate one or more signaling pathways. Finally, one or more of the intracellular signaling proteins alters the activity of effector proteins and thereby the activation or inhibition of specific processes that modify the behavior of the cell, among themselves the metabolism. NEPT transformations represent the first level in our multilayer architecture. The rest of signaling activities are modeled by specific architecture layers, as is extended in section *Architecture Proposal*.

Structure of the rest of the paper

In section "*Definitions and basic Concepts"*, concepts related with the NPB model and its extension as well as a brief biological background are introduced. Section "*Architecture Proposal"* contains the definition of the architecture proposed and its motivation. Section "*Using our architecture"* illustrates the representation of one instance of specific biological processes, such as the Krebs cycle and the MAS shuttle pathway, using our architecture and, finally, the last section shows the conclusions and future work.

Definitions and basic concepts

NBP model concepts

A Network of Bio-inspired Processors (NBP) consists of several processors, each one of which is placed in a node of a virtual graph. Each processor node acts on local data in accordance with some predefined rules. The data becomes a mobile agent which can navigate in the network following a given protocol. The data travel between nodes by means of a filtering process only. Note that this process may require to satisfy some other conditions that are imposed by the processors when sending or receiving data (even simultaneously), by using a variety of filtering strategies. A processor node is very simple; it can be either an evolutionary, splicing or genetic processor taking into account the operations that carries out:

- An *evolutionary processor* performs very simple operations, namely, point mutations in a DNA sequence (insertion, deletion or substitution of a pair of nucleotides). This type of processor is only specialized for just one of these evolutionary operations. More generally, it can be viewed as a cell that contains genetic information encoded in DNA sequences which may evolve by point mutations. A network containing this type of processors is denominated *Network of the Evolutionary Processors*.
- A *splicing processor* performs the operation of recombination, which is presented in form of splicing. This operation is one of the basic mechanisms by means of which, DNA sequences are recombined under the effect of enzymatic activities. A network containing this type of processors is called it *Network of the Splicing Processors*.
- A *genetic processor* has two operations: (1) *Mutation* between symbols (similar to the substitution operation in the evolutionary processors) and (2) *Pure and massive crossover* (similar to the splicing

operation by taking empty contexts). These processors become part of a *Network of the Genetic Processors*.

In addition, there exists another type of processor called *polarized processor* [Alarcón, 2012], which is a special type of evolutionary processor having a valuation mapping. This mapping represents the polarization of the processor in terms of an electrical charge: positive, negative or neutral, which is useful for modeling some biological properties as the inhibition or activation of the molecules or processes. A NBP having this type of processors is denominated *Network of Polarized Evolutionary Processors* (NPEP). For a more detailed view about all of the above networks, we refer the reader to [Castellanos, 2003] [Manea, 2005] [Campos, 2012].

Since a NBP is intended to be used as an universal problem solver, an important aspect is the part of encoding the instance of the problem and that of decoding the solution. It is natural to ask that these steps to be accomplished by a mechanism based on a particular NBP, the NEP as well. Then, we are going to consider a transducer based on the NEP structure which is formed by a directed graph whose nodes are evolutionary processors without filters. In [Gómez, 2012] is demonstrated that these new type of transducers, denominated NEPT, can simulate the work of generalized sequential machines (gsm), computing the set of all words obtained by the shortest computations. Unlike the case of gsm every recursively enumerable language can be the transduction defined by the new transducer of a very simple regular language. The computation on an input word starts with this word placed in an input node and halts as soon as the output node is nonempty. Therefore, a NEPT translates the input word into a set of words existing in the output node. Accordingly, they add new capacities at the NBP, allowing the extension of the model. As the novelty that we have introduced in the section *Introduction*, we use NPET as an overlay network that translate an input into a language that can be recognized by one or more NBP working as underlying networks.

Biological concepts

Cellular communication is a biological process that enables a cell to send chemical signals across the extra-cellular environment, allowing the molecular interchange between cells. Complex intracellular mechanisms are needed to control what signals are emitted and at what time, in order to enable the signal-receiving of the cell. These signals are interpreted and used to guide the behavior of the cell in question. A communication process begins with an extracellular-signal molecule, which arrives at the plasmid membrane. This molecule is received by a molecular receptor, which in turn, activates one or more intracellular signaling pathways. Within these pathways, there are relay chains of molecules (mainly intracellular signaling proteins) that process the signal and distribute it to the appropriate intracellular targets. Targets are generally effector proteins, which are activated (or inactivated) by the signaling pathway. They alter the cellular behavior through processes affecting the shape, movement, metabolism and gene expression, as it shown in Figure 1-(b). Many intracellular signaling proteins behave like molecular switches. When receiving a signal, they switch from an inactive to an active conformation, until another process switches them off [Alberts, 2008]. In summary, extracellular signal molecule is altered (transduced), amplified, distributed, and modulated *en route* through signaling pathway (see Figure 1-(a)). In this sense, activities in the top of communication process, like transduction and amplification, can be modeled by a network (within a given web) that works as *overlay network*. This concept is relevant because of it defines the first level in our multilayer architecture as we will see in the next section.

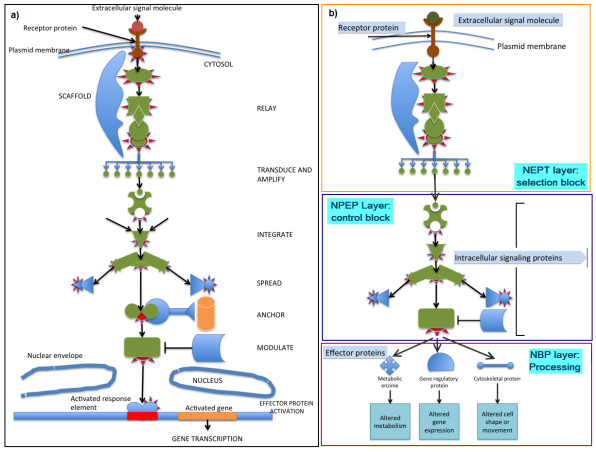


Figure 1: Architecture model of a cell communication process. a) Cell communication process. From receptors to the nucleus through intracellular signaling pathways (adaptation of [Alberts, 2008]). b) Architecture model showing NEPT, NPEP and NBP layers.

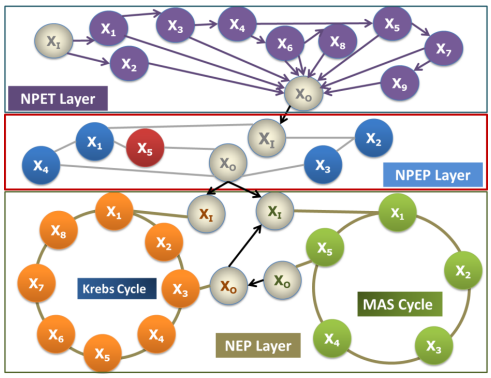


Figure 2: Multilayer architecture to simulate signaling by calcium, MAS and Krebs cycle (X_i represents the nodes of the respective network).

Architecture proposal

As we said before, a NEPT translates input words into a language that can be recognized by one or more NBP. We will see this NEPT as an *overlay network*, that is, a virtual network built on top of one or more existing networks (called the *underlying network*). The overlay network adds an additional layer of indirection/virtualization and changes properties in one or more areas of the underlying network. We propose that the activities in the top of a communication process (of a cell), like transduction or amplification, are modeled by a NEPT working as an *overlay network*.

In this context, our second contribution (see section *Introduction*) is the definition of an architecture for representing the signaling of a cell from reception until the activation of a target processes. We propose three blocks of computing (see Figure 1-(b)):

1. ***Selection***: represents the reception of an extracellular signal molecule arriving at the cellular membrane, and its alteration (transduction), amplification and distribution through of the adequate signaling pathway selected by the cell.
2. ***Control***: realizes the monitoring functionality of signaling pathway either activates or inactivates the target proteins (effectors) in order to unleash the respective cellular processes.
3. ***Processing***: represents the target activity which alters the cellular behavior, like shape, movement, metabolism and gene expression.

Our architecture is inspired in these three blocks, and associates each one of them with its corresponding layer. Each layer is modeled by a specific type of NBP (including NEPT), namely:

- **Top layer (overlay network):** it corresponds with the *selection* block of computing and is implemented by a NEPT.
- **Middle layer (controller network):** it corresponds with *monitoring* block of computing and is implemented by a NPEP.
- **Bottom layer (underlying network)**: it corresponds with *processing* block of computing, and is implemented by one or several NBP.

Using our architecture for modeling metabolic processes

Within the eukaryotic cell, hundreds of different protein kinases are organized into complex networks of signaling pathways that help to coordinate the cellular activities in a cooperative and involved manner. A very well-known metabolic process is the Krebs cycle, also called tricarboxylic acid cycle or simply acid cycle, which is critical in cellular respiration. One of the target activities of the Krebs cycle is the generation of energy. This energy is produced by a chain of reactions, some of them, controlled by calcium (also by the ion Ca^{2+}). The signaling by calcium helps to activate other metabolic processes, such as the malate-aspartate shuttle pathway (MAS for short). In fact, MAS and the Krebs cycle share common enzymes, and major target activity altered by Ca^{2+} is related with brain stimulation. These metabolic activities are of great interest in the brain research [Contreras, 2009]. We use our architecture to design the process of signaling by calcium and the relationship between the Krebs cycle and MAS as follows (see Fig. 2):

- **NEPT layer**: in this overlay layer, we define one NEPT which is able to receive different strings representing extracellular signaling molecules, and translate them into others representing receptor proteins (i.e. piruvate, calcium). These proteins can be recognized by some underlying NBP in the bottom layer, allowing to restart the Krebs cycle and MAS respectively.
- **NPEP layer:** in this control layer, we define only one NPEP. This network selects the respective pathway (represented by a NBP in the bottom layer) depending on the string that represents the adequate receptor traduced by NEPT. Also the NPEP is responsible either to activate or inhibit these networks through the model of the effector proteins function.
- **NBP layer:** in this underlying layer, we define two NBP that represent the Krebs cycle and the MAS pathway respectively. Both networks are communicated by means of the their input/output nodes, and they also compete by the enzymes necessaries for their internal processes, as well as the ketoglutarate, the nicotinamide adenine dinucleotide (in reduced NADH and oxided NAD+ forms), malate, etc. Both networks share the intracellular environment, where the before enzymes are located as well as other type of molecules such as proteins, metabolites, etc.

Conclusions

We have proposed as novelty, the use of NEPT to model some biological phenomena to simulate chemical transformations of substances within a computational framework. In addition, up to our knowledge, we have defined the first multilayer architecture that uses both NBP and NEPT to represent biological processes, for instance, signaling activities that involves cellular metabolism.

Currently, we are working in the modeling and simulation of some interrelated biological phenomena. In particular, we use our architecture to model the Krebs interplay between these processes sharing and competing for substances is an important study of the brain stimulation in vivo as was demonstrated in [Contreras,2009]. We consider important modeling these processes using our architecture, in order to develop a computational software framework that allows their computational simulation. Finally, the analysis of specific features coming from the interaction of a web of NBP and NEPT is a new line of research to be explored in many other fields. In particular, by using our extensible and reconfigurable architecture.

Bibliography

[Alarcón, 2012] P. Alarcón, F. Arroyo, V. Mitrana. Networks of Polarized Evolutionary Processors as Problem Solvers. *Adv. in Knowledge-Based and Intelligent Information and Engineering Systems*, pp: 807-815, 2012.

[Alberts, 2008] B. Alberts, A. Johson, J. Lewis, M. Raff, K. Roberts, P. Walter. Molecular Biology of the Cell. *Garland science, Taylor and Francis Group*. Fifth Edition. 2008.

[Campos, 2012] M. Campos, J.M. Sempere. Accepting Networks of Genetic Processors are computationally complete. *Theoretical Computer Science*, Volume 456, pp: 18-29, 2012.

[Castellanos, 2001] J. Castellanos, C. Martín-Vide, V. Mitrana, J.M. Sempere. Solving NP-complete problems with networks of evolutionary processors. *LNCS 2084*, pp: 621-628, 2001.

[Castellanos, 2003] J. Castellanos, C. Martín-Vide, V. Mitrana, J.M. Sempere. Networks of evolutionary processors. *Acta Informática* 39, pp: 517-529, 2003.

[Contreras, 2009] L. Contreras, J. Satrústegui. Calcium Signaling in Brain Mitochondria. Interplay of Malate Aspartate NADH shuttle and Ca uniporter/mitochondrial dehydrogenase pathways. *Journal of Biological Chemistry - Vol. 284*, No. 11. pp: 7091-7099, 2009.

[Errico, 1994] L. Errico, C. Jesshope. Towards a New Architecture for Symbolic Processing. *Artificial Intelligence and Information-Control Systems of Robots '94*, pp: 31-40, 1994.

[Gómez, 2012] S. Gómez Canaval, V. Mitrana, J.R. Sánchez Couso. Transducers Based on Networks of Evolutionary Processors. *Submitted*.

[Gutiérrez, 2008] A. Gutiérrez, L. Fernández, F. Arroyo, S. Alonso. Suitability of using microcontrollers in implementing new P system communication architectures. *In Masanori Sugisaka and Hiroshi Tanaka*, editors, pp: 496-499, 2008.

[Hillis, 1979] W.D. Hillis. The Connection Machine. *MIT Press*, Cambridge, 1979.

[Manca, 2009] V. Manca. Fundamentals of Metabolic P Systems. *Handbook of membrane computing*, Oxford, 2009.

[Manea, 2005] F. Manea, C. Martín-Vide, V. Mitrana. Accepting Networks of Splicing Processors. *New Computational Paradigms,* Volume 3526, pp: 300-309, 2005.

[Nguyen, 2009] V. Nguyen, D. Kearney, G. Gioiosa. An algorithm for non-deterministic object distribution in P systems and its implementation in hardware. *Membrane Computing, Vol. 5391, LNCS*, pp: 325-354, 2009.

[Novak, 2006] A. Csikasz, D. Battogtokh, K. Chen, B. Novak, J. Tysona. Analysis of a generic model of eukaryotic cell-cycle regulation. *Biophys Journal 90 (12),* pp: 4361-4379, 2006.

[Ortega, 2012] A. Ortega, M. Cruz, E. Del Rosal, C. Navarrete, A. Jiménez, J. De Lara, E. Anguiano, M. Cuéllar, J.M. Rojas. Developing Tools for Networks of Processors. *Triangle Journal, URV Publications*, pp: 38-72, 2012.

[Pâun, 2000] G. Pâun. Computing with membranes. *Journal Comput.System Sci.61*, pp: 108-143, 2000.

[Schaff, 1997] J. Schaff, C. Fink, M. Slepchenko, J. Carson, L. Loew. A General Computational Framework for Modeling Cellular Structure and Function. *Biophysical Journal*, Volume 73 , pp: 1135-1146, 1997.

Authors' Information

Sandra Gómez Canaval *– Department of Languages, Projects and Computer Systems, University College of Computer Science, Technical University of Madrid, Crta. de Valencia km. 7 - 28031 Madrid, Spain; e-mail: sgomez@eui.upm.es*

Major Fields of Scientific Research: Natural Computing, Systems Biology, Formal Languages Theory.

Fernando Arroyo Montoro *– Department of Languages, Projects and Computer Systems, University College of Computer Science, Technical University of Madrid, Crta. de Valencia km. 7 - 28031 Madrid, Spain; e-mail: farroyo@eui.upm.es*

Major Fields of Scientific Research: Natural Computing, Formal Languages Theory.

José Ramón Sánchez Couso - *Department of Languages, Projects and Computer Systems, University College of Computer Science, Technical University of Madrid, Crta. de Valencia km. 7 - 28031 Madrid, Spain;*

e-mail: jcouso@eui.upm.es

Major Fields of Scientific Research: Natural Computing, Formal Languages Theory.

Self-Organizing Architectural design based on Morphogenetic Programming

Nuria Gómez Blas, Luis F. de Mingo, Miguel A. Muriel

Abstract: *In this paper, we present our research into self-organizing building algorithms. This idea of self-organization of animal/plants behaviour interests researchers to explore the mechanisms required for this emergent phenomena and try to apply them in other domains. We were able to implement a typical construction algorithm in a 3D simulation environment and reproduce the results of previous research in the area. LSystems, morphogenetic programming and wasp nest building are explained in order to understand self-organizing models. We proposed Grammatical swarm as a good tool to optimize building structures.*

Keywords: *LSystems, Morphogenetic Programming, Swarm Computing, Particle Swarm Optimization.*

ACM Classification Keywords*: F.1.1 Theory of Computation - Models of Computation, I.2.6 Artificial Intelligence - Learning.*

Introduction

Many species of animals/plants exhibit simple individual actions but in groups are able to demonstrate quite complex emergent behaviours. This idea of self-organization of animal behaviour interests researchers to explore the mechanisms required for this emergent phenomena and try to apply them in other domains (such as computation). In this work, we look at lsystems, morphogenetic programming and nest building in social wasp species and use a computer model of this behaviour in order to build articial architectures. We propose the use of grammatical swarm to optimize generated structures.

LSystems

A Lsystem is a rule like description of a 3d form. It contains descriptions of parts and how they should be assembled together. A simulation program reads a Lsystem description and processes it into a 3d form, see figure 1, which can then be outputted in several formats. The description is applied to itself a number of times (recursion levels) so fractal and recursive forms are very easy to describe in a Lsystem. That's why they are used a lot for plants, trees and natural looking organic forms. By increasing the recursion level the form slowly grows and becomes more complex.

LSystems were created by biologist Lindenmayer [Lindenmayer,1990] as a method to simulate the growth of plants. But they really are an implementation of Chomsky's generative grammars. An LSystem is a set of terminal and non terminal symbols and some rules that define how non terminal symbols generate strings of new symbols. LSystems are also called recursive string substitution systems. LSystems are very useful to simulate some natural growing processes, like fungi, plants, or inorganic forms like crystals, or natural patterns. Since LSystems are basically recursive processes, they are good examples of self similarity, and are often considered a kind of fractals. The von Coch curve, a well known fractal object, can be produced easily with LSystems. LSystems are a very interesting tool for generative artists, in fact they let explore form within natural processes, they can also add scientific knowledge to artworks (grammars are linguistic models of natural processes), if well implemented they can offer an interactive laboratory for the investigation of natural and artificial forms [Lindenmayer,1990; Rozenberg,1992]. They can be applied to sculpture, painting, music and architecture. The good of LSystems is that even with simple rules can be reached a great complexity. In fact, LSystems are quite simple to implement, but the complexity of the growing process challenges all the artistic skills we may have. So if the use of LSystems is straightforward, the good use of LSystems is very complex and needs a lot of experimentation.

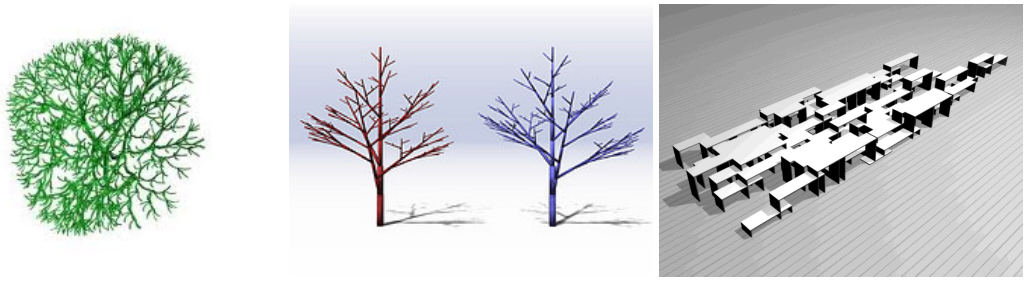


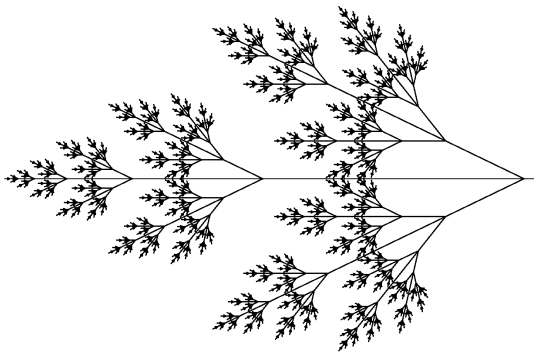
Figure 1: LSystems generated using LParser. [Lapré]

A generic implementation of LSystem is designed upon the following architecture:

- The language: A set of terminal and non terminal symbols. Non terminal symbols are symbols that may generate new strings of symbols of the language. Terminal symbols always remain the same, usually they are used to scale, rotate and move the elements of the system, see figure 2. In fact non terminal symbols usually are interpreted as 2D or 3D objects, like Logo's turtle graphic commands. But they could represent words or sounds as well.
- The generative grammar:
  - Axiom: a string of symbols of the language set used as a starting point to the substitution process.
  - Rules: a rule is a string of any symbol of the language set that will replace a non terminal symbol. It is possible to create rules for every non Terminal symbol, random rules, context sensitive rules, etc.
  - The substitution process: This is a recursive process that apply the rules to every non Terminal symbol of the axiom, thus generating a new and larger string. This new string will be used as axiom for a new substitution process. This process can be repeated many times, generating bigger and bigger strings.
  - The LSystem string: Is the final string, ready to be parsed.
- The parser: A parser is the module that will read the LSystem string token by token (char by char) and perform the right action, depending of the meaning of every symbol. Seymour Papert invented "Turtle graphics" as a system for translating a sequence of symbols into the motions of an automaton (the "turtle") on a graphics display. So "F" could mean go forward for some units in the 2D or 3D space, "+" could mean turn clockwise by sdome degrees, "a" could mean draw a line, etc. You have to learn the vocabulary of the specific LSystem implementation you are using.
- Editors: In the web there are many LSystems free editors, like LParser, Fractint or GDesign. While they are similar in their basic architecture, they can be and perform differently. GDesign is the only one with a visual editor and real time 2D / 3D display.

Morphogenetic programming

Morphogenesis is the biological process that causes an organism to develop its shape. It is one of three fundamental aspects of developmental biology along with the control of cell growth and cellular differentiation. The process controls the organized spatial distribution of cells during the embryonic development of an organism. Morphogenesis can take place also in a mature organism, in cell culture or inside tumor cell masses. Morphogenesis also describes the development of unicellular life forms that do not have an embryonic stage in their life cycle, or describes the evolution of a body structure within a taxonomic group. Morphogenetic responses may be induced in organisms by hormones, by environmental chemicals ranging from substances produced by other organisms to toxic chemicals



```
variables : X F
constants : + - [ ]
start     : X
rules     : (X -> F[+X][-X]FX), (F -> FF)
angle     : 25.7
```

Figure 2: Obtained figure using above grammar in a Lindenmayer system.

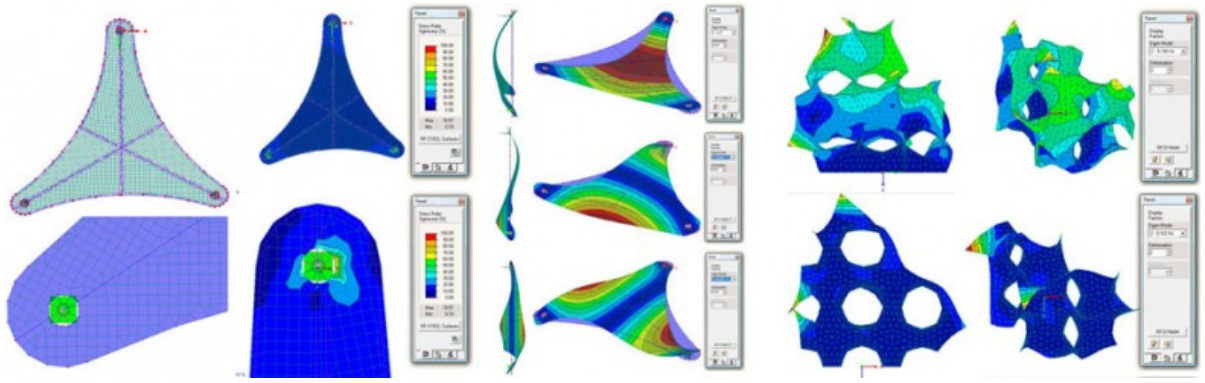


Figure 3: Minimal complexity prototype Houston 2011. [Tenu,2010]

or radionuclides released as pollutants, and other plants, or by mechanical stresses induced by spatial patterning of the cells.

Morphogenetic programming is the computational model of morphogenesis concept applied to architectural design, see figures 3, 4 and 5. Genetic algorithms, cellular automata, DNA models, neural networks, etc. take part on morphogenetic programming. On the other hand, swarm computing models could be considered macro models –they operate at individual level, no at molecular level-.

Achitectural behavior modeling based on cellular automata has been widely studied in recent years. Many of those models are based on the typical orthogonal mesh, the archetypical grid. The Cellular Automata orthogonal lattice is commonly used as the most neutral -thus general- geometrical base. It turns variables which are spatially extensive into their density-intensity equivalents and this immediately means that comparisons can be made [Batty,1999]. The geometric configuration of the spatial units used to represent the spatial data can have a profound effect, explaining why using spatial systems which neutralize the effect of configuration remove any bias caused by convoluted or distorting geometries.

The regular grid has other advantages, as the additional ability to work in layers without additional efforts to make different ones fit into the same system, and the significant work which proves its success even with highly refined Cellular Automata rules involving cultural and human factors [Portugali,1997].

For such reasons current models are centering their interest in orthogonal isotropic shape-constrained meshes (the regular grid); they are essentially analytic. Nevertheless, most of them function with a certain degree of deviation from classic Cellular Automata [Zhongwei,2003], redefining cell space, neighborhood, lattice and time concepts, which is necessary for some degree of flexibility.

Wasp nest building

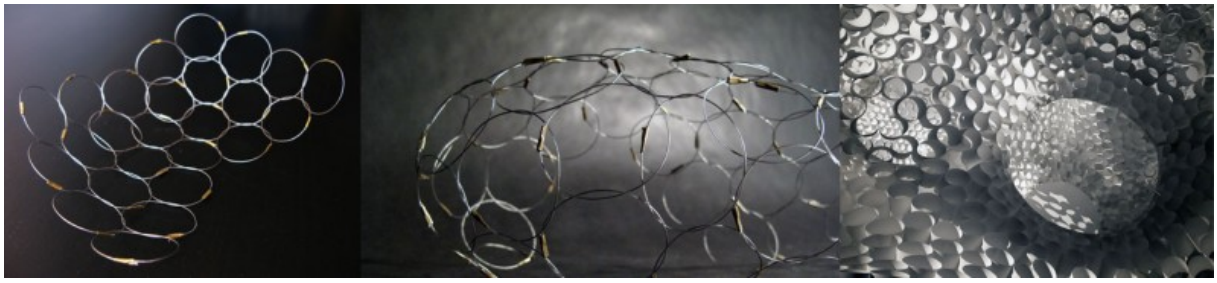


Figure 4: Minimal surfaces as architectural prototypes 2009. [Tenu,2010]

Figure 5: Self-organizing systems. Minimal complexity prototype London 2010. [Tenu,2010]

Figure 6: Examples of complex social wasp nests

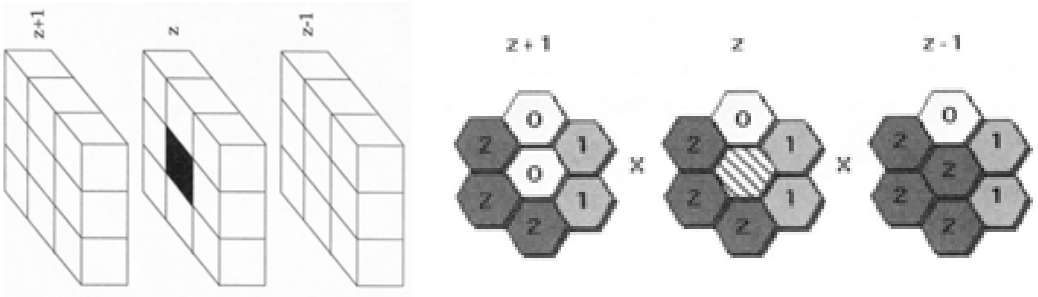
Theory proposed by Thorpe states that wasps store an internal blueprint of the type of nest structure that they build. While building the nest, the wasps compare their blueprint with the environment in order to decide which action to perform. Strong experimental evidence against the existence of such blueprint were later demonstrated. The experiments were conducted by modifying a nest structure during construction and observing the effect on the nest building behaviour. It was observed that the wasps were not able to finish the nest as per the blueprint. The experiments of Smith provided an insight onto the nest building strategy of social wasps by showing that cues in the environment seemed to be the driving force in the construction, see figure 6.

Stigmergy is defined as indirect communicating through the environment by leaving signs in the environment that could be picked up by others. Two forms of stigmergy have been identified: quantitative and qualitative. In quantitative or continuous stigmergy, the stimulus in the environment does not change, however, the amount of the stimulus can differ and evoke different responses to the stimulus. This model can be used to explain ant foraging behaviours and termite nest building.

Using the stigmergy model of wasp nest building, researchers have come up with a class of algorithms that can perform construction of artficial architectures. The algorithms use a swarm of agents that move randomly and independently in 3D space and try to match their stimulus-response systems with the local environment. The simulation space is usually a discrete cubic or hexagonal lattice hence the name lattice swarms, see figure 7. The elementary building blocks of the simulation are cubic or hexagonal bricks of different types. If an agent matches its local neighbourhood to a rule in the rule system, it deposits a brick of specified type at its current location in the lattice.

Several methods can be used for rule application. Rules can be matched deterministically or stochastically with a predefined probability. Only one set of rules can be matched or several sets of rules can be used either in seasonal manner or in a hierarchy of rule sets. The algorithms can be either coordinated or uncoordinated. In coordinated algorithms, several runs of the algorithm on the same rule system will yield architectures with common features. The architectures resulting from uncoordinated algorithms may not be similar. A high-level description of the construction algorithm is available in listing 1.

Grammatical Swarm



a) Squared lattice b) Hexagonal lattice

Figure 7: Lattices in wasp nest building

Algorithm 1 Wasp nest building basic algorithm

```
/* Initialization */
Construct lookup table
Put one initial brick at predefined site
for k = 1 to m do
    Assign agent k a random unoccupied site
end for
for t = 1 to t_max do
    for k = 1 to m do
        Sense local configuration
        if Local configuration is in lookup table then
            Deposit brick specified by lookup table
            Draw new brick
        else
            Do not deposit brick
        end if
        Move to randomly selected, unoccupied, neighboring site
    end for
end for
```

Grammatical Swarm (GS) relates Particle Swarm algorithm to a Grammatical Evolution (GE); genotype-phenotype mapping to generate programs in an arbitrary language. Grammatical Evolution (GE) is an evolutionary algorithm that can evolve computer programs in any language [O'Neill,2001; O'Neill,2003], and can be considered a form of grammar-based genetic programming. Rather than representing the programs as parse trees, as in GP [Koza,1999; Koza,2003], a linear genome representation is used. A genotype-phenotype mapping is employed such that each individual's variable length binary string, contains in its codons (groups of 8 bits) the information to select production rules from a Backus Naur Form (BNF) grammar. The grammar allows the generation of programs in an arbitrary language that are guaranteed to be syntactically correct, and as such it is used as a generative grammar, as opposed to the classical use of grammars in compilers to check syntactic correctness of sentences. The user can tailor the grammar to produce solutions that are purely syntactically constrained, or they may incorporate domain knowledge by biasing the grammar to produce very specific forms of sentences. BNF is a notation that represents a language in the form of production rules.

Algorithm 2 PSO Hybrid Algorithm

Creation of the population of particles $\{x^i\}$ in the interval
for Each particle i in the swarm **do**
  if $(f(x) > f(p))$ **then**
    for $(d = 1$ until $D)$ **do**
      $p_{id} = x_{id}$
    end for
  end if
  $g = i$
  for $j \in J$ **do**
    if $(f(p_j) > f(p_g))$ **then**
      $(g = j)$
    end if
  end for
  for $d = 1$ until D **do**
    $v_{id}(t) = v_{id}(t-1) + c_1\epsilon_1(p_{id} - x_{id}(t-1)) + c_2\epsilon_2(g_d - x_{id}(t-1))$
  end for
  $nmut = rand(1, M)$
  for $j = 1$ until $nmut$ **do**
    $k = rand(1, M)$
    $x_{ik} = p_{ik}$
  end for
  $nmut = rand(1, M)$
  for $j = 1$ until $nmut$ **do**
    $k = rand(1, M)$
    $x_{ik} = g_k$
  end for
end for

The equations for the particle swarm algorithm, see algorithm 2, are updated by adding new constraints to velocity and location dimension values, such us $vmax$ (bounded to 255), and dimension which is bounded to the range [0; 255] (denoted as $cmin$ and $cmax$, respectively). Note that this is a continuous swarm algorithm with real-valued particle vectors. The standard GE mapping function is adopted, with the real-values in the particle vectors being rounded up or down to the nearest integer value for the mapping process. In the current implementation of GS,

fixed-length vectors are used, which implies that it is possible for a variable number of dimensions to be used during the program construction genotypephenotype mapping process. A vector's elements (values) may be used more than once if wrapping occurs, and it is also possible that not all dimensions are used during the mapping process.

(This can happen whenever a program is generated before reaching the end of the vector). In this latter case, the extra dimension values are simply ignored and are considered as in trons that may be switched on in subsequent iterations.

Let us suppose the following BNF grammar:

```
<expr> :: = <expr><op><expr>
          | <var>
<op> :: = +
        | -
        | *
<var> :: = x
         | y
```

And the following genotype:

14	8	27	254	5	17	12

In the example individual (see figure 8), the left-most `<expr>` in `<expr> <op> <expr>` is mapped by reading the next codon integer value 240 and used in $240\%2 = 0$ to become another `<expr> <op> <expr>`. The developing program now looks like `<expr> <op> <expr> <op> <expr>`. Continuing to read subsequent codons and always mapping the left-most non-terminal the individual finally generates the expression `y * x - x - x + x`, leaving a number of unused codons at the end of the individual, which are deemed to be introns and simply ignored.

This is the classic benchmark problem in which evolution attempts to find the five input even-parity boolean function [Geva]. The grammar adopted here is:

```
<prog> ::= <expr>
<expr> ::= <expr> <op> <expr>
         | ( <expr> <op> <expr> )
         | <var> | <pre-op> ( <var> )
<pre-op> ::= not
<op> ::= "|" | & | ^
<var> ::= d0 | d1 | d2 | d3 | d4
```

The result is given by the best individual, see transcript bellow. Figure 9 shows a graphic with the best, average and variance of the swarm population. This figure has been obtained using the *GEVA* simulator [Geva].

```
( not ( d1 ) | d2 ^ d4 ) &
not ( d3 ) ^ ( not ( d1 ) &
( not ( d2 ) &
( not ( d2 ) |
( d1 ^ not ( d3 ) ^ not ( d1 ) ^
( not ( d1 ) ^ ( d0 | not ( d4 ) ) ) ) ) ^ d4 )
 ^ not ( d0 ) ) ^ d1
```

Wasp nest building and Grammatical swarm

This part describes the process to follow in order to obtain a structure that minimize a certain function, according to all previous self-organizing tools. Steps needed are the following:

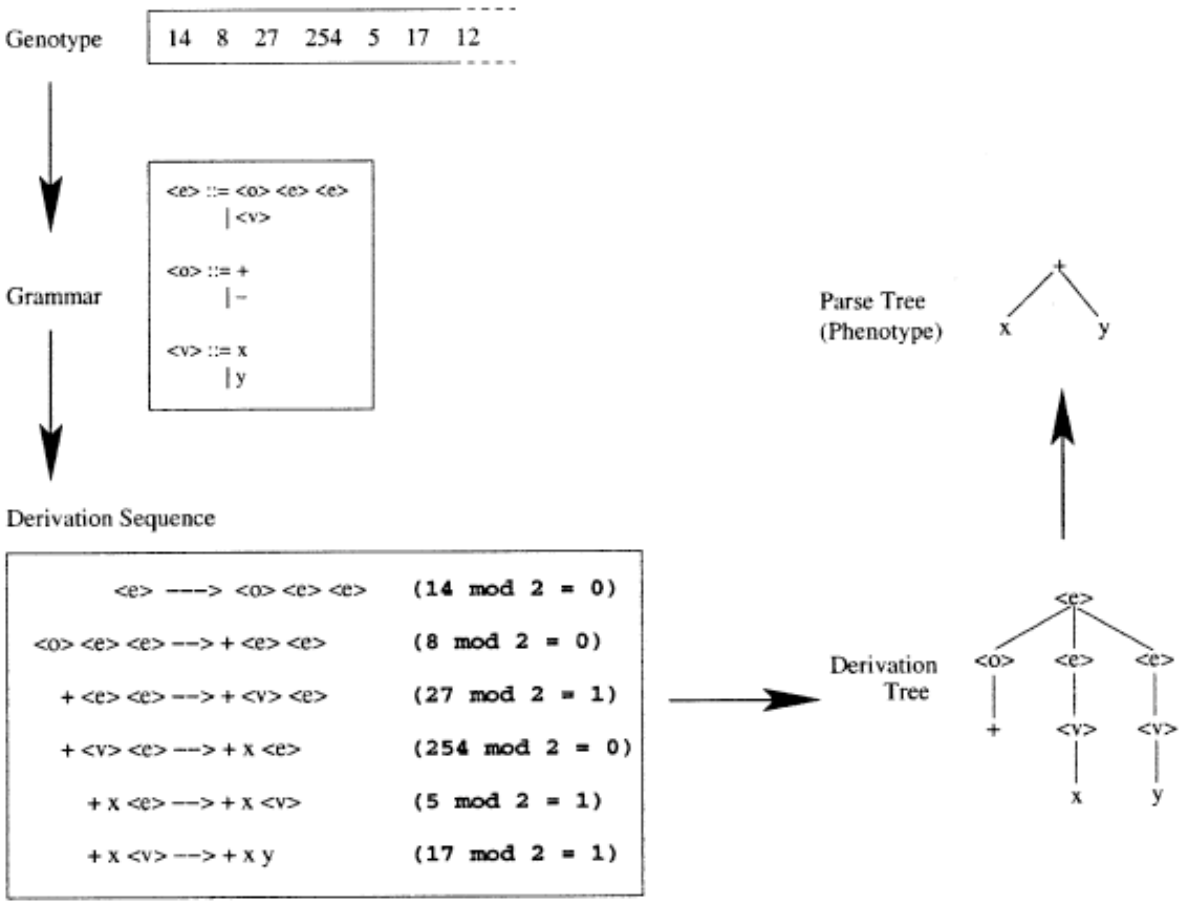


Figure 8: Grammatical Swarm concepts.

1. Define a grammar to generate patterns in a wasp nest building algotihm:

   Used grammar to generate self-organizing models (squared lattice, see figure 7, in case an hexagonal lattice is desired just generate 7 `<block>` in rules `<z+1>, <z>, <z-1>` instead 9), where 1 means there is a block and -1 means there is no block:

   ⟨*building_patterns*⟩ ::= ⟨*pattern*⟩ ':' ⟨*building_patterns*⟩
   | ⟨*pattern*⟩

   ⟨*pattern*⟩ ::= ⟨*z-1*⟩ ⟨*z*⟩ ⟨*z+1*⟩

   ⟨*z-1*⟩ ::= ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩

   ⟨*z*⟩ ::= ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*wasp*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩

   ⟨*z+1*⟩ ::= ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩ ⟨*block*⟩

   ⟨*block*⟩ ::= '-1'
   | '1'

   ⟨*wasp*⟩ ::= '0'

2. Define a fitness function for a given generated structure.

3. Apply the grammatical swarm algorithm to generate patterns:

   These patterns are the rules of the wasp nest building algorithm. Fitness function is computed once the structure is finished.

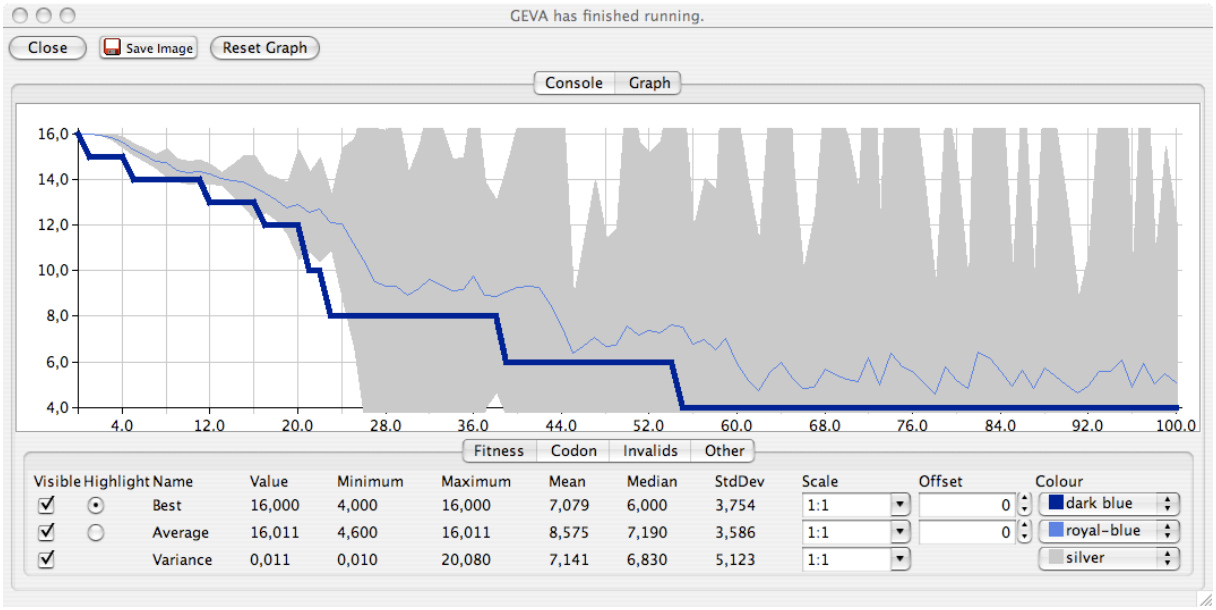


Figure 9: Results of *even-5-parity* problem simulated with GEVA.

Acknowledgements

The research was supported by the Spanish Research Agency projects TRA2010-15645 and TEC2010-21303-C04-02.

Bibliography

[Tenu,2010] Tenu, V.: Minimal Surfaces as Self-Organizing Systems. ACADIA Conference, Cooper Union in New York in October 2010. http://www.vladtenu.com

[Lapré] Laurens Lapré. http://laurenslapre.nl/index.html

[Lindenmayer,1990] Prusinkiewicz P., Lindenmayer A.;The Algorithmic Beauty of Plants. Springer-Verlag. 1990

[Rozenberg,1992] Rozenberg G., Salomaa A.; Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology, ISBN 978-3-540-55320-5. Springer-Verlag. 1992.

[O'Neill,2001] O'Neill M., Ryan C.; Grammatical Evolution, IEEE Trans. Evolutionary Computation, Vol. 5, No.4, 2001.

[O'Neill,2003] O'Neill M., Ryan C., Keijzer M., Cattolico M.; Crossover in Grammatical Evolution, Genetic Programming and Evolvable Machines, Vol. 4 No. 1. Kluwer Academic Publishers, 2003.

[Koza,1999] Koza J.R., Andre D., Bennett III F.H., Keane M.; Genetic Programming 3: Darwinian Invention and Problem Solving, Morgan Kaufmann, 1999.

[Koza,2003] Koza J.R., Keane M.,Streeter M.J., Mydlowec W., Yu J., Lanza G.; Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers, 2003.

[Geva] http://www.grammatical-evolution.org/

[Batty,1999] M. Batty, Y. Xie, Z. Sun. Modeling urban dynamics through GIS-based CA. In: Computers, Environment and Urban Systems. Elsevier, 1999.

[Zhongwei,2003] Sun Zhongwei. Simulating urban growth using cellular automata. Intl. Inst. for Geoinformation science and Earth observation. The Netherlands, 2003.

[Portugali,1997] J. Portugali, I. Benenson. Individuals' cultural code and residential self-organization in the city space. In: Proceedings of GeoComputation 97 and SIRC 97. University of Otago, New Zealand, 1997.

Authors' Information

Luis Fernando de Mingo López *- Dept. Organización y Estructura de la Información, Escuela Univesitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain; e-mail: lfmingo@eui.upm.es*
Major Fields of Scientific Research: Artificial Intelligence, Social Intelligence

Nuria Gómez Blas *- Dept. Organización y Estructura de la Información, Escuela Univesitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain; e-mail: ngomez@eui.upm.es*
Major Fields of Scientific Research: Bio-inspired Algorithms, Natural Computing

Miguel A. Muriel *- Dept. Tecnología Fotónica, ETSI Telecomunicación, Universidad Politécnica de Madrid, Avenida Complutense 30, Ciudad Universitaria, 28040 Madrid, Spain; e-mail: m.muriel@upm.es*
Major Fields of Scientific Research: Theoretical Computer Science, Microwave Photonics

TECHNICAL P-SYSTEMS: OPERATING IN THE STOCK MARKETS WITH TRANSITION P-SYSTEMS

Alberto Arteta, Angel Luis Castellanos, Nuria Gómez Blas

Abstract: *During last 50 years, the markets have been object of study. Technical and fundamental indicators have been used to try to predict the behaviour of the market and then execute buying or selling orders. This paper use the capabilities and functionalities of the transition P-systems to create an intelligent model which is able to process the information returned by the indicators in a efficient way .Membrane techniques provide a new approach of markets behaviour estimation*

Keywords: *Trading strategies, Technical indicators, Intelligent Transition P-Systems.*

Introduction

Markets indicators are numerous. Markets Operators try to determine the existence of trends. Once a trend is found it is generally a sign of earning profit by selling or buying depending on the trend. One of the most popular are the average (MACD) and the relative strength index (RSI). These 2 indicators can suggest and confirm the existence of a trend [Azzopardi, 2010]. Furthermore these indicators are helpful to establish the trend predictions patterns for the markets operators. This paper provides a new approach to an automatic view of these through the use of transition P-systems properties.

In order to do that this paper is organized as follows:

- Brief introduction of Transition P-systems
- Properties of the two market indicators.
- Theoretical Method to use the transition P-system in the stock markets with these indicators.
- Conclusions and further work.

Introduction to P-systems theory

Natural computing is a new field within computer science which develops new computational models. These computational models can be divided into three major areas:

- Neural networks.
- Genetic Algorithms
- Biomolecular computation.

Membrane computing is included in biomolecular computation. Within the field of membrane computing a new logical computational device appears: The P-system. These P-systems are able to simulate the behavior of the membranes on living cells. This behavior refers to the way membranes process information. (Absorbing nutrients, chemical reactions, dissolving, etc)

I. A P-system is a computational model inspired by the way the living cells interact with each other through their membranes. The elements of the membranes are called objects. A region within a membrane can contain objects or other membranes. A p-system has an external membrane (also called skin membrane) and it also contains a hierarchical relation defined by the composition of the membranes. A multiset of objects is defined within a region (enclosed by a membrane). These multisets of objects show the number of objects existing within a region. Any object 'x' will be associated to a multiplicity which tells the number of times that 'x' is repeated in a region.

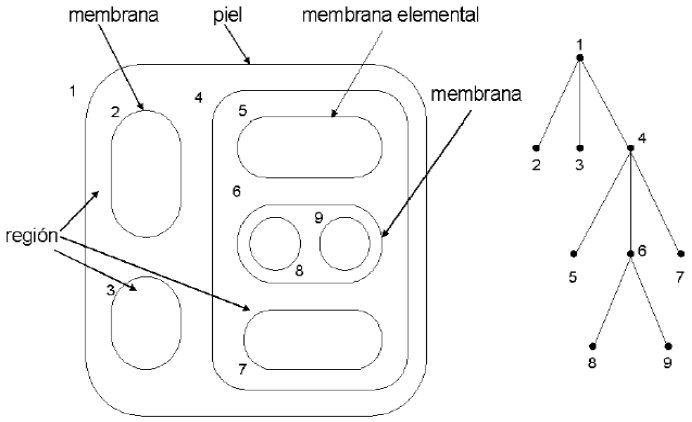


Figure 1. The membrane's structure (left) represented in tree shape (right)

According to Păun 's definition, a transition P System of degree n, n > 1 is a construct: [Păun, 1998]

$$\prod = \left(V, \mu, \omega_1, .., \omega_n, (R_1, \rho_1), .. (R_n, \rho_n), i_0\right)$$

Where:

1. *V* is an alphabet; its elements are called objects;
2. μ is a membrane structure of degree n, with the membranes and the regions labeled in a one-to-one manner with elements in a given set ; in this section we always use the labels 1, 2, .., n;
3. $\omega_i \; 1 \le i \le n$, are strings from V^* representing multisets over V associated with the regions 1, 2, .., n of μ
4. $R_i \; 1 \le i \le n$, are finite set of evolution rules over V associated with the regions 1, 2, .., n of μ; ρ_i is a partial order over $R_i \; 1 \le i \le n$, specifying a priority relation among rules of R_i . An evolution rule is a pair (u,v) which we will usually write in the form $u \to v$ where u is a string over V and v=v' or v=v' δ where v' is a string over $\left(V \times \{here, out\}\right) \cup \left(V \times \{in_j \; 1 \le j \le n\}\right)$, and δ is a special symbol not in. The length of u is called the radius of the rule $u \to v$
5. i_o is a number between 1 and n which specifies the output membrane of $\prod$.

Let *U* be a finite and not an empty set of objects and N the set of natural numbers. A *multiset of objects* is defined as a mapping:

$$M : V \to \mathrm{N}$$

$$a_i \to u_1$$

Where a_i is an object and u_i its multiplicity.

As it is well known, there are several representations for multisets of objects.

$$M = \{(a_1, u_1), (a_2, u_2), (a_3, u_3)...\} = a_1^{u_1} \cdot a_2^{u_2} \cdot a_n^{u_n} .....$$

Evolution rule with objects in *U* and targets in *T* is defined by $r = (m, c, \delta)$ where $m \in M(V), c \in M(VxT)\ and\ \delta \in \{to\ dissolve, not\ to\ dissolve\}$

From now on *'c'* will be referred to as the consequent of the evolution rule *'r'*

The *set of evolution rules* with *objects* in *V* and targets in *T* is represented by R *(U, T).*

We represent a rule as:

$$x \to y \quad or \quad x \to y\delta$$

where x is a multiset of objects in M((*V*)xTar) where Tar ={here, in, out} and y is the consequent of the rule. When δ is equal to "dissolve", then the membrane will be dissolved. This means that objects from a region will be placed within the region which contains the dissolved region. Also, the set of evolution rules included on the dissolved region will disappear.

P-systems evolve, which makes it change upon time; therefore it is a dynamic system. Every time that there is a change on the p-system we will say that the P-system is in a new transition. The step from one transition to another one will be referred to as an evolutionary step, and the set of all evolutionary steps will be named computation. Processes within the p-system will be acting in a massively parallel and non-deterministic manner. (Similar to the way the living cells process and combine information).

We will say that the computation has been successful if:

1. The halt status is reached.
2. No more evolution rules can be applied.
3. Skin membrane still exists after the computation finishes.

Properties of the markets indicators (MACD, RSI)

1. MACD

Developed by Gerald Appel in the late seventies, the Moving Average Convergence-Divergence (MACD) indicator is one of the simplest and most effective momentum indicators available. The MACD turns two trend-following indicators, moving averages, into a momentum oscillator by subtracting the longer moving average from the shorter moving average. As a result, the MACD offers the best of both worlds: trend following and momentum. The MACD fluctuates above and below the zero line as the moving averages converge, cross and diverge. Traders can look for signal line crossovers, centerline crossovers and divergences to generate signals. Because the MACD is unbounded, it is not particularly useful for identifying overbought and oversold levels.

The MACD indicator is special because it brings together momentum and trend in one indicator. This unique blend of trend and momentum can be applied to daily, weekly or monthly charts. The standard setting for MACD is the difference between the 12 and 26-period EMAs. Chartists looking for more sensitivity may try a shorter short-term moving average and a longer long-term moving average. MACD(5,35,5) is more sensitive than MACD(12,26,9) and might be better suited for weekly charts. Chartists looking for less sensitivity may consider lengthening the moving averages. A less sensitive MACD will still oscillate above/below zero, but the centerline crossovers and signal line crossovers will be less frequent.

Figure 1. MACD 200 periods of instrument EURUSD

2. RSI

The relative strength index (RSI) is a technical indicator used in the analysis of financial markets. It is intended to chart the current and historical strength or weakness of a stock or market based on the closing prices of a recent trading period. The indicator should not be confused with relative strength.

The RSI is classified as a momentum oscillator, measuring the velocity and magnitude of directional price movements. Momentum is the rate of the rise or fall in price. The RSI computes momentum as the ratio of higher closes to lower closes: stocks which have had more or stronger positive changes have a higher RSI than stocks which have had more or stronger negative changes.

The RSI is most typically used on a 14 day timeframe, measured on a scale from 0 to 100, with high and low levels marked at 70 and 30, respectively. Shorter or longer timeframes are used for alternately shorter or longer outlooks. More extreme high and low levels—80 and 20, or 90 and 10—occur less frequently but indicate stronger momentum.

The relative strength index was developed by J. Welles Wilder and published in a 1978 book, New Concepts in Technical Trading Systems, and in Commodities magazine (now Futures magazine) in the June 1978 issue.[1] It has become one of the most popular oscillator indices.[2]

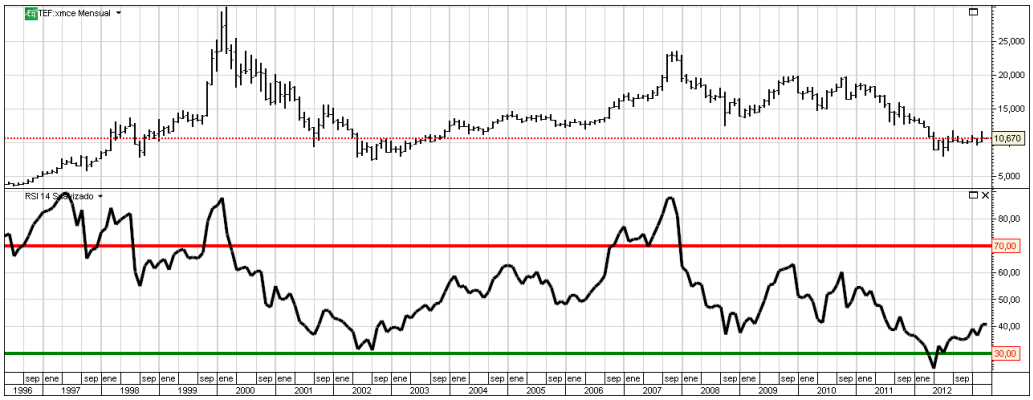


Figure 2. Index Relative (RSI) of strength for the instrument EURUSD

Transition P-systems working with indicators

Once the characteristics of P-systems and the indicators are explained separately, a proposal for building a community P-system working with the indicators is made. In particular, it is possible to define a P-system able to learn through the use of the technical indicators

As there are regions in the living cells a master membrane will synchronize the rest, which contains the info from the indicators

The way to do this is:

In the regions, given a set of membranes $M = \{m_i \mid i \in \mathrm{N}, 1 \le i \le n\}$ where m_i is a membrane, MACD and RSI are placed in it. In the end every membrane sends a signal to the master membrane, which determine one the following signals = {buy,sell,null)

If the info sent by the membranes does not return any buying or selling signal, the P.system returns null which means that buying or selling is not recommended.

The P-system considers three major stages:

- Static structure of the P-system;
- Dynamic behavior of the P-system;
- Synchronization between membranes.

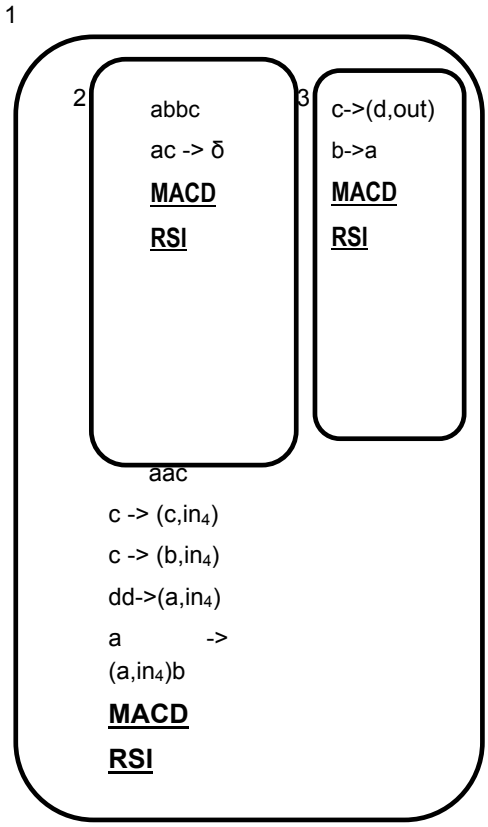


Figure 3. P-system with technical indicators, Technical P-system

The result will be placed outside of the membrane after the P-system finishes its execution.

Every membrane gets control the info and sends the signal to the master membrane. Every membrane is storing the information about the information (MACD and RSI) processing for every region in a given moment. The P-system evolves following the master membrane patterns. Moreover the master membrane outside the P-system controls the execution steps of the P-system.

Membranes learn from the times that evolution rules are applied and the results obtained based on the technical indicators. The aim of this P-system supervised by the master membrane is not just to returning buying or selling signals but also to improve the time the signals are returned so the operations can be placed in the market immediately.

According to Paun's model all the rules application processes occur in every region in a parallel manner. Moreover the process is non-deterministic.

- The evolution rules that are applied in every membrane m_i . The number of times that every rule r_i is applied to obtain an extinguished multiset are referred as ki.

For every membrane records are stored as: $((r_1,1),(r_2,2),..,(r_i,7),..,(r_m,3)$ +the technical indicators and the success or failure of placing an operation. In that way the P-system can learn.

The more time the P-system works the better results are obtained. At the end tof the computation he master membrane will return the final signal (buying, selling, null) based on the info and hits. The optimal number of times the rules are applied determines the execution time to be reduced. This will reduce the execution time.

In order to implement this system it is important to take into account:

- The need of auxiliary space to store information about the rules' election, hits and failures of every membrane output.
- The need of extra time to calculate the times that the evolution rules have to be applied
- In the beginning the results might take shorter. This might occur because as there are not enough information from the P-system computation, the decisions made by the robot about the rules could be worse. As the P-system learns, outputs can be more accurate. When changing non-determinism by intelligent elections, Paun's biological model is not useful anymore to implement the living cells behavior which means that these technical P-systems cannot be used to implement the living cells model.

Conclusions

The idea of implementing this biological model is taking advantages of the parallelism and synchronization to perform markets operations. This theoretical model proposed here shows that it'd be possible to obtain and to place accurate orders in the markets in a fast way with a small margin of error. As a first approach it shows it might be a good starting point to introduce the membrane models into the markets so they can theoretically provide a large source of signals to operate successfully.

Bibliography

[Azzopardi (2010)]. *Behavioural Technical Analysis: An introduction to behavioural finance and its role in technical analysis*. Harriman House. ISBN 1905641419.

[Ciobanu, Pérez-Jiménez, Ciobanu, Păun 2006] M. Pérez-Jiménez, G. Ciobanu, Gh. Păun. Applications of Membrane Computing, Springer Verlag. Natural Computing Series, Berlin, October, 2006.

[Fernández, Castellanos, Arroyo, tejedor, Garcia 2006] L. Fernández, J.Castellanos, F. Arroyo, J. tejedor, I.Garcia. New algorithm for application of evolution rules. Proceedings of the 2006 International Conferenceon Bioinformatics and Computational Biology, BIOCOMP'06, Las Vegas, Nevada, USA, 2006.

[Fernández, Martínez, Arroyo, Mingo 2005] L. Fernández, V.J. Martínez, F. Arroyo, L.F. Mingo. A Hardware Circuit for Selecting Active Rules in Transition P Systems. Proceedings of International Workshop on Theory and Applications of P Systems.Timisoara (Romania), September, 2005.[

[Pan, Martin 2005] L. Pan, C. Martin-Vi de. Solving multidimensional 0-1 knapsack problem by P systems with input and active membranesl. Journal of Parallel and Distributed Computing Volume 65 , Issue 12 (December 2005)

[Păun 1998] "Computing with Membranes", Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Păun 2000] Gh. Păun. Computing with Membranes. Journal ofComputer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Păun 2005] Gh. Păun. Membrane computing. Basic ideas, results, applications. Pre-Proceedings of First International Workshop on Theory and Application of P Systems,Timisoara (Romania), pp. 1-8, September , 2005.

[Qi, Li, Fu, Shi, You 2006] Zhengwei Qi, Minglu Li, Cheng Fu, Dongyu Shi, Jinyuan You. Membrane calculus: A formal method for grid transactions. Concurrency and Computation: Practice and Experience Volume18,Issue14 , Pages1799-1809. Copyright © 2006 John Wiley & Sons, Ltd.

Authors' Information

Alberto Arteta Albert *– Associate professor U.P.M., Crtra Valencia km 7, Madrid-28031, Spain;*
e-mail: aarteta@eui.upm.es
Research: Membrane computing, Education on Applied Mathematics and Informatics

Angel Luis Castellanos *– Associate professor U.P.M., Crtra Valencia km 7, Madrid-28031, Spain; e-mail: angel.castellanos@upm.es*
Research: Membrane computing, Education on Informatics

Nuria Gómez Blas *– Associate professor U.P.M. Crtra Valencia km 7, Madrid-28031, Spain; e-mail: ngomez@eui.upm.es*
PhD on Artificial Intelligence, Education on Biology and Informatics

OVERLAPPING RANGE IMAGES USING GENETIC ALGORITHMS

Fernando Ortega, Javier San Juan, Francisco Serradilla, Dionisio Cortes

Abstract: *This work introduces a solution based on genetic algorithms to find the overlapping area between two point cloud captures obtained from a three-dimensional scanner. Considering three translation coordinates and three rotation angles, the genetic algorithm evaluates the matching points in the overlapping area between the two captures given that transformation. Genetic simulated annealing is used to improve the accuracy of the results obtained by the genetic algorithm.*

Keywords: *Range images, genetic algorithms, overlaping point cloud captures, genetic simulated annealing.*

Introduction

There are many different tools available on the market to convert range images taken from a three-dimensional grabber on representable models in real-time on a computer. However, a common characteristic in existing tools is the need for a large amount of manual work for the overlapping of the different captures that are required to register the whole figure.

In this paper, we present a novel solution which aims to automate the range image overlapping process using genetic algorithms (GA's). The GA proposed will receive as input two partially overlapped range images and it must return the set of rotations and translations that must be performed on the second range image to ensure it is perfectly aligned with the first.

This paper's main challenge is to design a GA that manages to resolve the problem. The size of the search space in which the GA moves is very large, and therefore, great care must be taken with the design of the algorithm in order to resolve the problem in a reasonable amount of time. The GA will mainly focus on the use of Genetic Simulated Annealing (GSA) to increase the GA convergence speed, and also on the design of an effective fitness capable of providing the overlapping level of two captures.

The rest of the paper is structured as follows:

- In Section 2 we will analyze the work related to the study presented here.
- In Section 3 we will formalize the proposed solution, explaining the GA developed in detail.
- In Section 4 we will present the experiments that we use to verify the performance of the GA.
- In Section 5 we will show the results of experiments with the algorithm developed.
- In Section 6 we will present the conclusions obtained and we will develop the possible future papers that have arisen as a result of this study.

Related work

The process of computer representation of real figures starts with the input of data from a series of range images taken by a grabber. From this series, we have to obtain one single range image in a unique reference system. In addition, we assume that each of the images captured may have been taken in random position and rotation

conditions. For that reason, the objective of this phase is to find a series of transformations which, based on a random reference system for each of the captures, merges the information from all of these in one single reference system.

The main research into this field belongs to [1]. In this work, they tackle manual calibration techniques such as the overlapping of multiple captures with random transformations. They propose the use of an inverse calibration system, in which the viewing frame is associated with certain coordinates of the 3D scene (which can be performed via the pre-identification of certain elements captured on the view), allowing each of the capture points to be associated with certain coordinates of the scene.

[11] propose an interesting approach for registration in various phases. Firstly, they perform edge-based segmentation for broad overlapping which guides the next phase in fine-grain overlapping based on the ICP (iterative closest point) algorithm.

[10] covers techniques for the alignment of multiple capture points. He proposes the association of different capture points via a point-plane approach.

[3] design a new technique, called DARCES (based on the RANSAC technique) for the overlap of overlapped range images, using their own algorithms for the cataloguing of dispersed objects in range images.

[7] seeks an algorithm that allows the transformation matrix to be found between two systems of coordinates with different rotation. He calculates the rotation between the two systems in search of the appropriate quaternion.

The case we are dealing with has some peculiarities which make it difficult to apply the solutions proposed in the articles. On the one hand, it must be possible to find the overlapping areas without the need to resort to the use of markers. And, although using GPS and scanner compass information maybe of help when performing the process, the system could not depend on the availability of this data, but rather, on the comparison of the distribution of the points in each of the captures to be compared. Nor is it possible to resolve the problem by searching for a change of base matrix which overlaps one capture with the other, as it will be discrete areas of each capture which must be overlapped, and here lies the main problem to be resolved.

Genetic algorithm

Genetic algorithms (GA's) are a type of evolutive algorithms used to resolve search and optimization problems [5, 6]. They are based on simulating the evolutive process produced in nature to resolve problems of adaptation to the environment. GA's simulate, via populations of individuals, the evolution suffered through different operators called genetic operators, such as reproduction, crossover and mutation. Each operator plays a different role in the evolution of the population. This way, the reproduction can be understood as a competition, whilst the rest, like crossover and mutation are capable of creating new individuals from the existing ones in the population.

The individuals in our GA will represent the rotations and translations that must be carried out on the second capture to ensure it overlaps with the first. We have opted to use a GA with binary coding and we assign 6 chromosomes to each individual of the population, 3 for the translations and 3 for the rotations. The chromosomes of the translations have 8 bits, whilst those of the rotations have 12 bits. Figure 1 contains a graphical representation of the coding of the individuals and their chromosomes.

The genetic operators used in the GA are as follows:

- Selection. The chosen method is the Roulette Wheel Selection operator. That is to say, the selection probability of an individual depends on its fitness level.
- Crossover. We use the one-point crossover technique. Two parents are selected, a cut-off point is chosen and the two parents are combined to generate two children.

- Mutation. We use a single point mutation technique in order to introduce diversity. Some bits are modified randomly according to a mutation probability.
- Replacement operator: replacement with elitism. The best individuals of the current population move on to the next population without being modified (elitism) and the rest of the individuals are obtain using crossovers.

Since we have obtained quick satisfactory results using these four classical operators, we have not used other possible operators like migration, regrouping or colonization-extinction.

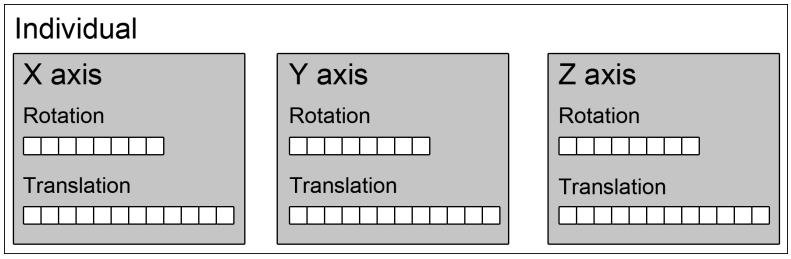


Figure 1. Representation of the individuals and their chromosomes of the GA developed.

Fitness function

In a GA the fitness function is fundamental as it allows the convergence of the population towards the optimum solution of the problem we want to resolve. In our case, we have developed a fitness function that allows us to check the level of overlapping of two captures simply and effectively. The basic idea is to divide the overlapped area into a series of cells and to count the number of points of each capture which belong to each cell. The cells that contain an equal number of points from both captures will have a high level of overlapping, whilst those that have a different number of points will have a very low level of overlapping.

Formally, in order to calculate the fitness we need to define the following elements:

- Let $C_1 = \{p_{1,1}, p_{1,2}, \mathrm{K}\ , p_{1,n-1}, p_{1,n}\}$ be the set of *n* points of the capture 1. (1)
- Let $C_2 = \{p_{2,1}, p_{2,2}, \mathrm{K}\ , p_{2,n-1}, p_{2,n}\}$ be the set of *m* points of the capture 2. (2)
- We define $p_{c,i}^x$ as the value of the coordinate of point $p_{c,i}$ on the X axis. (3)
- We define $p_{c,i}^y$ as the value of the coordinate of point $p_{c,i}$ on the Y axis. (4)
- We define $p_{c,i}^z$ as the value of the coordinate of point $p_{c,i}$ on the Z axis. (5)

We also need to define the zone in which both captures intersect:

- Let $I_{i,j}$ be the set of points belonging to the capture *i* which overlap with the capture *j*. (6)
- We define $\left[\min_{i,j}^x, \max_{i,j}^x\right]$ as the interval of the X axis in which the points of the intersection of capture *i* with capture *j* move. (7)
- We define $\left[\min_{i,j}^y, \max_{i,j}^y\right]$ as the interval of the Y axis in which the points of the intersection of capture *i* with capture *j* move. (8)
- We define $\left[\min_{i,j}^z, \max_{i,j}^z\right]$ as the interval of the Z axis in which the points of the intersection of capture *i* with capture *j* move. (9)
- Likewise, we define the operator $\#Set$ as the number of points of a set. E.g. $\#C_1$ represents the number of points of capture 1. (10)

We divide the volume overlapped in N^3 cells which will determine the level of overlapping of both captures in different places of the overlapped space. We define $cell_{1,2}^{r,s,t}$ as the set of points of capture 1 which are overlapped with capture 2 and which belong to the r, s, t cell.

$$cell_{1,2}^{r,s,t} = \left\{ p_{1,i} \in C_1 \mid \min_{1,2}^x + r \cdot \left(\max_{1,2}^x - \min_{1,2}^x\right) \Big/ N \le p_{1,i}^x < \min_{1,2}^x + (r+1) \cdot \left(\max_{1,2}^x - \min_{1,2}^x\right) \Big/ N \right.$$
$$\wedge \min_{1,2}^y + s \cdot \left(\max_{1,2}^y - \min_{1,2}^y\right) \Big/ N \le p_{1,i}^y < \min_{1,2}^y + (s+1) \cdot \left(\max_{1,2}^y - \min_{1,2}^y\right) \Big/ N$$
$$\left. \wedge \min_{1,2}^z + t \cdot \left(\max_{1,2}^z - \min_{1,2}^z\right) \Big/ N \le p_{1,i}^z < \min_{1,2}^z + (t+1) \cdot \left(\max_{1,2}^z - \min_{1,2}^z\right) \Big/ N \right\} \quad (11)$$

In the same way, we define $cell_{2,1}^{r,s,t}$ as the set of points of capture 2 which are overlapped with capture 1 and which belong to the r, s, t cell.

$$cell_{2,1}^{r,s,t} = \left\{ p_{2,i} \in C_2 \mid \min_{2,1}^x + r \cdot \left(\max_{2,1}^x - \min_{2,1}^x\right) \Big/ N \le p_{2,i}^x < \min_{2,1}^x + (r+1) \cdot \left(\max_{2,1}^x - \min_{2,1}^x\right) \Big/ N \right.$$
$$\wedge \min_{2,1}^y + s \cdot \left(\max_{2,1}^y - \min_{2,1}^y\right) \Big/ N \le p_{2,i}^y < \min_{2,1}^y + (s+1) \cdot \left(\max_{2,1}^y - \min_{2,1}^y\right) \Big/ N$$
$$\left. \wedge \min_{2,1}^z + t \cdot \left(\max_{2,1}^z - \min_{2,1}^z\right) \Big/ N \le p_{2,i}^z < \min_{2,1}^z + (t+1) \cdot \left(\max_{2,1}^z - \min_{2,1}^z\right) \Big/ N \right\} \quad (12)$$

With $o^{r,s,t}$ we denote the level of overlapping of the r, s, t cell. The idea is that the cells that contain many points from both captures have a high level of overlapping, whilst those which have a different level of points from both captures have a low level of overlapping. The calculation of the level of overlapping is reflected in Table 1.

Table 1. Level of overlapping according to the number of points of each range image in the r, s, t cell.

$o^{r,s,t}$	$\#\, cell_{1,2}^{r,s,t} = 0$	$0 < \#\, cell_{1,2}^{r,s,t} \le Few$	$Few < \#\, cell_{1,2}^{r,s,t}$
$\#\, cell_{2,1}^{r,s,t} = 0$	*LO*	*-LO*	*-HO*
$0 < \#\, cell_{2,1}^{r,s,t} \le Few$	*-LO*	*MO*	*-MO*
$Few < \#\, cell_{2,1}^{r,s,t}$	*-HO*	*-MO*	*HO*

Where *HO* represents a constant assigned to High Overlapping, *MO* a constant assigned to Medium Overlapping and *LO* a constant assigned to Low Overlapping. Likewise, *-HO*, *-MO* and *-LO* represent the same negated values to indicate Low Non Overlapping, Medium Non Overlapping and High Non overlapping, respectively. The value *Few* quantify the number of points that should have a range image to have a high grade of overlapping / non overlapping.

The fitness will be calculated as the summation of the overlapping of all the cells.

$$fitness = \sum_{r=0}^{N} \sum_{s=0}^{N} \sum_{t=0}^{N} o^{r,s,t} \quad (13)$$

Genetic simulated annealing

One of the problems of GA's is that, although they progress very well in the first stages of learning, they have difficulties with the fine adjustment of the final solution. In our problem, the fitting of scenes, this is translated into the rapid establishment of an approximated fit and a much longer delay to find the definitive value of rotation and translation of the fragments of the scene.

In order to improve this weakness of GA's, [2] theoretically proposed a new family of algorithms which combine the GA with a well-known technique in the field of optimization, Simulated Annealing (SA), in what they called Genetic Simulated Annealing (GSA).

Subsequent research has provided algorithms which mainly modify the operator selection of the GA to incorporate a temperature factor [4, 15], based on a neighborhood criteria among individuals which is gradually restricted as the temperature drops. This method reminds us of the operation of the neighborhood environment in self-organizing maps [8]. This approach has been used in various applications, among which we can highlight telecommunications network planning [14], economics [15], or forecast prediction [9].

In our paper, in order to reduce the computational cost of the search for the optimum fit of the fragments of the scene we have introduced the SA in the mutation and crossover operator. Given that we use a binary coding alphabet, we propose to use the temperature factor (*T*) of the annealing to rule out the *T* less significant bits in early stages of the process, and, as the temperature drops, to gradually incorporate more bits.

The effect of ignoring the less significant bits is to eliminate degrees of freedom from the search process in the solutions space, by sacrificing the precision in the adjustment obtained, which allows the algorithm to quickly converge towards the optimum solution. As more bits are incorporated to the mutation operator and the crossover operator the precision of the solution contributed is increased, ultimately reaching the precision required in the design in accordance with the total number of bits chosen to represent the values of position and orientation of the scenes.

Specifically, we propose that the mutation and crossover operator is applied to the most significant (*L -T*) bits of each rotation or translation, taking for *L* the number of bits of the translation / rotation and for T the value of the temperature represented by the number of bits to be ruled out. E.g. if the rotation has 12 bits, and the value of *T* is 4, the crossover and mutation operations will only be performed on the 8 most significant bits. The temperature is updated every certain number of generations (to be defined in each problem) increasing by 1 the number of significant bits.

In the same way, we will include the temperature concept in the fitness function. Our objective is that, as the GA advances, we increase the number of cells (*N*) into which the overlapped area is divided. This way, at the start, we will be able to carry out a poorly detailed adjustment of the captures and, as the generations elapse, the level of overlapping will be much more detailed. We must define an initial value for *N* and, every certain number of generations, we increase the value of *N* by one unit.

Experiments

In this section we are going to explain the experiments that we have used to test the algorithm. We have chosen two range images: a rosette with 528977 points and a house with 515255 points. The two captures were cut in half with a 38,5% overlap, which means that there is a 38,5% of the fragment in the other.

The fitness of the GA is highly dependent on the parameters. After several tests, we found optimal values for the proper functioning of the GA, this varies depending on the capture. The Table 2 shows the parameters used in the experiments.

Table 2. Parameters used in the experiments.

Parameter	House	Rosette
Number of generations	1000	1000
Population size	100	120
Crossover probability	0,7	0,7
Mutation probability	0,01	0,01
Elitism percent	5%	5%

Initial number of cells (*N*)	16	20
Number of generations to update *N*	50	50
Initial discarted bits (*T*)	6	6
Number of generations to update *T*	25	25
Few (points)	500	500
High Overlapping (*HP*)	0.001	0.001
Medium Overlapping (*MO*)	0.01	0.01
Low Overlapping (*LO*)	0.1	0.1

In order to see the evolution of fitness, we are going to calculate the maximum fitness and we observe how good the fitness is as the generations evolve. The maximum fitness is calculated when we cut the captures, at the end of the cut, the two captures are perfectly overlapped and in this moment the maximum fitness is calculated. The maximum fitness is used to normalize the graph, with different captures fitness results may vary, so it is a good idea normalize to compare and have a way to measure how good it is in different cases.

Results

The figures 2 and 3 show the evolution of the genetic algorithm, in a graphical way. The figure 2a and 3a shows the first state of the algorithm, with the two fragments in completely different positions. The figure 2b and 3b shows the firsts steps of the algorithm, the fragment is starting to aproach to a good solution, the algorithm is in the 200 generation. The figure 2c and 3c shows the lastest steps of the algorithm, the fragment is almost in its position, the algorithm is in the 600 generation. The figure 2d and 3d shows the result of the algorithm, the fragment is in a position near to the perfect position. The adjustment is better in the rosette than in the house because is a complex capture, although the results are still good and we can see a graph of a typical GA, evolving quickly at first, and later doing small adjustments.

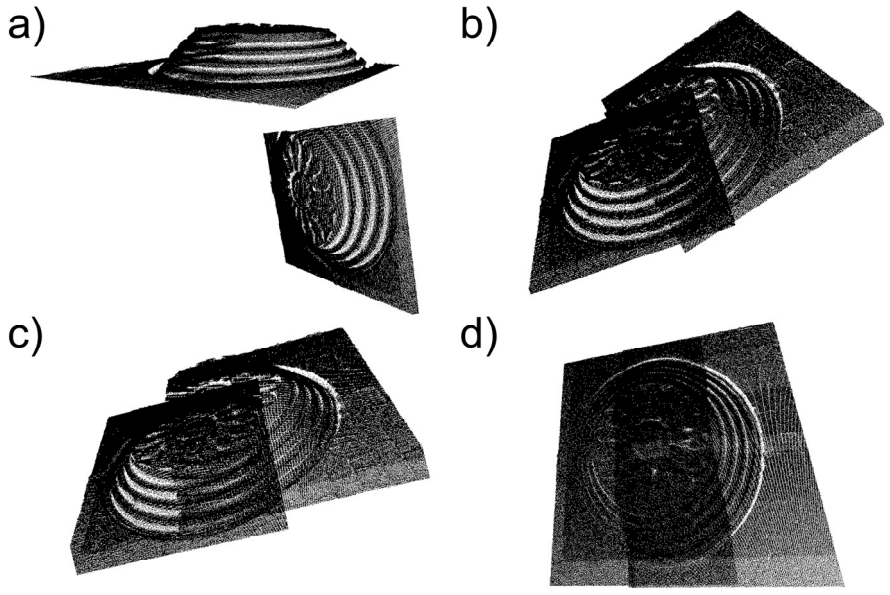


Figure 2. Results of overlapping the range image of the rosette with the proposed GA.

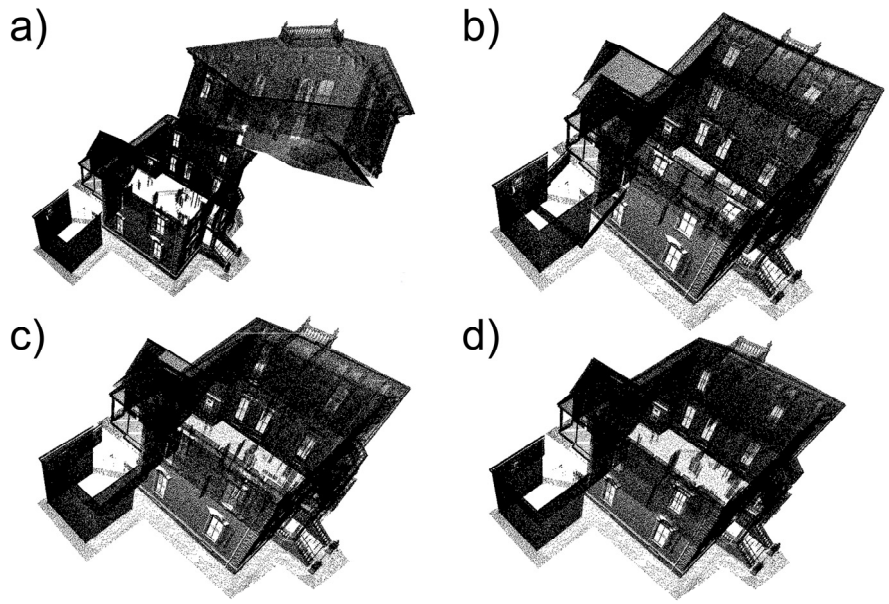


Figure 3. Results of overlapping the range image of the house with the proposed GA.

The figure 4 shows the evolution of the individuals in the genetic algorithm. The graph contains the average of the population normalized fitness in each generation. At first, the population evolves quickly and then, GA makes a slow and more precise adjustment. This is due to the nature of the GA, which at first have a quickly convergence and later have a slowly adjustment, and to the inclusion of the GSA which allows a better adjustment as the GA evolutions.

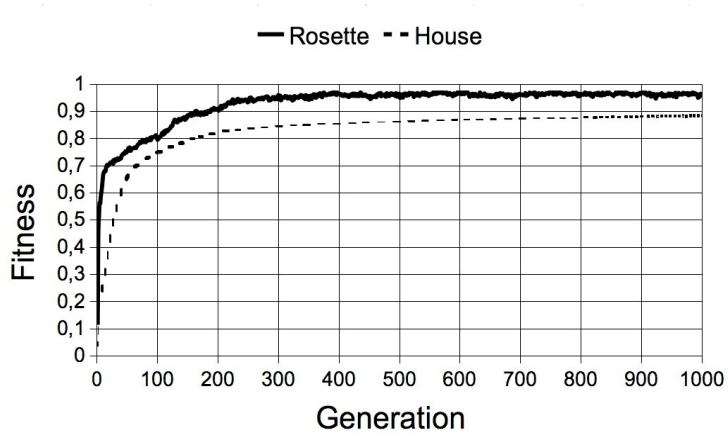


Figure 4. Normalized fitness evolution during de GA with de rosette and the house.

Conclusions and future work

This article presents a strategy to tackle the problem of overlapping two captures using GA's automatically instead of manually as has been the case until the now.

GA's are a great help for the resolution of many optimization problems. As we have seen, it is also possible to use them to resolve the problem of overlapping clouds of points, which is a highly complex problem as, a priori, we have no information about the placement of the captures and they can have random translations and rotations.

In the article we have presented a specific GA and we have placed special importance on including the GSA and on the development of the fitness function. After various experiments we have seen that the GA's behavior greatly depends on the parameters that we use for its execution. This fact highlights that, in order to achieve the overlapping between a pair of range images, we must be perfectly aware of the nature of these range images in order to correctly establish the parameters to be used. If the parameters are not adjusted correctly, the algorithm will converge towards suboptimal solutions and, therefore, towards erroneous overlapping.

Despite the impediment of the parameters, the proposed algorithm continues to present huge benefits compared to the manual overlapping of captures, as, with a little experience, the parameters are quickly adjusted and the algorithm is capable of returning the captures perfectly overlapped.

Normally, the scanners present geographical coordinates, as well as a compass for their orientation. In this paper, we have dispensed with any help as we cannot guarantee that all the captures presented contain this information, and, therefore, the proposed GA has an added value as it works with the minimum information possible on the range images.

In a future paper we will continue to study different fitness functions to see which offers better results. It is possible to establish other fitness functions which offer advantages as regards the precision of the algorithm or the time of its calculation.

It is more and more common to have equipment with multiple process cores, and, therefore, parallelizing these algorithms to reduce their execution time is essential. GA's are highly parallelizable. With little effort we could parallelize the algorithm to obtain faster response times on moving to the exploitation phase.

In this article, we use range images with the same density of points, and, therefore, another problem to resolve is the adaptation of the proposed GA to captures that present different point density. The current fitness is not capable of detecting the overlapping of two range images with different point density because it uses absolute constants to measure whether a capture has many or few points in a cell. It would be necessary to include variables related to the size of the capture to resolve this problem.

Finally, it is possible to replace the genetic algorithms with binary coding used with genetic algorithms with real coding, which can provide a greater convergence speed due to the variety of genetic operators available for these.

Acknowledgement

This project has been partially financed by the Ministerio de Industria, Turismo y Comercio Español through the program AVANZA (TSI-020110-2009-438).

Bibliography

1. Blais, G.; Martin D. L.: Registering Multiview Range Data to Create 3D Computer Objects, Centre for Intelligent Machines, TR-CIM-93-16, Center for Intelligent Machines, McGill University, 1993.
2. Chen H.; Flann, N. S.: Paralled Simulated Annealing and Genetic Algorithm: A Space of Hybrid Methods. *Paralled Problem Solving from Nature*, Vol. 3, pp. 428-438 (1994).
3. Chen, C.; Hung, Y.; Cheng, J.: RANSAC-based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1229 –1234 (1999).
4. Chen, H.; Flann, N.S.; Watson, D.W.: Parallel genetic simulated annealing: a massively parallel SIMD algorithm. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 2, pp. 126-136 (1998).
5. Davis, L. D.; Mitchell M.: *Handbook of genetic algorithms*. Van Nostrand Reinhold (1991).

6. Holland, J. H.: Adaptation in Natural and Artificial Systems. *University of Michigan Press, Ann Arbor* (1975).
7. Horn, B. K.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, Vol. 4, No. 4, pp. 629-642 (1987).
8. Kohonen, T.: Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, Vol. 43, pp. 59–69 (1982).
9. Lee, L.; Wang, L.; Chen, S.: Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques. *Expert Systems with Applications*, Vol. 34, No. 1, pp. 328-336 (2008).
10. Pulli, K.: Multiview Registration for Large Data Sets, Stanford University. Proceedings. *Second International Conference on 3-D Digital Imaging and Modeling*, pp. 160-168 (1999).
11. Sappa, A. D.; Spetch, A. R.; Devi, M.: Range Image Registration by using an Edge-Based Representation. *In Proceedings of th 9th International Symposium on Intelligent Robotic Systems* (2001).
12. Stein, F.; Medioni, G.: Structural Indexing: Efficient 3-D Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2 (1992).
13. Tavakolpour, A. R.; Mat Darus, I. Z.; Tokhi O., Mailah T.: Adaptive simulated annealing genetic algorithm for system identification. *Engineering Applications of Artificial Intelligence*, Vol. 9, No. 5, pp. 523-532 (1996).
14. Tsenov, A.: Simulated Annealing and Genetic Algorithm in Telecommunications Network. *Proceedings of the 7-th IEEE International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services - TELSIKS*, Vol. 2, pp. 555 - 558 (2005).
15. Wong, K. P.; Wong, Y.W.: Genetic and genetic/simulated-annealing approaches to economic dispatch. *IEEE Proceedings Generation, Transmission and Distribution*, Vol. 141, No. 5, pp.507-513 (1994).

Authors' Information

Fernando Ortega, Javier San Juan, Francisco Serradilla, Dionisio Cortes

Dpto. de Sistemas Inteligentes Aplicados, Universidad Politécnica de Madrid, Crta. de Valencia, Km 7. 28031 Madrid, Spain

{fortegarequena, javier.sanjuan, francisco.serradilla, dio.cortes.fernandez}@gmail.com

NUMERICAL INTEGRATION BY GENETIC ALGORITHMS

Vladimir Morozenko, Irina Pleshkova

***Abstract**: It is shown that genetic algorithms can be used successfully in problems of definite integral calculation especially when an integrand has a primitive which can't be expressed analytically through elementary functions. A testing of the program, which uses the genetic algorithm developed by authors, showed that the best results are reached if the size of population makes 30-50 chromosomes, approximately 40-60% of its take a part in crossover, and the program stops if the population's leader didn't change during 5-10 generations. An answer of genetic algorithm is more exact than answer received by the classical numerical methods, even if a quantity of partition's points into segment is small or if an integrand is quickly oscillating. So genetic algorithms can compete both on the accuracy of calculations and on operating time with well-known classical numerical methods such as midpoint approximation, top-left corner approximation, top-right corner approximation, trapezoidal rule, Simpson's rule.*

***Keywords:** definite integral, integral sum, numerical integration, genetic algorithm, fitness-function.*

***ACM Classification Keywords**: F.1.2 COMPUTATION BY ABSTRACT DEVICES: Models of computation – Probabilistic computation. G.1.6 NUMERICAL ANALYSIS: Optimization – Stochastic programming.*

Introduction

A solution of many problems in physics, chemistry, mechanics and other natural sciences requires calculation definite integrals. Unfortunately, an exact analytic calculation of definite integrals is often impossible. Many functions do not have primitive that can be expressed analytically through elementary functions. For example, this is true for the function $f(x) = \exp(-x^2)$. Moreover, a symbolic integration is a much more difficult problem than a finding the primitive and there is no universal algorithm solving this problem. That's why an exact calculation of definite integrals by the fundamental theorem of calculus is often difficult or impossible at all.

Traditionally, many algorithms for calculating of the integral's value are used in numerical analysis. In most of them, the integrand $f(x)$ is replaced with the approximating function $\varphi(x)$ which is easier to integrate [Samarsky, 1989].

In this paper a new method of numerical integration is described. This method doesn't require knowledge of integrand's primitive because it is based on a genetic algorithm.

Such an unusual method of numerical integration expands area of applicability of genetic algorithms, which are traditionally used for solving of optimization problems [Gladkov, 2009].

Theoretical premises

Let a function $f(x)$ be defined on segment $[a,b]$. It is required to calculate a definite integral I of the function $f(x)$ over the segment $[a,b]$:

$$I = \int_a^b f(x)dx. \tag{1}$$

Let $P = \{x_0, x_1, x_2, ..., x_k\}$ be a partition of the segment $[a,b]$. It is assumed that $a = x_0 < x_1 < ... < x_k = b$. The nodes $x_0, x_1, x_2, ..., x_k$ of partition P subdivide the segment $[a,b]$ into k small segments $[x_0, x_1], [x_1, x_2], ..., [x_{k-1}, x_k]$ so that $x_i - x_{i-1} = (b-a)/k$ for all $i = \overline{1,k}$.

According to the additive property of definite integrals we have the equality

$$\int_a^b f(x)dx = \sum_{i=1}^{k} \int_{x_{i-1}}^{x_i} f(x)dx.$$

The theorem of mean value integration states that into segment $[x_{i-1}, x_i]$ there exists a point ξ_i for which the following equality is true [Shipachev, 2005]:

$$\int_{x_{i-1}}^{x_i} f(x)dx = f(\xi_i) \cdot (x_i - x_{i-1}),\ x_{i-1} \le \xi_i \le x_i.$$

Let us pick some point c_i into each segment $[x_{i-1}, x_i]$ and then define the following integral sum $S(c_1, c_2, ..., c_k)$:

$$S(c_1, c_2, ..., c_k) = \sum_{i=1}^{k} f(c_i) \cdot (x_i - x_{i-1}). \tag{2}$$

Geometric interpretation of this integral sum (2) is shown on Fig. 1.

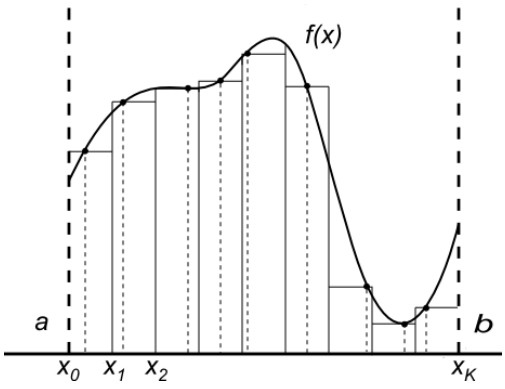


Figure 1. Geometric interpretation of the integral sum S with random points c_i

The integral sum $S(c_1, c_2, ..., c_k)$ is random variable, because points c_i into each segment $[x_{i-1}, x_i]$ are selected by random way. However it approximately equals to the value of the definite integral (1) and the calculation error does not exceed the error given by the rectangle method. Moreover the mathematical expectation $M[S]$ of integral sum's value must be equal to the exact value of the integral (1), i.e.

$$M[S] = \int_a^b f(x)dx.$$

If another set of points c_i is selected, the value of the integral sum is different, but still it must be approximately equal to the integral's value. In case of large N arithmetic average value of integral sum S^* converges to the value of integral (1) when N tends to infinity:

$$\lim_{N\to\infty} S^* = \lim_{N\to\infty} (S_1 + S_2 + ... + S_N)/N = \int_a^b f(x)dx. \quad (3)$$

The closer value of the integral sum (2) to the average sum *S** (3) is, the closer it is to the exact value of the integral (1). Therefore if we select points fortunately, we can calculate integral (1) with high accuracy. So we reduced the problem of the integral's calculation to the optimization problem: to find the points $c_1, c_2, ..., c_k$ into segments $[x_0, x_1], [x_1, x_2], ..., [x_{k-1}, x_k]$ so that

$$\left| S(c_1, c_2, ..., c_k) - S^* \right| \to min.$$

Further we shall show that this minimization problem can be solved by a genetic algorithm with a high accuracy. For this purpose we

- worked out a genetic algorithm for calculating the numerical value of definite integral of arbitrary functions defined on an segment;
- developed a program that implements the genetic algorithm;
- tested and debugged the program, using our genetic algorithm.

Genetic algorithm's description

Before we develop the genetic algorithm we need to:

- define the space of search;
- select the method of coding the possible solutions;
- set rules of crossover and mutation;
- define the fitness-function.

The main requirement to the fitness-function is following: its minimum's point must be an exact solution of the minimization problem [Gladkov, 2009].

By means of our genetic algorithm we will find the set of points $c_1, c_2, ..., c_k$, which minimize the difference between the integral sum (2) and the integral value (1) as far as it is possible.

Coding of possible solutions

The solution is an ordered set of *k* points, so it can be encoded by a sequence of *k* numbers – the coordinates of these points (Fig. 2). It's important to note, that there is only one point c_i into segment $[x_{i-1}, x_i]$ for each $i = \overline{1,k}$.

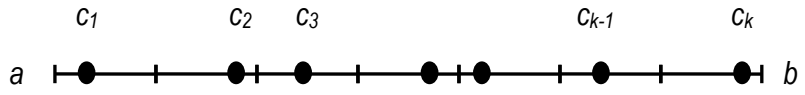


Figure 2. The encoding of the solution

So each chromosome is a sequence of *k* numbers $c_1, c_2, ..., c_k$, where *k* is quantity of small segments into the segment $[a,b]$.

Crossover and mutation

We choose the classical crossover technique – the one-point crossover. The crossover point is a random point from the partition $P = \{x_0, x_1, x_2, ..., x_k\}$ of the segment $[a,b]$, where $x_i - x_{i-1} = (b-a)/k$ for all $i = \overline{1,k}$.

Since all the segments $[x_0, x_1], [x_1, x_2], ..., [x_{k-1}, x_k]$ have the same length, all children after the one-point crossover are viable undoubtedly, because it is impossible that there would be no point or would be more than one point into segment $[x_{i-1}, x_i]$. An example of crossover's result is shown on Fig. 3 (the vertical line cuts chromosomes).

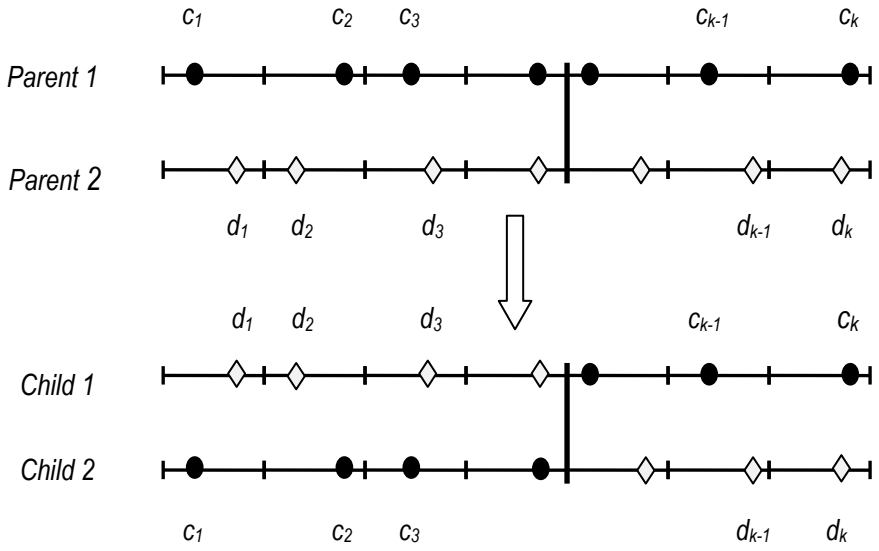


Figure 3. One-point crossover's result

Let p_{cross} be a quantity of percents of the best chromosomes in the generation, which participate in crossover (the parameter p_{cross} can be adjusted). The crossover's point is random.

Let p_{mut} be a quantity of percents of the chromosomes, which mutate (the parameter p_{mut} can be adjusted, usually it does not exceed to 5%). The mutation's operator randomly changes the coordinate of the point c_i into segment $[x_{i-1}, x_i]$. An example of the mutation is shown on Fig. 4 (point c_3 is replaced by point d_3).

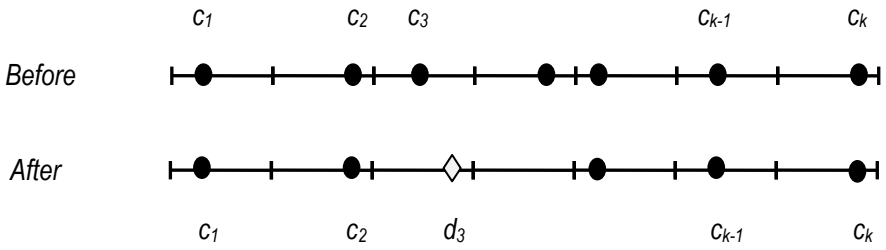


Figure 4. Mutation's result

After the mutation operator is applied to a chromosome, the resulting chromosome always is viable, because the point c_i moves within the segment $[x_{i-1}, x_i]$.

Fitness-function

Fitness-function has to meet the following requirements: the exact decision has to settle down in a point of its global minimum and value of function has to reflect a level of fitness of a chromosome [Gladkov, 2009].

For definition of fitness-function we will calculate the integrated sum for each chromosome in population on a formula (2). The problem of optimization is to minimize a divergence between the integrated sum and the value of

integral (1). As it was noted earlier, if the value of the integrated sum S is close to average value S^* of all integrated sums, then it is closer to value of integral too.

Therefore for calculation of fitness-function F we will use a formula:

$$F(c_1, c_2, ..., c_k) = \left| S(c_1, c_2, ..., c_k) - S^* \right|, \tag{4}$$

where S^* is the average value of all integrated sums in a given generation. It is obvious that the most adapted chromosomes are those who have the smallest value of fitness-function (4).

The algorithm stops when the leading chromosome doesn't change during several generations.

Testing of genetic algorithm

During a testing the quality and the operating time of genetic algorithm were investigated, their dependence on number of points in partition of a segment and parameters of genetic algorithm (such as size of the population, percent of the chromosomes chosen for crossing, a stop condition, etc.) was estimated.

To estimate the quality of the received decision in case of $I \neq 0$ the relative error of result is calculated by formula:

$$\delta = \left| \frac{I^* - I}{I} \right| \cdot 100\%, \tag{5}$$

where I is exact value of integral (1), I^* is answer received by means of genetic algorithm. If the integral (1) can't be calculated precisely, then I is received by means of some classical numerical method (such as midpoint approximation, Simpson's rule etc.). During the testing only one of parameters of the genetic algorithm was changed, and all the others parameters remained fixed.

Dependence of quality and operating time of genetic algorithm on partition's size

It is obvious that if number k of points into segment $[a,b]$ grows, the value $\Delta = (b-a)/k$ decreases. Therefore it is possible to calculate the integral (1) more precisely if we increase the number k. Suitable value of k depends on properties of integrand $f(x)$.

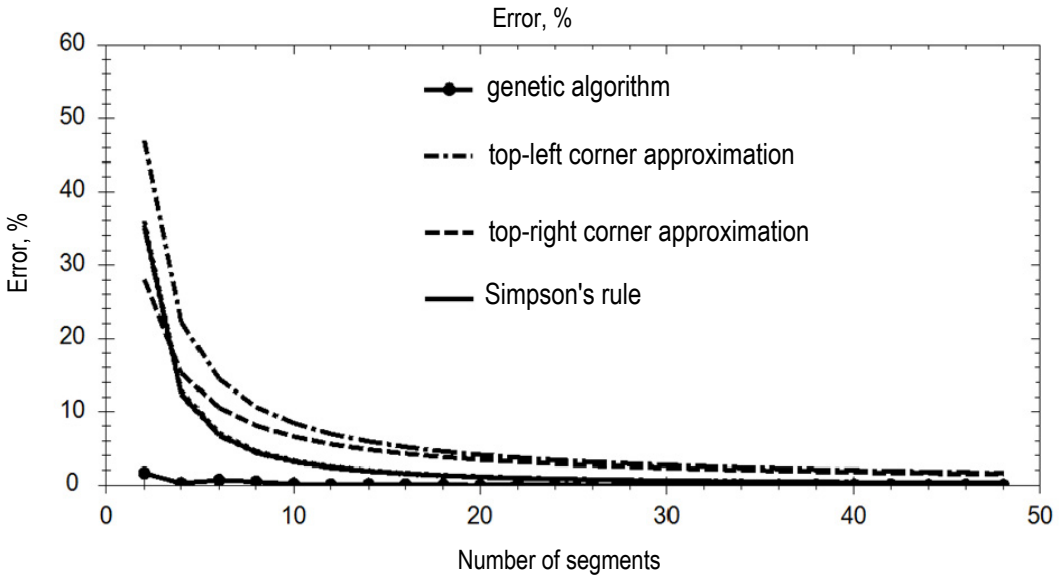


Figure 5. A result's precision given by genetic algorithm and by numerical methods

For example we consider an integral with a monotonic continuous integrand such as $\int_0^1 \sqrt{x}dx = 2/3$. The dependence of result's precision (5) given by genetic algorithm and by numerical methods on quantity of partition's points is presented on Fig. 5. It is important to note that the result's precision (5) given by genetic algorithm is much higher than ones given by known numerical methods such as top-left corner approximation, top-right corner approximation, Simpson's rule, especially in case of small partition's points.

The essential error arises in case of quickly oscillating functions if the integral is calculated by means of classical numerical methods. For example, we mean the integral $\int_{0.01}^{1} sin(1/x)dx$. The diagram of quickly oscillating integrand $sin(1/x)$ is shown on Fig. 6.

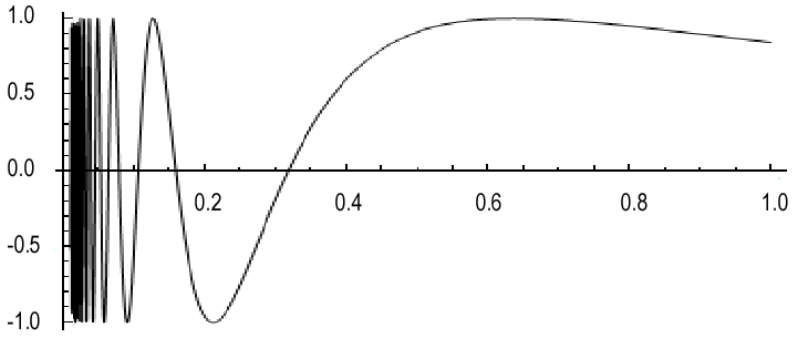


Figure 6. Diagram of quickly oscillating function sin(1/x)

But even in case of quickly oscillating function the genetic algorithm gives more exact answer, than other known numerical methods such as midpoint approximation, top-left corner approximation, top-right corner approximation, trapezoidal rule, Simpson's rule (Fig. 7).

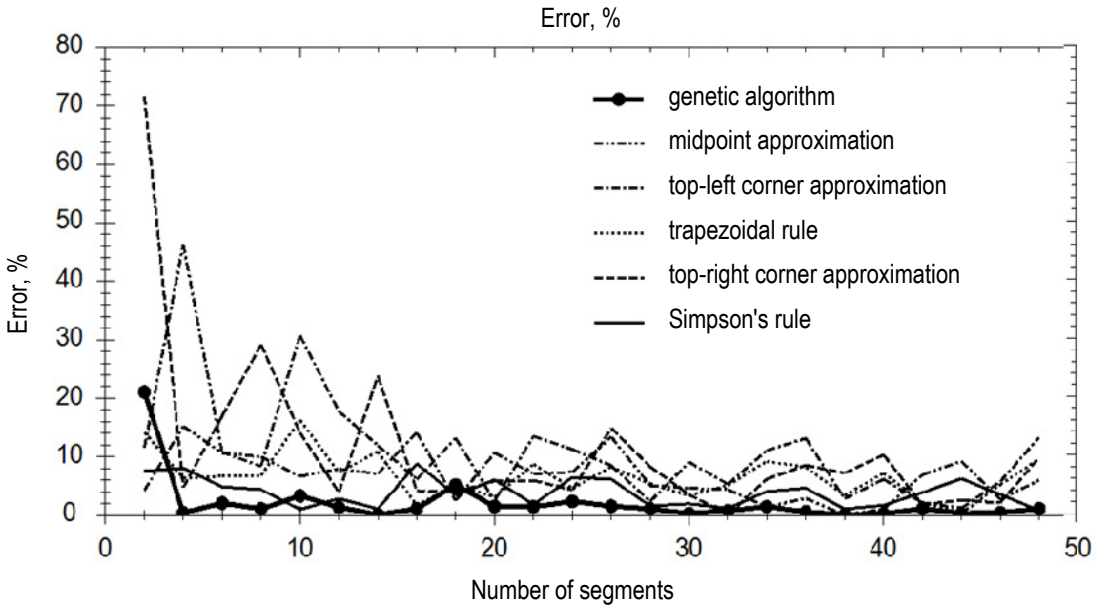


Figure 7. Precision given by genetic algorithm and by numerical methods in case of a quickly oscillating integrand

If the quantity *k* of partition's segments increases the operating time of the genetic algorithm increases too of course. For example, if the number of partition's segments is equal to 1000 the operating time of the genetic

algorithm doesn't exceed 1 minute. But if the number of partition's segments is less than 100, an operating time doesn't exceed 4 seconds.

Dependence of a quality and an operating time of genetic algorithm on its parameters

As a result of testing of the program using the genetic algorithm the following regularities were found:

- In case of small quantity of chromosomes in population (less than 15) the error of the decision is rather great. If to increase number of chromosomes in population then the average error (5) monotonically decreases and stabilizes.
- The percent of the chromosomes who are taking a part in crossover slightly influences an operating time of algorithm and doesn't influence almost response accuracy.
- The algorithm stops if the leader doesn't change during several generations. Such quantity of generations is one of the algorithm's parameter. A user can set its value voluntarily. If this parameter's value is small (less than 5), then an error of received decision (5) can be big. However in case of this parameter's value is more than 10 an error considerably decreases, and in case of its further increase the error practically doesn't change.

Calculating multiple integrals

The developed genetic algorithm can be extended for calculating multiple integrals. The task is to compute an integral where integrand depends on *n* arguments and exists into a domain *D*, where $D \subset R^n$:

$$\int_D f(x_1, x_2, ..., x_n)dx_1dx_2...dx_n, \tag{6}$$

The domain of the function $f(x_1, x_2, ..., x_n)$ is a set *D* which satisfies the following requirements:

- *D* is bounded in R^n, i.e. $\exists I^n : D \subseteq I^n$, where I^n – *n*-dimensional parallelepiped;
- the bound of *D* is a null-set in Lebesgue measure.

At the beginning we consider a simple case when the domain *D* is a *n*-dimensional parallelepiped. For clarity, we describe a genetic algorithm when integrand has two arguments and *D* is a rectangle.

Integration over a rectangle

The domain *D* of the function *f*(*x*,*y*) is a rectangle which is separated into equal rectangles by lines $x = x_i, i = \overline{0,k_x}, y = y_j, j = \overline{0,k_y}$ Then we pick a point c_{ij} in each rectangle (Fig. 8).

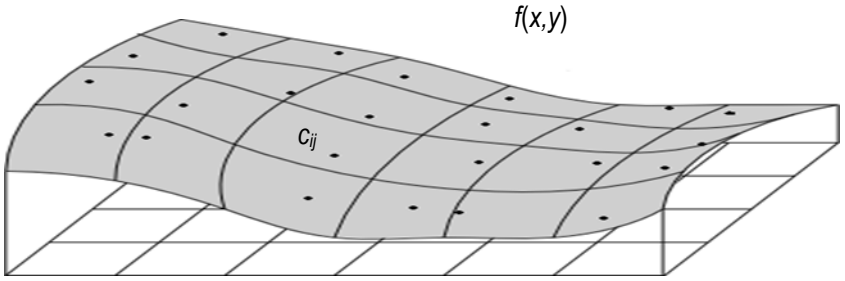


Figure 8. Diagram of function f(x,y), the partition of the rectangle and chosen points c_{ij}

In case of n = 2 a chromosome is a two-dimensional array and selected points $\left\{c_{ij} = (x_i, y_j) \,\middle|\, i = \overline{0,k},\ j = \overline{0,m}\right\}$ are elements of this array. In this case we need to select 2 crossover lines for

the crossover operator. But when D is n-dimensional parallelepiped we need to select n hyperplanes. Children inherit fragments of parents' chromosomes. The example of crossover is shown on Fig. 9.

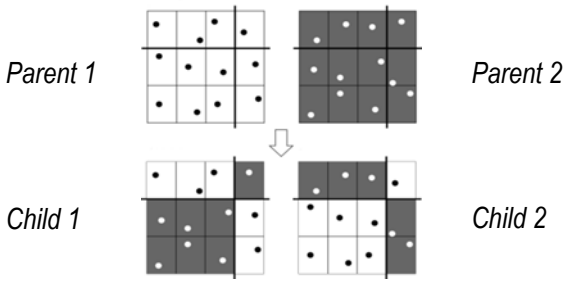


Figure 9. Crossover operator

During the mutation a point randomly changes its position inside the rectangle (Fig. 10).

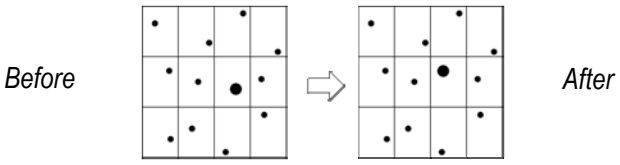


Figure 10. Mutation operator

Fitness-function F is computed by the formula (4), where S^* is the average value of the integral sums in the generation and S is integral sum for the chromosome calculated with following formula:

$$S = \sum_{i=1}^{k} \sum_{j=1}^{m} f(c_{ij}) \cdot (x_i - x_{i-1}) \cdot (y_j - y_{j-1}) \cdot$$

Integration over an arbitrary domain

For calculating integrals (6) over more complicated areas D, we introduce the concept of characteristic function. Let *characteristic function* for the set D be the function

$$\chi_D(x) = \begin{cases} 1, & x \in D, \\ 0, & x \notin D. \end{cases}.$$

The integral (6) of the function $f(x)$ over the domain D is defined as:

$$\int_D f(x_1, x_2, ..., x_n) dx_1 dx_2 ... dx_n = \int_{I^n} f(x_1, x_2, ..., x_n) \cdot \chi_D(x_1, x_2, ..., x_n) dx_1 dx_2 ... dx_n \cdot$$

where $D \subseteq I^n$ and I^n – n-dimensional parallelepiped.

Thereby, the task of integration over an arbitrary domain D is reduced to the integration over a parallelepiped I^n.

We tested our genetic algorithm with the purpose to evaluate its accuracy and an operating time. The testing procedure is similar to the one-dimensional case above. One of the parameters is changing while the others are fixed.

First, we investigated the dependence of the solutions' precision on number of the segments in the partitions. These solutions were obtained by genetic algorithm and classical numerical methods. There are two examples of the test below. In the first test, the integrand is smooth continuous function which is defined on a rectangle:

$$\int_0^1 dy \int_0^1 (sin(3x) + sin(5y))dx = \frac{2}{3} sin\left(\frac{3}{2}\right)^2 + \frac{2}{5} sin\left(\frac{5}{2}\right)^2 \approx 0.8066$$

The genetic algorithm shows more accurate result than other methods, even if the number of segments is small. The dependence of the solutions' precision on number of the segments in the partitions is presented on Fig. 11.

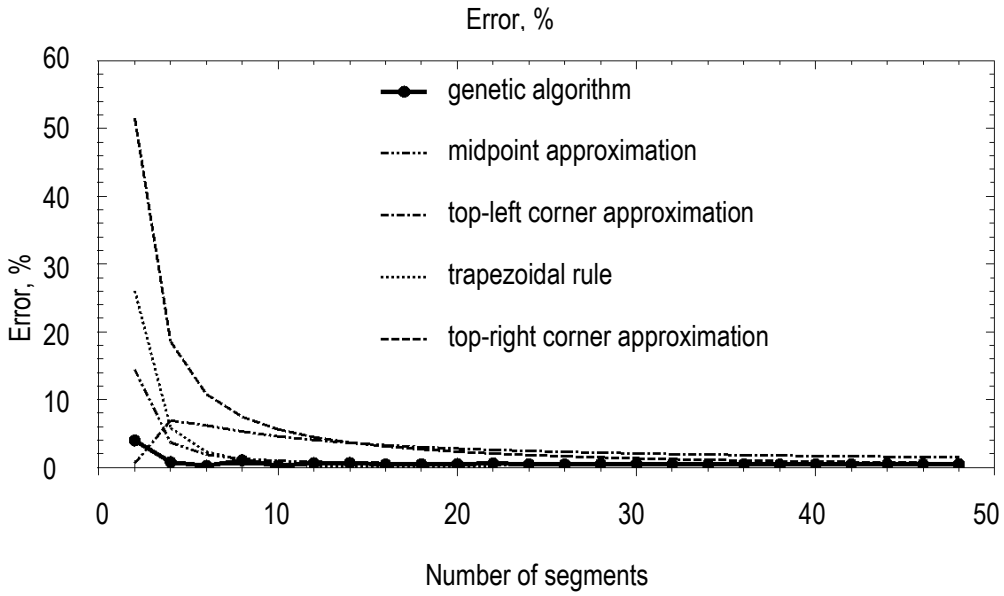


Figure 11. Results given by the genetic algorithm and numerical methods

The second example shows us a dependence of precision on the number of segments for quickly oscillating integrand.

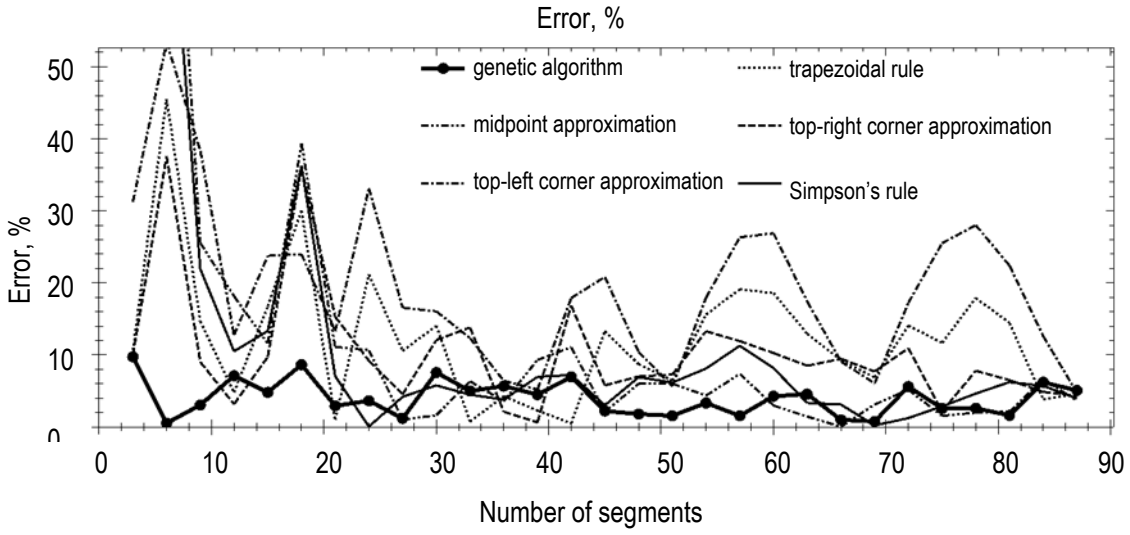


Figure 12. Results given by the genetic algorithm and numerical methods in case of quickly oscillating integrand

In this example we calculated the integral with quickly oscillating integrand

$$\int_{0.1}^{1} dx \int_{0.1}^{\sqrt{1-x^4}} \sin\left(\frac{1}{xy}\right) dy \approx 0.023 \cdot$$

The error increased when a quickly oscillating function is integrated, but the result given by genetic algorithm is more precise even than one given by Simpson's rule, not to mention other more rough approximations (Fig. 12).

Some more examples in the table below are shown:

Integral	**Accuracy, %**	**Operating time, sec.**	**Quantity of generations**
$\int_{\frac{3\pi}{2}}^{2\pi} d\varphi \int_{-\frac{\pi}{2}}^{0} \cos\psi\, d\psi \int_{0}^{2} r^2 (2r\cos\varphi\cos\psi + 1) dr$	0,0088	129,64	142
$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} d\varphi \int_{0}^{2\cos\varphi} r\sqrt{1+\cos(\varphi+\sin\varphi)}\, dr$	0,0783	19,34	180
$\int_{0}^{2\pi} d\varphi \int_{0}^{1} r dr \int_{2r^2}^{\sqrt{4-r^2}} z dz$	0,1125	130,53	123
$\int_{-6}^{2} dx \int_{\frac{x}{4}-1}^{2-x} \frac{3-\sin(6x+2y)}{\ln(x+10)} dy$	0,3369	12,02	109
$\int_{0}^{2} 5dx \int_{0}^{\sqrt{4-x^2}} y\, dy \int_{0}^{\frac{x^2+y^2}{4}} dz$	0,4125	229,67	145
$\int_{0}^{4} dx \int_{\frac{3}{4}x}^{\sqrt{25-x^2}} (x+\sin y)\, dy$	0,5254	4,08	36
$\int_{0}^{2\pi} d\varphi \int_{0.00001+\sqrt{11}}^{\sqrt{12}} r\sqrt{1+\frac{r^2}{r^2-11}}\, dr$	1,2788	8,29	81

Conclusion

A purpose of our investigation was to research a possibility of genetic algorithms' application to a task of definite integral's computation. To do this we developed a genetic algorithm and created the software product using this algorithm.

The developed genetic algorithm allows calculating definite integrals with an acceptable accuracy. Testing of a software product showed that the best accuracy of the decision is reached if the size of population makes from 30 to 50 chromosomes, 40-60% of chromosomes participate in crossover and the algorithm stops if the leader of population doesn't change during 5-10 generations.

The genetic algorithm with the specified parameters provides the more exact result than if we would apply other well-known numerical methods such as midpoint approximation, top-left corner approximation, top-right corner

approximation, trapezoidal rule, Simpson's rule. Advantage of the genetic algorithm is especially noticeable, when a quantity of partition's points is small and also when an integrand is quickly oscillating.

Executed research shows that genetic algorithms can be used for numerical integration when integrand has a primitive which can't be expressed analytically through elementary functions. Also developed genetic algorithms allow to calculate multiple integrals with integrand function of *n* arguments defined over *n*-dimensional parallelepiped or arbitrary bounded domain into *n*-dimensional space. Thus was confirmed that genetic algorithms can successfully compete with classical numerical methods both on the accuracy of computation and on an operating time.

Acknowledgement

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

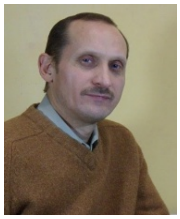
Bibliography

[Гладков и др., 2009] Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов П.В. Биоинспирированные методы в отпимизации. – М: ФИЗМАТЛИТ, 2009.

[Самарский, 1989] Самарский А.А., Гулин А.В. Численные методы. – М: Наука, 1989.

[Шипачев, 2005] Шипачев В.С. Высшая математика / Учебник для вузов. – М: Высш.шк. 2005.

Authors' Information



Vladimir Morozenko *– National Research University Higher School of Economics, Department of Information Technologies in Business, Associate Professor, Perm, Russia, 614070, Studencheskaya,38; e-mail: v.morozenko@mail.ru.*

Major Fields of Scientific Research: Genetic algorithms, Combinatorial algorithms, Optimization tasks, Cryptology, Cryptography

Irina Pleshkova *– Perm State National Research University, Department of Software and Computing Systems Mathematical Support, University undergraduate, Perm, Russia, 614990, Bukireva, 15; e-mail: elf_irina@mail.ru.*

Major Fields of Scientific Research: Genetic algorithms, Optimization tasks

INTRODUCTION TO STORING GRAPHS BY NL-ADDRESSING

Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko

***Abstract**: This paper introduces an approach to storing graphs based on the Natural Language Addressing in multidimensional numbered information spaces. A sample graph is analyzed to find its proper NL-representation. Taking in account the interrelations between nodes and edges, a "multi-layer" representation is possible and identifiers of nodes and edges can be avoided. The advantages and disadvantages of NL-addressing for the multi-layer representation of graphs are discussed.*

***Keywords**: graphs; addressing; natural language addressing.*

***ACM Classification Keywords**: D.4.3 File Systems Management, Access methods.*

Introduction

This paper introduces an original approach to storing graphs based on the so called "Natural Language Addressing" (NL-addressing) in multidimensional numbered information spaces [Ivanova et al, 2012; Ivanova et al, 2013].

Firstly, our attention will be paid to addressing and naming (labeling) in graphs with regards to introducing the NL-addressing. A sample graph will be analyzed to find its proper NL-representation. Taking in account the interrelations between nodes and edges, we will see that a "multi-layer" representation is possible and the identifiers of nodes and edges can be avoided.

As result of the analysis of the examples, the advantages and disadvantages of NL-addressing for the multi-layer representation of graphs will be discussed. The conclusion is that if we will use the indexed files or relational data bases, the disadvantages may make the implementation impossible. A solution of this problem may be the using of NL-addressing which consists in assuming the internal computer codes of letters as co-ordinates in a multidimensional information space.

Addressing and naming (labeling) in graphs

Graph theory may be said to have its beginning in 1736 when EULER considered the (general case of the) Königsberg bridge problem:

> "Is there a walk crossing each of the seven bridges of Königsberg (now Kaliningrad, Russia) exactly once?" [Euler, 1736] (Figure 1)

What is interesting in the scheme on Figure 1 is that every bridge has two kinds of identification:

- Its own ***name***: Krämer Br. (Shopkeeper Br.), Schmiede Br. (Blacksmith Br.), Grüne Br. (Green Br.), Köttel Br. (Guts, Giblets Br.), Honig Br. (Honey Br.), Holz Br. (Wooden Br.), and Hohe Br. (High Br.);
- Its own ***address***: a, b, c, d, e, f, g.

This reflects in our further research in two directions – *naming* and *addressing*, respectively.

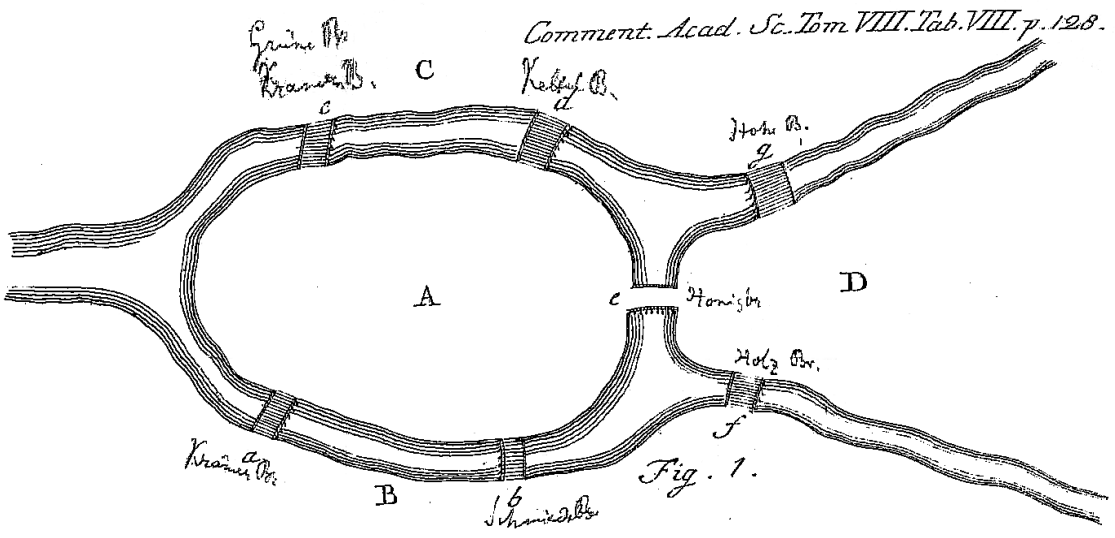


Figure 1. Illustration of Königsberg bridge problem [Euler, 1736]

As example of the ***addressing direction***, we may point the term "*addressable graph*" [Harju, 2011].

For instance, a computer network can be presented as a graph ***G***, where the vertices are the node computers, and the edges indicate the direct links. Each computer ***v*** has an address ***a(v)***, a bit string (of zeros and ones). The length of an address is the number of its bits. A message, that is sent to ***v***, is preceded by the address ***a(v)***. The Hamming distance ***h(a(v), a(u))*** of two addresses of the same length is the number of places, where ***a(v)*** and ***a(u)*** differ; e.g., h(00010, 01100) = 3 and h(10000, 00000) = 1.

It would be a good way to address the vertexes so that the Hamming distance of two vertices is the same as their distance in ***G***. In particular, if two vertices were adjacent, their addresses should differ by one symbol. This would make it easier for a node computer to forward a message.

A graph ***G*** is said to be addressable, if it has an addressing such that ***dG(u, v) = h(a(u), a(v))*** (Figure 2).

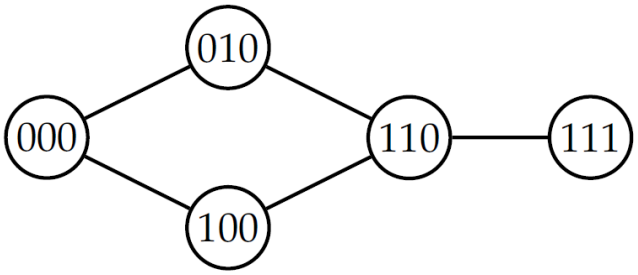


Figure 2. Example of an addressable graph

As example of the ***naming direction***, we may point the term "*named graphs*" [NG, 2013].

"Named Graphs" is the idea that having multiple RDF graphs in a single document/repository and naming them with URIs provides useful additional functionality built on top of the RDF Recommendations [RDF, 2013].

Let remember, the Resource Description Framework (RDF) is the W3C recommendation for semantic annotations in the Semantic Web. RDF is a standard syntax for Semantic Web annotations and languages [Klyne & Carroll, 2004]. The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an ***RDF graph***. This can be illustrated by a

node and directed-arc diagram, in which each triple is represented as a *node-arc-node* link (hence the term "graph") (Figure 3).

Figure 3. RDF triple

Each triple represents a statement of a relationship between the things denoted by the nodes that it links. Each triple has three parts: (1) subject, (2) object, and (3) a predicate (also called a property) that denotes a relationship. The direction of the arc is significant: it always points toward the object. The nodes of an RDF graph are its subjects and objects. The assertion of an RDF triple says that some relationship, indicated by the predicate, holds between the things denoted by subject and object of the triple. The assertion of an RDF graph amounts to asserting all the triples in it, so the meaning of an RDF graph is the conjunction (logical AND) of the statements corresponding to all the triples it contains. A formal account of the meaning of RDF graphs is given in [Hayes, 2004].

A *Named Graph* is an RDF graph which is assigned a name in the form of an URIref. The name of a graph may occur either in the graph itself, in other graphs, or not at all. Graphs may share URIrefs but not blank nodes. Named Graphs can be seen as a reformulation of quads in which the fourth element's distinct syntactic and semantic properties are clearly distinguished, and the relationship to RDF's triples, abstract syntax and semantics is clearer [Carroll et al, 2005].

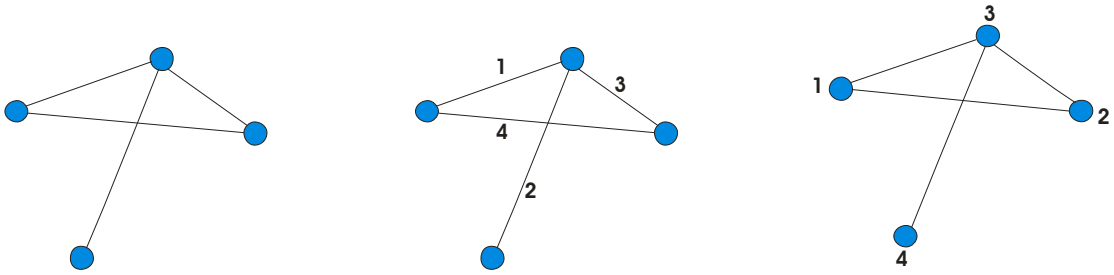
In other words, a Named Graph is a set of triples named by an URI. This URI can then be used outside or within the graph to refer to it.

Named Graphs aim at more complex RDF application areas like:

- Data syndication and lineage tracing;
- Ontology versioning;
- Modeling context;
- Modeling access control;
- Expressing privacy preferences;
- Scoping assertions.

Named graphs are kind of "*Labeled Graphs*". A graph labeling is an assignment of integers to the vertices or edges, or both, subject to certain conditions. Graph labeling was first introduced in the late 1960s. In the intervening years dozens of graph labeling techniques have been studied in over 1000 papers [Gallian, 2011].

A labeled graph ***G = (V, E)*** is a finite series of graph vertices ***V*** with a set of graph edges ***E*** of 2-subsets of ***V***. The term "labeled graph" when used without qualification means a graph with each node labeled differently (but arbitrarily), so that all nodes are considered distinct for purposes of enumeration [Weisstein, 2013] (Figure 4).



unlabeled graph *edge-labeled graph* *vertex-labeled graph*

Figure 4. Labeled graphs

NL-addressing in graphs

As it is seen from Figure 1, the set of addresses is isomorphic to set of names, i.e. the correspondence between two sets is one-one. This means that using of one or other of these sets closely depends of the interpreter and its functionality. If the interpreter is a computer or a mathematician, the addresses (numbers or letters) are preferable. If the interpreter is an end-user (human), the names (natural language words or phrases) are preferable.

Is it possible one and the same string of letters to be used as both name and address?

The positive answer is given by NL-addressing.

To illustrate NL-addressing in the multi-dimensional information spaces let see an example (Figure 5).

For instance, a separate *basic information element* (**BIE**) may be a letter, a word, a phrase of words (string). Such information element is colored in magenta on Figure 5 and has number 114.

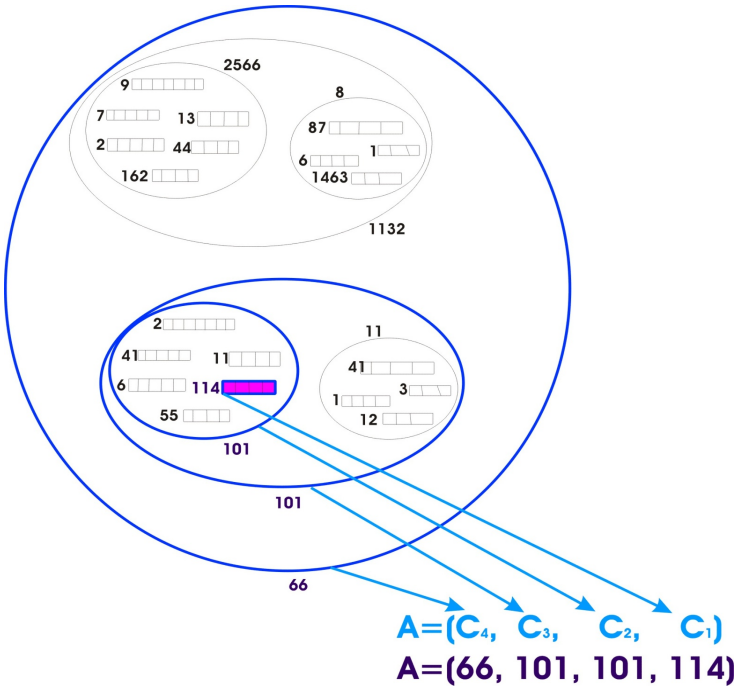


Figure 5. Example of a space address

Let a set of such elements is numbered by the human and stored in any archive (file). If we have several such sets, we may number them again and store in a common archive. And so on. This way we receive a specific hierarchy of numbered sets. If one will write the sequence of numbers of the sets starting from the one which contain all others, he will create a *space address*. The space address showed on Figure 5 means that the space with number 66 contains the space with number 101. The numbering is unique for every set. Because of this, there is no problem to have the same numbers in the included sets what is illustrated at the Figure 5 – set 101 contains element with number 101 which is a set of elements. Finally, the last set contains element with number 114 which is not a set but string of symbols.

In other words, the space address of this string is *A = (66, 101, 101, 114)* and its content may be written or read directly using this address.

Now we may to illustrate the idea of NL-addressing.

Consider the space address we just have seen – *(66, 101, 101, 114)*.

If we assume these numbers as ASCII codes, i.e. *66 = B, 101 = e, 114 = r,* we may "understand" the space address as the word "*Beer*" (Figure 6).

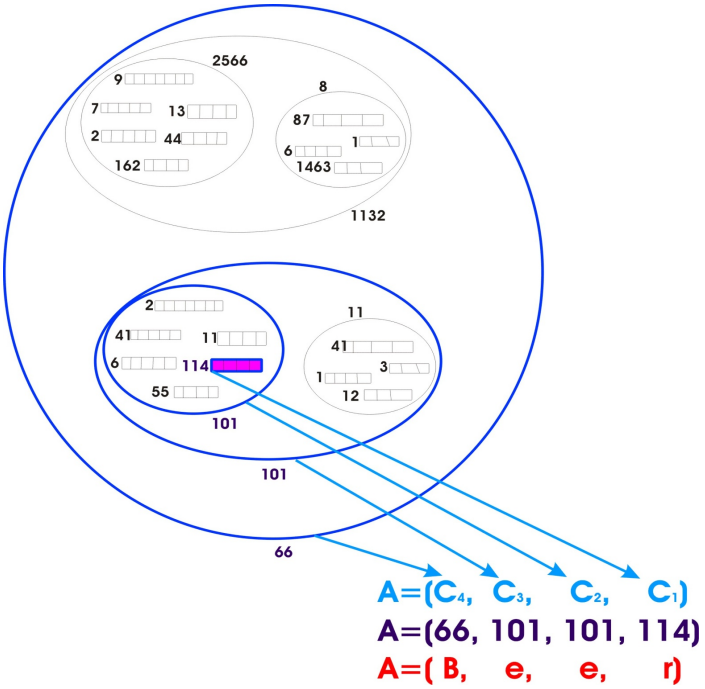


Figure6. Example of natural language space address

At the end, we have to illustrate the **BIE** (content) which may be stored at such NL-space address. It may be arbitrary long string of words. In our example we choose the **BIE** to be the remarkable aphorism of Benjamin Franklin: *"Beer is proof that God loves us and wants us to be happy"* (Figure 7).

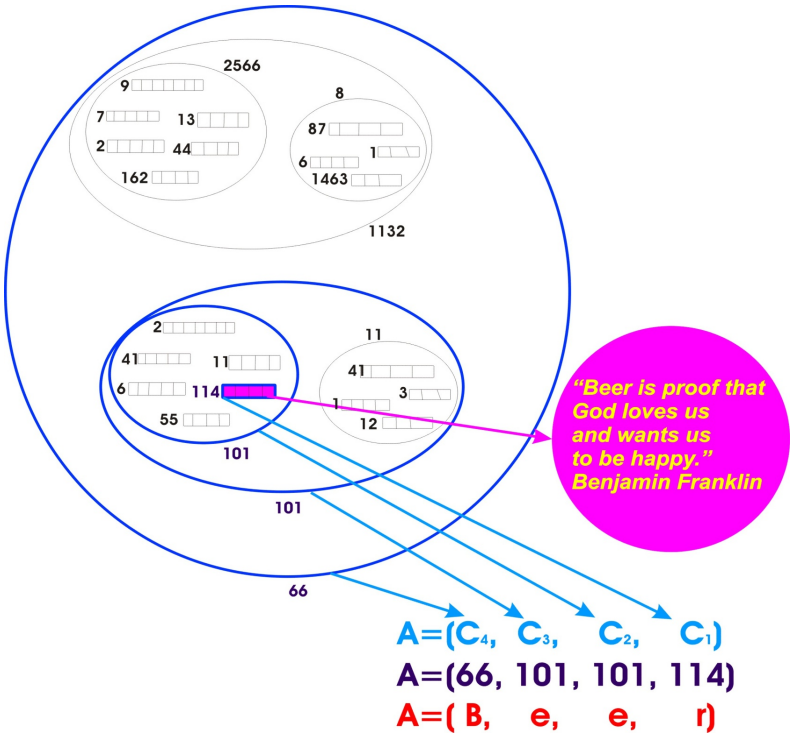


Figure 7. Example of content accessed by NL-addressing

In this case (Figure 7) the couple *{(space address A), (BIE)}* is:

{(B, e, e, r), ("Beer is proof that God loves us and wants us to be happy." Benjamin Franklin)}

To access the text, we have to convert index *(B, e, e, r)* to index *(66, 101, 101, 114)* and to use corresponded access operations, i.e. we have the consequence:

Beer =>
=> (B, e, e, r) =>
=> (66, 101, 101, 114) =>
=> ("Beer is proof that God loves us and wants us to be happy." Benjamin Franklin).

To illustrate using of NL-addressing, let see two simple examples.

- **Example 1. Multi-layer representation of a sample graph**

Consider a sample graph [GraphDB, 2012], which contains three named nodes *(Alice, Bob, Chess)* with addresses *(Id.1, Id.2 and Id.3)*, six labeled edges connecting them *(knows, knows, is_member, members, members, is_member)* with addresses *(Id.100, ..., Id.105)*, and some features and their values (Figure 5).

To represent this graph in the computer memory we have to use (identificators as) pointers. It is possible to present the information in two tables – one for nodes (Table 1) and another for the edges (Table 2). The names of columns and values of the corresponded cells are easy understandable.

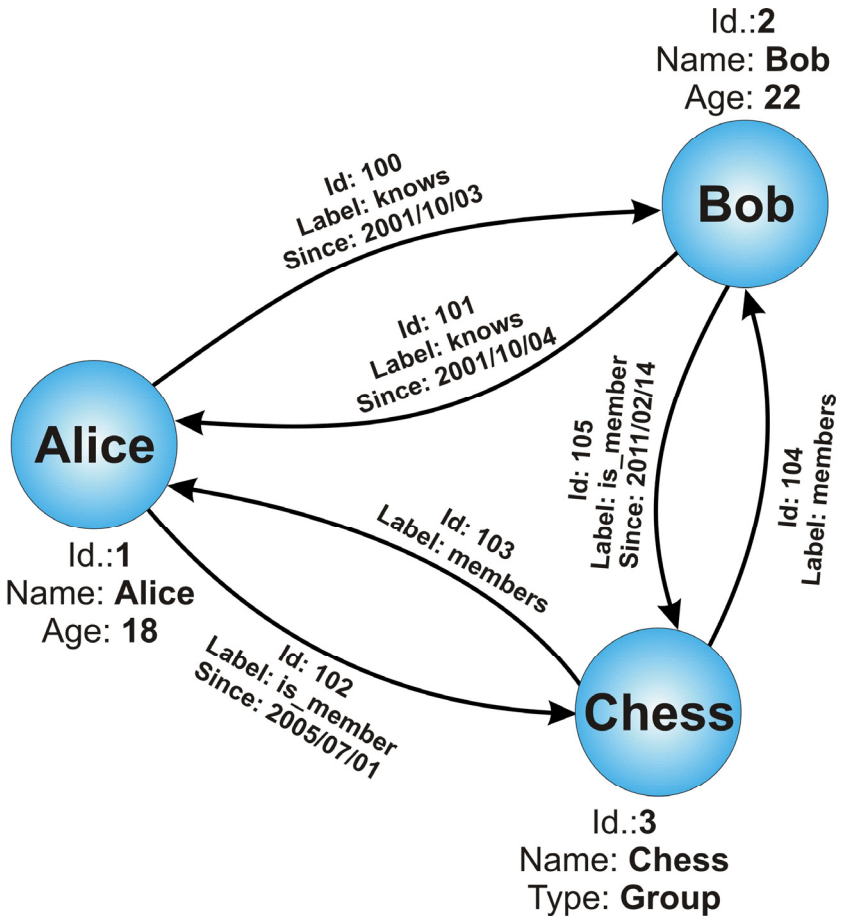


Figure 5. A sample named/addressed graph

Table 1. Description of nodes of the sample graph

Node Id.	Name	from-edges	to-edges	Age	Type
1	**Alice**	**101; 103**	**100; 102**	**18**	**-**
2	**Bob**	**100; 104**	**101; 105**	**22**	**-**
3	**Chess**	**102; 105**	**103; 104**	**-**	**Group**

Table 2. Description of the edges of the sample graph

Edge Id.	Label	from node	to node	Since
100	**knows**	**1**	**2**	**2001/10/03**
101	**knows**	**2**	**1**	**2001/10/04**
102	**is_member**	**1**	**3**	**2005/07/01**
103	**members**	**3**	**1**	**-**
104	**members**	**3**	**2**	**-**
105	**is_member**	**2**	**3**	**2011/02/14**

Each table corresponds to one type of components – to nodes or to the edges. Every row corresponds to one identificator (address) and connected to it features.

If we will take in account the interrelations between nodes and edges, we will see that another ("multi-layer") representation is possible and the identifiers of nodes and edges can be avoided (Table 3).

Table 3. Multi-layer representation of the sample graph.

		addresses		
		Alice	*Bob*	*Chess*
layers	*Age*	**18**	**22**	
	Type			**Group**
	knows; since	**Bob; 2001/10/03**	**Alice; 2001/10/04**	
	members			**Alice; Bob**
	is_member since	**Chess; 2005/07/01**	**Chess; 2011/02/14**	

In this case, names of nodes are addresses (i.e. names of columns) and names of edges will define the different "layers" (i.e. – rows of the table) in which the corresponded values are stored at the address (column) given by node names.

To find all edges from given node we have to start with node name (column), for instance "Bob", and read all information from different layers (rows) stored at address (in column) "Bob". In other words, all needed information for the node is in the column with corresponded node name.

If we will have possibility for NL-addressing it will reduce the information to be stored on the disc - only the cells with text in bold of Table 3 will be stored. At a rough estimate, the sum of filled cells in:

- The first case (Table 1 + Table 2) has at least 18 + 30 = 48 filled cells for the two tables and real need of additional indexing to speed up the access;
- In the second case (Table 3) – 8 filled cells and no need of additional indexing.

The graph database models

Let remember that the "*graph database model*" is a model in which the data structures for the schema and/or instances are modeled as a directed, possibly labeled, graph, or generalizations of the graph data structure, where data manipulation is expressed by graph-oriented operations and type constructors, and appropriate integrity constraints can be defined over the graph structure [Angles & Gutierrez, 2008].

Graph database model can be defined as those in which data structures for the schema and instances are modeled as graphs or generalizations of them, and data manipulation is expressed by graph-oriented operations and type constructors.

The notion of graph database model can be conceptualized with respect to three basic components, namely: (1) Data structures; (2) Transformation language; (3) Integrity constraints. Hence, a graph database model is characterized as follows:

- Data and/or the schema are represented by graphs, or by data structures generalizing the notion of graph (hypergraphs or hypernodes) [Guting, 1994; Levene & Loizou, 1995; Kuper & Vardi, 1984; Paredaens et al, 1995; Kunii, 1987; Graves et al, 1995a; Gyssens et al, 1990].
- Data manipulation is expressed by graph transformations, or by operations whose main primitives are on graph features like paths, neighborhoods, subgraphs, graph patterns, connectivity, and graph statistics (diameter, centrality, etc.) [Gyssens et al, 1990; Graves et al, 1995a; Guting, 1994];
- Integrity constraints enforce data consistency. These constraints can be grouped in schema-instance consistency, identity and referential integrity, and functional and inclusion dependencies. Examples of these are: labels with unique names, typing constraints on nodes, functional dependencies, domain and range of properties [Graves et al, 1995b; Kuper & Vardi, 1993; Klyne & Carroll, 2004; Levene & Poulovassilis, 1991].

Graph database models are applied in areas where information about data interconnectivity or topology is more important, or as important, as the data itself. In these applications, the data and relations among the data are usually at the same level. Introducing graphs as a modeling tool has several advantages for this type of data:

- It allows for a more natural modeling of data. Graph structures are visible to the user and they allow a natural way of handling applications data, for example, hypertext or geographic data. Graphs have the advantage of being able to keep all the information about an entity in a single node and showing related information by edges connected to it [Paredaens et al, 1995]. Graph objects (like paths and neighborhoods) may have first order citizenship; a user can define some part of the database explicitly as a graph structure [Guting, 1994], allowing encapsulation and context definition [Levene & Poulovassilis, 1990].
- Queries can refer directly to this graph structure. Associated with graphs are specific graph operations in the query language algebra, such as finding shortest paths, determining certain subgraphs, and so forth. Explicit graphs and graph operations allow users to express a query at a high level of abstraction. To some extent, this is the opposite of graph manipulation in deductive databases, where often, fairly complex rules need to be written [Guting, 1994]. It is not important to require full knowledge of the structure to express meaningful queries [Abiteboul et al, 1997]. Finally, for purposes of browsing it may be convenient to forget the schema [Buneman et al, 1996].
- For implementation, graph databases may provide special graph storage structures, and efficient graph algorithms for realizing specific operations [Guting, 1994]. These structures are very important – *the features of the graph storage structures influence the corresponded to them operations and algorithms*.

➢ **Example 2. The toy genealogy graph database**

Let's look at another example - the toy genealogy from [Angles & Gutierrez, 2008] (Figure 6.). The genealogy diagram (right-hand side) is represented as two tables (left-hand side) NAME-LASTNAME and PERSON-PARENT (Children inherit the last name of the father just for modeling purposes).

NAME	LASTNAME
George	Jones
Ana	Stone
Julia	Jones
James	Deville
David	Deville
Mary	Deville

PERSON	PARENT
Julia	George
Julia	Ana
David	James
David	Julia
Mary	James
Mary	Julia

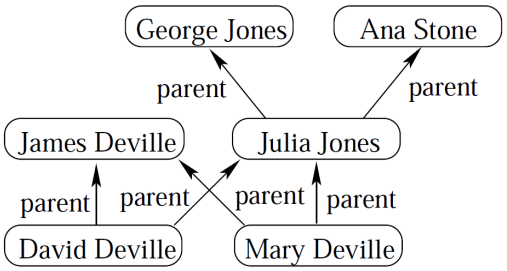


Figure 6. Running example: the toy genealogy graph database

The "multi-layer" representation of the family tree from Figure 6 is given in Table 5.

Table 5. Multi-layer representation of the family tree from Figure 6

		addresses					
		George	Ana	Julia	James	David	Mary
layers	lastname	***Jones***	***Stone***	***Jones***	***Deville***	***Deville***	***Deville***
	parent_of			***George; Ana***		***James; Julia***	***James; Julia***

The NL-addressing means direct access to content of each cell. Because of this, for NL-addressing the problem of recompiling the database after updates does not exist. In addition, the multi-layer representation and natural language addressing *reduce resources* (24 cells in Figure 6 vs. 9 cells in the case of Table 5) and *avoid using of supporting indexes for information retrieval services* (B-trees, hash tables, etc.).

Discussion

The state of the art with respect to existing storage and retrieval technologies for RDF graphs is given in [Hertel et al, 2009]. Different repositories are imaginable, e.g. main memory, files or databases. RDF schemas and instances can be efficiently accessed and manipulated in main memory. For persistent storage the data can be serialized to files, but, for large amounts of data, the use of a database management system is more reasonable. Examining currently existing RDF stores we found that they are using relational and object-relational database management systems. Storing RDF data in a relational database requires an appropriate table design. There are different approaches that can be classified in (1) generic schemas, i.e. schemas that do not depend on the ontology, and (2) ontology specific schemas.

Graph database models took off in the eighties and early nineties alongside object oriented models. Their influence gradually died out with the emergence of other database models, in particular geographical, spatial, semi-structured, and XML. Recently, the need to manage information with graph-like nature has reestablished the relevance of this area [Angles & Gutierrez, 2008].

The graph oriented approach for storing ontologies became one of the preferred. Some of the world's leading companies and products which support extra-large ontology bases are presented on page of W3C [LTS, 2012]. It should be noted, there exists a gradual transition from relational to non-relational models for organizing

ontological data. Perhaps the most telling example is the system AllegroGraph® 4.9 [AlegroGraph, 2012] of the FRANZ Inc.

Concluding the examples, let point on advantages and disadvantages of the proposed here multi-layer NL-representation of graphs.

The main advantages are:

- Reducing the number of tables, which represent the graph;
- Reducing the number of filled cells.

The main disadvantages are:

- The tables are sparse;
- Avoiding pointers we receive a variety of names, which have different lengths and cause difficulties for the implementations in the data bases where the fixed length is preferable;
- The number of nodes may be very great and this way needs corresponded number of columns in the table (in any cases hundreds or thousands).

Multi-domain information model (MDIM)

If we will use the indexed files or relational data bases, the disadvantages of such data models may make the implementation impossible. We propose another approach which is based on addressing with variable number of co-ordinates. For instance, the ASCII internal representation of the word "accession" is the following unique sequence of numbers: *(97, 99, 99, 101, 115, 115, 105, 111, 110)*. It is its NL-address (9 co-ordinates) and its internal computer representation. Using the same approach, the names of our sample graph will be represented as follow (Table 6):

Table 6. Names of the nodes of the sample graph and theirs co-ordinate representations

name	co-ordinates	comment
Alice:	(65, 108, 105, 99, 101)	5 co-ordinates (dimensions)
Bob:	(66, 111, 98)	3 co-ordinates (dimensions)
Chess:	(67, 104, 101, 115, 115)	5 co-ordinates (dimensions)

As it is seen from Table 6, words (and phrases) have different lengths and require using of addressing by co-ordinate arrays with variable length, i.e. to have variable dimensions in one and the same time. Such addressing we call "multi-dimensional".

Fortunately, there exist approach called "Multi-Domain Information Model" (MDIM) and corresponded to it "Multi-Domain Access Method" (MDAM) which permit operating with multi-dimensional addressing. Detailed and formal presentations of MDIM have been given in [Markov, 1984; Markov, 2004]. There exist several realizations of MDAM for different hardware and/or software platforms. The most resent one is the Archive Manager – ArM [Markov et al, 2008]. We have upgraded it for NL-addressing approach and applied for NL-storing of graphs. Below we will illustrate this model by short comments and several figures.

We consider the type of memory organization, which is based on the numbering as a main approach. The main idea consists in mapping the values of the objects' attributes (symbol or real; point or interval) to integer numbers of the elements of corresponding ordered sets. This way, each object will be described by a vector of integer values, which may be used as the co-ordinate address in the multi-dimensional information space. This type of memory organization is called "*Multi-dimensional numbered information spaces*".

The process of mapping the names to numbers permits the use of mathematical functions and address vectors for accessing the information instead of search engines.

As an example we will consider a numbered modification of Table 3 where the names of addresses and layers are replaced by integer numbers (Table 7).

Table 7. Numbered representation of the sample graph

		addresses		
		1	*2*	*3*
layers	*1*	**18**	**22**	
	2			**Group**
	3	**Bob; 2001/10/03**	**Alice; 2001/10/04**	
	4			**Alice; Bob**
	5	**Chess; 2005/07/01**	**Chess; 2011/02/14**	

Every row of the Table 7is a numbered information space of range 1 because all elements in the every row are numbered separately. In the example this is indicated by couples *(number : element)*. This way we have:

row 1: (1 : 18); (2 : 22); (3 : None).

row 2: (1 : None); (2 : None); (3 : Group).

row 3: (1 : Bob; 2001/10/03); (2 : Alice; 2001/10/04); (3 : None).

row 4: (1 : None); (2 : None); (3 : Alice; Bob).

row 5: (1 : Chess; 2005/07/01); (2 : Chess; 2011/02/14); (3 : None).

The set of rows of the Table 7 is numbered information space of range 2 because all rows are numbered and all elements in the every row are numbered, too. This is indicated by triple

(number of row, number of element : element).

This way we have a matrix below (called Example_table 1 to be distinguished from other tables in this paper):

Example_table 1:

(1, 1 : 18); (1, 2 : 22); (1, 3 : None).

(2, 1 : None); (2, 2 : None); (2, 3 : Group).

(3, 1 : Bob; 2001/10/03); (3, 2 : Alice; 2001/10/04); (3, 3 : None).

(4, 1 : None); (4, 2 : None); (4, 3 : Alice; Bob).

(5, 1 : Chess; 2005/07/01); (5, 2 : Chess; 2011/02/14); (5, 3 : None).

Space addresses of the elements of Example_table 1 above are the couples *(1, 1), (1, 2), ..., (5, 3)*. For instance, the couple *(3, 2)* is address of string "*Alice; 2001/10/04*". If we add leading zeroes the address remain the same, i.e. *(0, 0, 0, 3, 2) = (3, 2) => "Alice; 2001/10/04"*.

A space indexes i_1 and i_2 over Example_table 1 above may be the sequences:

i_1*: {(2, 3), (3, 2), (5, 1), (3, 3)}*

i_2*: {(1, 2), (1, 3), (1, 2), (1, 3)}*

The indexes may be not sorted (see i_1) and may contain repeated addresses (see i_2).

If we assume that the equally numbered elements of the rows of Example_table 1 above corresponds each other than we may build different "sub-tables" (*ST*) such as:

ST_1*: {row 3, row 4, row 2};*

ST_2*: {row 5, row 5, row 1}.*

ST_1 and ST_2 are *aggregates*. There are no restrictions on the rules for creating the aggregates.

The aggregate ST_1 contains the next table

ST_1:

(3, 1 : Bob; 2001/10/03); (3, 2 : Alice; 2001/10/04); (3, 3 : None).

(4, 1 : None); (4, 2 : None); (4, 3 : Alice; Bob).

(2, 1 : None); (2, 2 : None); (2, 3 : Group).

The aggregate ST_2 contains the next table

ST_2:

(5, 1 : Chess; 2005/07/01); (5, 2 : Chess; 2011/02/14); (5, 3 : None).

(5, 1 : Chess; 2005/07/01); (5, 2 : Chess; 2011/02/14); (5, 3 : None).

(1, 1 : 18); (1, 2 : 22); (1, 3 : None).

Finally, the Table 7 is an aggregate which is constructed by all layers (rows). We may represent it as an aggregate as follow:

- Every layer (row) will be separate information space of range 1 and will be stored in a separate file with name of the layer.
- The names Alice, Bob, and Chess will be space addresses common for all layers, i.e. the corresponded elements will be accessible via one and same addresses.

As result we will have the following structure of files (Table 8).

Table 8. Aggregate built by information from Table 3

file name	content
Age	(Alice - 18); (Bob - 22); (Chess - None).
Type	(Alice - None); (Bob - None); (Chess - Group).
knows_since	(Alice – Bob : 2001/10/03); (Bob – Alice : 2001/10/04); (Chess - None).
members	(Alice - None); (Bob - None); (Chess - Alice; Bob).
is_member	(Alice – Chess : 2005/07/01); (Bob – Chess : 2011/02/14); (Chess - None).

Taking in account that the Alice, Bob, and Chess are not real text strings but NL-addresses, i.e. *Alice = (65, 108, 105, 99, 101)*, *Bob = (66, 111, 98)*, and *Chess = (67, 104, 101, 115, 115)* the aggregate has the following structure (Table 9).

Table 9. Storing format of the aggregate from Table 8

file name	space addresses		
	(65, 108, 105, 99, 101)	(66, 111, 98)	(67, 104, 101, 115, 115)
Age	**18**	**22**	
Type			**Group**
knows_since	**Bob; 2001/10/03**	**Alice; 2001/10/04**	
members			**Alice; Bob**
is_member	**Chess; 2005/07/01**	**Chess; 2011/02/14**	

Only the strings in bold will be stored on the disk.

Assuming that names of the files are NL-addresses, too, we may build more complex structure (Table 10) in one file using concatenated space addresses *(name | file_name)* where NL-addresses are the elements of the sets:

name : {Alice, Bob, Chess}

file_name : {Age, Type, knows_since, members, is_member}.

Again, only strings in bold will be stored on disk.

Table 10. Concatenated NL-addresses of data from Table 3

(Alice|Age : **18**)	(Bob|Age : **22**)	(Chess|Age : None)
(Alice|Type : None)	(Bob|Type : None)	(Chess|Type : **Group**)
(Alice|knows_since : **Bob; 2001/10/03**)	(Bob|knows_since : **Alice; 2001/10/04**)	(Chess|knows_since : None)
(Alice|members : None)	(Bob|members : None)	(Chess|members : **Alice; Bob**)
(Alice|is_member : **Chess; 2005/07/01**)	(Bob|is_member : **Chess; 2011/02/14**)	(Chess|is_member : None)

Representing characteristics of the nodes and edges

A question about representing the characteristics of the nodes and edges rises from the analysis of the Table 9 and Table 10. At the Figure 5 they have been written as any comments to nodes and edges.

The ***characteristics of nodes*** (viz. age, type) may be represented as additional loop edges of type "*has_characteristics*" and different characteristics may be given by keywords and corresponded values for these edges.

The ***characteristics of edges*** (viz. since) may be represented as additional information to the node pointed by the corresponded edge. This information may be given again by corresponded keywords and theirs values.

For instance, the final table representation of our sample graph is given in Table 11 and corresponded to it representation by triples is given in Table 12.

The final version of the sample graph based on the information of Table 11 (or Table 12) is shown at Figure 9.

Table 11. Final variant of the aggregate from Table 8

file names	space addresses		
	Alice	*Bob*	*Chess*
has_characteristics	**Alice - Age: 18**	**Bob - Age: 22**	**Chess - Type: Group**
knows	**Bob - since : 2001/10/03**	**Alice - since: 2001/10/04**	
members			**Alice - since: 2005/07/01; Bob - since: 2011/02/14**
is_member	**Chess - since: 2005/07/01**	**Chess - since: 2011/02/14**	

Finally, using NL-addressing, the multi-layer representation is easily understandable by humans and interpretable by the computers. To illustrate this, let see the description by triples of the sample graph (Table 12).

Table 12. Representation of the sample graph by triples

Subject	Relation	Object
Alice	*has_characteristics*	**Alice – Age : 18**
Alice	*knows*	**Bob – since : 2001/10/03**
Alice	*is_member*	**Chess – since : 2005/07/01**
Bob	*has_characteristics*	**Bob – Age : 22**
Bob	*knows*	**Alice – since : 2001/10/04**
Bob	*is_member*	**Chess – since : 2011/02/14**
Chess	*has_characteristics*	**Chess –Type : Group**
Chess	*members*	**Alice; Bob**

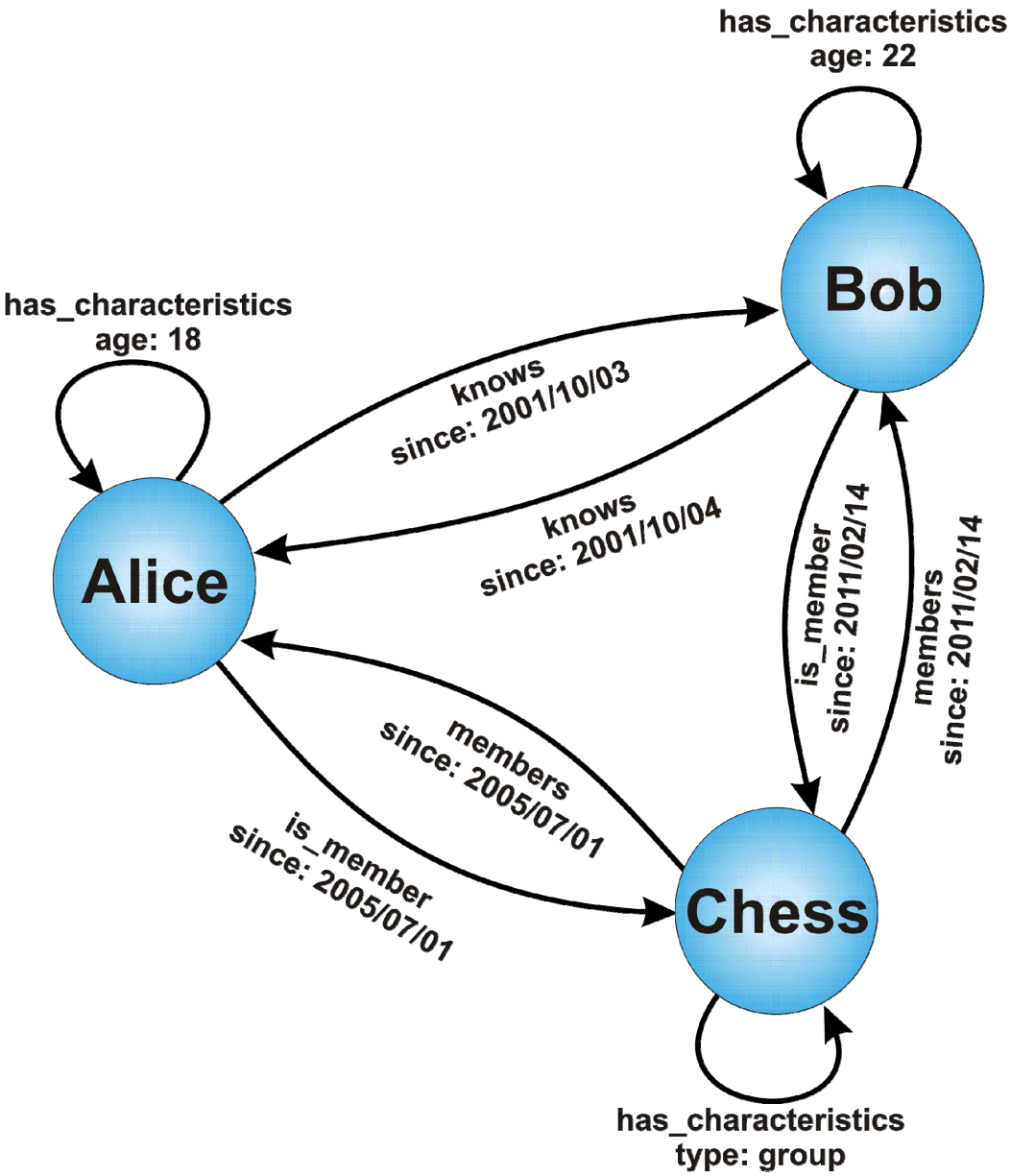


Figure 9. Final variant of the sample graph

OWL and RDF descriptions of the sample graph

To make comparison, corresponded OWL and RDF descriptions of the sample graph are given in Table 13 and Table 14 in the Appendix. They were prepared using Protégé 4.2.

Protégé is developed by the "Stanford Center for Biomedical Informatics Research" at the Stanford University School of Medicine. This is a tool which allows a user to construct domain ontology, customize data entry forms and enter data. The tool can be easily extended to access other knowledge based embedded applications. For example, Graphical widgets can be added for tables and diagrams. Protégé can also be used by other applications to access the data [protégé, 2012; protege-owl, 2012]. There is an additional option in Protégé, which serves the storing of ontologies in various relational databases, called OntoBase [Yabloko, 2011]. It should be noted that the same name "OntoBase" is used in [Pan & Pan, 2006], but without any connection to Protégé. Figure 10 illustrates the using of Protégé by graphical representation of our sample graph.

The example on Figure 10 and descriptions given in the Appendix show that the sentence:

"OWL and RDF are easy readable by humans"

is not the all truth.

Linearization the information is suitable solution for telecommunication and computer processing, but it is not easy understandable by humans. Let remember two dimensional representation of the sample graph. What is presented in four rows in Table 11 is the same as one presented on the pages with small font in the Appendix.

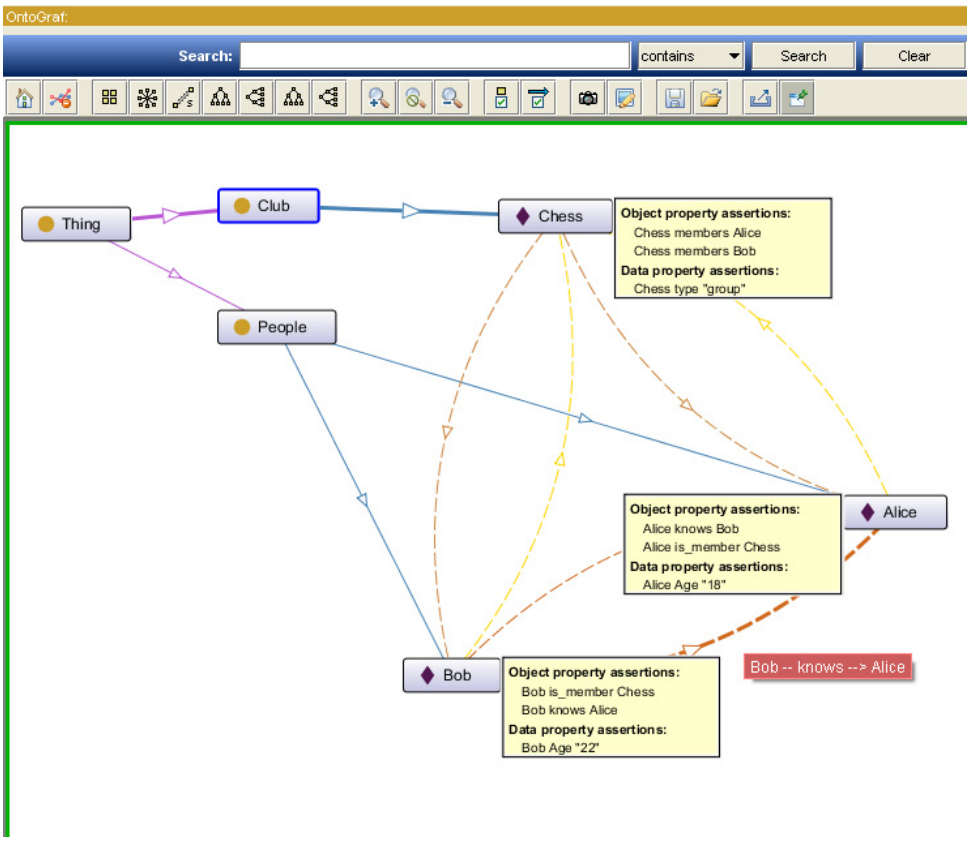


Figure 10. Protégé graphical representation of the sample graph

Conclusion

In January 1978, John F. Shoch from "Xerox Palo Alto Research Center" had written a very interesting note [Shoch, 1978a]. Later in the same year he had published this note in the paper [Shoch, 1978b]. This classical paper became as a mile stone in the further research concerning the naming, addressing and routing at the first place with its "extremely general definition" [Shoch, 1978a]:

The "name" of a resource indicates "what" we seek,

an "address" indicates "where" it is, and

a "route" tell us "how to get there".

This definition gives us a quick and intuitive understanding of the fundamental concepts of naming. Informally, a *name is a string of symbols that identifies an object*, thus both a human readable text-string and a binary number can be a name. Ideally, all objects would be named and handled in a uniform manner [Jording & Andreasen, 1994].

Shoch gave "some further detail to flesh this out" [Shoch, 1978a]:

I. A "***name***" is a symbol, usually a human-readable string, identifying some resource, or set of resources. The name (what we seek) needs to be bound to the address (where it is).

II. An "***address***", however, is the data structure whose format can be recognized by all elements in the domain, and which defines the fundamental addressable object. The address (where something is) needs to be bound to the route (how to get there).

III. A "***route***" is the specific information needed to forward a piece of information to its specified address.

Thus, a "name" may be used to derive an "address", which may then be used to derive a "route".

There is an interesting similarity between this structure and mechanisms used in programming languages (where one must bind a value to a variable), or in operating systems (where one must link a particular piece of code into a module) [Shoch, 1978a].

Shoch's definition failed to capture that addresses are names too and names must eventually be mapped to routes [Jording & Andreasen, 1994]. In this sense, the idea of NL-addressing is *to use encoding of the name as route* in a multi-dimensional information space and this way to speed the access to stored information.

In this paper, firstly our attention was paid to addressing and naming (labeling) in graphs with regards to introducing the NL-addressing in graphs. A sample graph was analyzed to find its proper representation by triples.

Taking in account the interrelations between nodes and edges, we saw that a "multi-layer" representation is possible and the identifiers of nodes and edges can be avoided. As result of the analysis of the examples, the advantages and disadvantages of the multi-layer representation of graphs were pointed and the conclusion was that if we will use the indexed files or relational data bases, the disadvantages of such data models may make the implementation impossible.

A solution of this problem may be the using of NL-addressing which consists in assuming the internal computer codes of letters as co-ordinates in multi-dimensional information space. Different words and phrases have different lengths and require using of addressing with variable length of the co-ordinate arrays, i.e. to have variable dimensions in one and the same time. Such addressing we call "multidimensional".

Our starting point of realization of our approach was the Multi-Domain Information Model (MDIM) [Markov, 2004] and corresponded Multi-Domain Access Method (MDAM) [Markov, 1984], which we upgraded to NL-addressing

approach to apply for storing graphs. The possibility to use coordinates is good for graph models where it is possible to replace search with addressing. Hence, the advantages of the numbered information spaces are:

- The possibility to build growing space hierarchies of information elements;
- The great power for building interconnections between information elements stored in the information base;
- The practically unlimited number of dimensions (this is the main advantage of the numbered information spaces for graphs where it is possible "*to address, not to search*");

The NL-addressing and multi-layer organization of the information, together with the model of representing the characteristics, are good basis for implementing this approach for real solutions.

Appendix

Table 13. The Protégé QWL description of the sample graph

```
Prefix(owl:=<http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-
        18>

Declaration(Class(<http://www.co-ode.org/ontologies/ont.owl#Club>))
Declaration(Class(<http://www.co-ode.org/ontologies/ont.owl#People>))
Declaration(ObjectProperty(<http://www.co-ode.org/ontologies/ont.owl#is_member>))
AnnotationAssertion(<http://www.co-ode.org/ontologies/ont.owl#since> <http://www.co-ode.org/ontologies/ont.owl#is_member> "")
InverseObjectProperties(<http://www.co-ode.org/ontologies/ont.owl#members> <http://www.co-ode.org/ontologies/ont.owl#is_member>)
InverseFunctionalObjectProperty(<http://www.co-ode.org/ontologies/ont.owl#is_member>)
Declaration(ObjectProperty(<http://www.co-ode.org/ontologies/ont.owl#knows>))
AnnotationAssertion(<http://www.co-ode.org/ontologies/ont.owl#since> <http://www.co-ode.org/ontologies/ont.owl#knows> "")
Declaration(ObjectProperty(<http://www.co-ode.org/ontologies/ont.owl#members>))
AnnotationAssertion(<http://www.co-ode.org/ontologies/ont.owl#since> <http://www.co-ode.org/ontologies/ont.owl#members> "")
InverseObjectProperties(<http://www.co-ode.org/ontologies/ont.owl#members> <http://www.co-ode.org/ontologies/ont.owl#is_member>)
FunctionalObjectProperty(<http://www.co-ode.org/ontologies/ont.owl#members>)
Declaration(DataProperty(<http://www.co-ode.org/ontologies/ont.owl#Age>))
DataPropertyDomain(<http://www.co-ode.org/ontologies/ont.owl#Age> <http://www.co-ode.org/ontologies/ont.owl#People>)
DataPropertyDomain(<http://www.co-ode.org/ontologies/ont.owl#Age> DataAllValuesFrom(<http://www.co-ode.org/ontologies/ont.owl#Age>
        <http://www.w3.org/2001/XMLSchema#decimal>))
Declaration(DataProperty(<http://www.co-ode.org/ontologies/ont.owl#type>))
DataPropertyDomain(<http://www.co-ode.org/ontologies/ont.owl#type> DataAllValuesFrom(<http://www.co-ode.org/ontologies/ont.owl#type>
        <http://www.w3.org/2000/01/rdf-schema#Literal>))
Declaration(NamedIndividual(<http://www.co-ode.org/ontologies/ont.owl#Alice>))
ClassAssertion(<http://www.co-ode.org/ontologies/ont.owl#People> <http://www.co-ode.org/ontologies/ont.owl#Alice>)
ObjectPropertyAssertion(Annotation(<http://www.co-ode.org/ontologies/ont.owl#since> "2005/07/01") <http://www.co-
        ode.org/ontologies/ont.owl#is_member> <http://www.co-ode.org/ontologies/ont.owl#Alice> <http://www.co-ode.org/ontologies/ont.owl#Chess>)
ObjectPropertyAssertion(Annotation(<http://www.co-ode.org/ontologies/ont.owl#since> "2001/10/03") <http://www.co-ode.org/ontologies/ont.owl#knows>
        <http://www.co-ode.org/ontologies/ont.owl#Alice> <http://www.co-ode.org/ontologies/ont.owl#Bob>)
DataPropertyAssertion(<http://www.co-ode.org/ontologies/ont.owl#Age> <http://www.co-ode.org/ontologies/ont.owl#Alice> "18")
Declaration(NamedIndividual(<http://www.co-ode.org/ontologies/ont.owl#Bob>))
ClassAssertion(<http://www.co-ode.org/ontologies/ont.owl#People> <http://www.co-ode.org/ontologies/ont.owl#Bob>)
ObjectPropertyAssertion(Annotation(<http://www.co-ode.org/ontologies/ont.owl#since> "2011/02/14") <http://www.co-
        ode.org/ontologies/ont.owl#is_member> <http://www.co-ode.org/ontologies/ont.owl#Bob> <http://www.co-ode.org/ontologies/ont.owl#Chess>)
ObjectPropertyAssertion(Annotation(<http://www.co-ode.org/ontologies/ont.owl#since> "2001/10/04") <http://www.co-ode.org/ontologies/ont.owl#knows>
        <http://www.co-ode.org/ontologies/ont.owl#Bob> <http://www.co-ode.org/ontologies/ont.owl#Alice>)
DataPropertyAssertion(<http://www.co-ode.org/ontologies/ont.owl#Age> <http://www.co-ode.org/ontologies/ont.owl#Bob> "22")
Declaration(NamedIndividual(<http://www.co-ode.org/ontologies/ont.owl#Chess>))
ClassAssertion(<http://www.co-ode.org/ontologies/ont.owl#Club> <http://www.co-ode.org/ontologies/ont.owl#Chess>)
ObjectPropertyAssertion(Annotation(<http://www.co-ode.org/ontologies/ont.owl#since> "2011/02/14") <http://www.co-ode.org/ontologies/ont.owl#members>
        <http://www.co-ode.org/ontologies/ont.owl#Chess> <http://www.co-ode.org/ontologies/ont.owl#Bob>)
ObjectPropertyAssertion(Annotation(<http://www.co-ode.org/ontologies/ont.owl#since> "2005/07/01") <http://www.co-ode.org/ontologies/ont.owl#members>
        <http://www.co-ode.org/ontologies/ont.owl#Chess> <http://www.co-ode.org/ontologies/ont.owl#Alice>)
DataPropertyAssertion(<http://www.co-ode.org/ontologies/ont.owl#type> <http://www.co-ode.org/ontologies/ont.owl#Chess> "group")
Declaration(AnnotationProperty(<http://www.co-ode.org/ontologies/ont.owl#since>))
)
```

Table 14. The Protégé RDF description of the sample graph

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY ont "http://www.co-ode.org/ontologies/ont.owl#" >
   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
   <!ENTITY owl "http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/2002/07/owl#" >
   <!ENTITY xsd "http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/2001/XMLSchema#" >
   <!ENTITY xml "http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/XML/1998/namespace" >
   <!ENTITY rdfs "http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY rdf "http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-
ontology-18#"
    xml:base="http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-
18"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:ont="http://www.co-ode.org/ontologies/ont.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdf="http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#&rdf;"
    xmlns:xml="http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18#http://www.w3.org/XML/1998/namespace">
   <owl:Ontology rdf:about="http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-
18#http://www.semanticweb.org/wiki/ontologies/2013/2/untitled-ontology-18"/>
    <!--
   ///////////////////////////////////////////////////////////////////////////////////
   // Annotation properties
   ///////////////////////////////////////////////////////////////////////////////////
   -->
<!-- http://www.co-ode.org/ontologies/ont.owl#since -->
   <owl:AnnotationProperty rdf:about="&ont;since"/>

   <!--
   ///////////////////////////////////////////////////////////////////////////////////
   // Object Properties
   ///////////////////////////////////////////////////////////////////////////////////
    -->

   <!-- http://www.co-ode.org/ontologies/ont.owl#is_member -->
   <owl:ObjectProperty rdf:about="&ont;is_member">
      <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
      <ont:since></ont:since>
   </owl:ObjectProperty>

   <!-- http://www.co-ode.org/ontologies/ont.owl#knows -->
   <owl:ObjectProperty rdf:about="&ont;knows">
      <ont:since></ont:since>
   </owl:ObjectProperty>

   <!-- http://www.co-ode.org/ontologies/ont.owl#members -->
   <owl:ObjectProperty rdf:about="&ont;members">
      <rdf:type rdf:resource="&owl;FunctionalProperty"/>
      <ont:since></ont:since>
      <owl:inverseOf rdf:resource="&ont;is_member"/>
   </owl:ObjectProperty>

   <!--
   ///////////////////////////////////////////////////////////////////////////////////
   // Data properties
   ///////////////////////////////////////////////////////////////////////////////////
    -->

   <!-- http://www.co-ode.org/ontologies/ont.owl#Age -->
   <owl:DatatypeProperty rdf:about="&ont;Age">
      <rdfs:domain rdf:resource="&ont;People"/>
      <rdfs:domain>
         <owl:Restriction>
            <owl:onProperty rdf:resource="&ont;Age"/>
            <owl:allValuesFrom rdf:resource="&xsd;decimal"/>
         </owl:Restriction>
      </rdfs:domain>
```

```
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/ont.owl#type -->
    <owl:DatatypeProperty rdf:about="&ont;type">
        <rdfs:domain>
            <owl:Restriction>
                <owl:onProperty rdf:resource="&ont;type"/>
                <owl:allValuesFrom rdf:resource="&rdfs;Literal"/>
            </owl:Restriction>
        </rdfs:domain>
    </owl:DatatypeProperty>

    <!--
    ///////////////////////////////////////////////////////////////////////////////////
    // Classes
    ///////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- http://www.co-ode.org/ontologies/ont.owl#Club -->
    <owl:Class rdf:about="&ont;Club"/>

    <!-- http://www.co-ode.org/ontologies/ont.owl#People -->
    <owl:Class rdf:about="&ont;People"/>

    <!--
    ///////////////////////////////////////////////////////////////////////////////////
    // Individuals
    ///////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- http://www.co-ode.org/ontologies/ont.owl#Alice -->
    <owl:NamedIndividual rdf:about="&ont;Alice">
        <rdf:type rdf:resource="&ont;People"/>
        <ont:Age>18</ont:Age>
        <ont:knows rdf:resource="&ont;Bob"/>
        <ont:is_member rdf:resource="&ont;Chess"/>
    </owl:NamedIndividual>
    <owl:Axiom>
        <ont:since>2001/10/03</ont:since>
        <owl:annotatedSource rdf:resource="&ont;Alice"/>
        <owl:annotatedTarget rdf:resource="&ont;Bob"/>
        <owl:annotatedProperty rdf:resource="&ont;knows"/>
    </owl:Axiom>
    <owl:Axiom>
        <ont:since>2005/07/01</ont:since>
        <owl:annotatedSource rdf:resource="&ont;Alice"/>
        <owl:annotatedTarget rdf:resource="&ont;Chess"/>
        <owl:annotatedProperty rdf:resource="&ont;is_member"/>
    </owl:Axiom>

    <!-- http://www.co-ode.org/ontologies/ont.owl#Bob -->
    <owl:NamedIndividual rdf:about="&ont;Bob">
        <rdf:type rdf:resource="&ont;People"/>
        <ont:Age>22</ont:Age>
        <ont:knows rdf:resource="&ont;Alice"/>
        <ont:is_member rdf:resource="&ont;Chess"/>
    </owl:NamedIndividual>
    <owl:Axiom>
        <ont:since>2011/02/14</ont:since>
        <owl:annotatedSource rdf:resource="&ont;Bob"/>
        <owl:annotatedTarget rdf:resource="&ont;Chess"/>
        <owl:annotatedProperty rdf:resource="&ont;is_member"/>
    </owl:Axiom>
    <owl:Axiom>
        <ont:since>2001/10/04</ont:since>
        <owl:annotatedTarget rdf:resource="&ont;Alice"/>
        <owl:annotatedSource rdf:resource="&ont;Bob"/>
        <owl:annotatedProperty rdf:resource="&ont;knows"/>
    </owl:Axiom>

    <!-- http://www.co-ode.org/ontologies/ont.owl#Chess -->
    <owl:NamedIndividual rdf:about="&ont;Chess">
        <rdf:type rdf:resource="&ont;Club"/>
        <ont:type>group</ont:type>
```

```
      <ont:members rdf:resource="&ont;Alice"/>
      <ont:members rdf:resource="&ont;Bob"/>
   </owl:NamedIndividual>
   <owl:Axiom>
      <ont:since>2005/07/01</ont:since>
      <owl:annotatedTarget rdf:resource="&ont;Alice"/>
      <owl:annotatedSource rdf:resource="&ont;Chess"/>
      <owl:annotatedProperty rdf:resource="&ont;members"/>
   </owl:Axiom>
   <owl:Axiom>
      <ont:since>2011/02/14</ont:since>
      <owl:annotatedTarget rdf:resource="&ont;Bob"/>
      <owl:annotatedSource rdf:resource="&ont;Chess"/>
      <owl:annotatedProperty rdf:resource="&ont;members"/>
   </owl:Axiom>
</rdf:RDF>

<!-- Generated by the OWL API (version 3.4.2) http://owlapi.sourceforge.net -->
```

Acknowledgments

The article is partially financed by the project ITHEA NLA (www.ithea.org).

Bibliography

[Abiteboul et al, 1997] Abiteboul S., Quass D., McHugh J., Widom J., and Wiener J. L "The Lorel query language for semistructured data" International Journal on Digital Libraries (JODL) 1, 1, 1997, pp. 68–88.

[AlegroGraph, 2012] AllegroGraph® 4.8, http://www.franz.com/agraph/allegrograph/ (accessed: 25.08.2012).

[Angles & Gutierrez, 2008] Angles R., C. Gutierrez "Survey of Graph Database Models", ACM Computing Surveys, Vol. 40, No. 1, Article 1, Publication date: February 2008, DOI 10.1145/1322432.1322433, http://doi.acm.org/10.1145/1322432.1322433, pp.1 - 39

[Buneman et al, 1996] Buneman, P., Davidson, S., Hillebrand, G., and Suciu, D. "A Query Language and Optimization Techniques for Unstructured Data", SIGMOD Record. 25, 2, 1996, pp. 505-516.

[Carroll et al, 2005] J.J. Carroll, C. Bizer, P. Hayes, P. Stickler. Named Graphs, Provenance and Trust; International World Wide Web Conference (IW3C2); WWW 2005, Chiba, Japan. ACM 1595930469/05/0005. pp. 613-622. http://www.ra.ethz.ch/cdstore/www2005/docs/p613.pdf (accessed: 21.02.2013)

[Euler, 1736] Leonhard Euler. Solutio Problematis a geometriam situs pertinentis. Commentarii Academiae Scientiarum Imperialis Petropolitanae 8 (1736), pp. 128-140. http://www.math.dartmouth.edu/~euler/docs/originals/E053.pdf (accessed: 21.02.2013)

[Gallian, 2011] Joseph A. Gallian. A Dynamic Survey of Graph Labeling - The electronic journal of combinatorics 18, (2011); #DS6; pp. 1-256. http://emis.matem.unam.mx/journals/EJC/Surveys/ds6.pdf (accessed: 21.02.2013)

[GraphDB, 2012] http://www.smartlab.at/tag/graphdb/ (accessed: 01.12.2012)

[Graves et al, 1995a] Graves, M., Bergeman, E. R., and Lawrence, C. B. "A Graph-Theoretic Data Model for Genome Mapping Databases", In Proc. of the 28th Hawaii Int. Conf. on System Sciences (HICSS), IEEE Computer Society, 32, 1995a.

[Graves et al, 1995b] Graves, M., Bergeman, E. R., and Lawrence, C. B. "Graph Database Systems for Genomics", IEEE Engineering in Medicine and Biology, Special issue on Managing Data for the Human Genome Project 11, 6, 1995b.

[Guting, 1994] Guting, R. H. "GraphDB: Modeling and Querying Graphs in Databases", in: Proc. of 20th, Int. Conf. on Very Large Data Bases (VLDB). Morgan Kaufmann, 1994, pp. 297–308.

[Gyssens et al, 1990] Gyssens, M., Paredaens, J., den Bussche, J. V., and Gucht, D. V. A "Graph-Oriented Object Database Model", in: Proc. of the 9th Symposium on Principles of Database Systems (PODS), ACM Press, 1990, pp. 417–424.

[Harju, 2011] Tero Harju "Lecture Notes on Graph Theory", Department of Mathematics University of Turku - Turku, Finland. 2011. http://cs.bme.hu/fcs/graphtheory.pdf (accessed: 21.02.2013).

[Hayes, 2004] Patrick Hayes, Editor, RDF Semantics, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-mt-20040210/ . Latest version available at http://www.w3.org/TR/rdf-mt/ (accessed: 21.02.2013).

[Hertel et al, 2009] Alice Hertel, Jeen Broekstra, and Heiner Stuckenschmidt. RDF Storage and Retrieval Systems. In: S. Staab and R. Studer (eds.), Handbook on Ontologies, International Handbooks on Information Systems, DOI 10.1007/978-3-540-92673-3, Springer-Verlag Berlin Heidelberg 2009. pp 489-508.

[Ivanova et al, 2012] Kr. Ivanova, V. Velychko, Kr. Markov. About NL-addressing (К вопросу о естествено-языконой адрессации). In: V. Velychko et al (ed.), Problems of Computer in Intellectualization. ITHEA® 2012, Kiev, Ukraine - Sofia, Bulgaria, ISBN: 978-954-16-0061-0 (printed), ISBN: 978-954-16-0062-7 (online), pp. 77-83 (in Russian).

[Ivanova et al, 2013] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Introduction on the Natural Language Addressing", International Journal of Information Technologies and Knowledge, Vol.7, Number 2, 2013, pp. 139–146. ISSN 1313-0455

[Jording & Andreasen, 1994] Nick Jording and Flemming Andreasen "A Distributed Wide Area Name Service for an Object Oriented Programming System", DIKU, Department of Computer Science, University of Copenhagen, Denmark, 1994.

[Klyne & Carroll, 2004] Graham Klyne and Jeremy J. Carroll, Editors, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ . Latest version available at http://www.w3.org/TR/rdf-concepts/ (accessed: 21.02.2013).

[Kunii, 1987] Kunii, H. S. DBMS with Graph Data Model for Knowledge Handling. In Proc. of the 1987 Fall Joint Computer Conference on Exploring technology: today and tomorrow. IEEE Computer Society Press, 1987, pp. 138–142.

[Kuper & Vardi, 1984] Kuper, G. M. and Vardi, M. Y. "A New Approach to Database Logic", In: Proc. of the 3th Symposium on Principles of Database Systems (PODS), ACM Press, 1984, pp. 86 96.

[Kuper & Vardi, 1993] Kuper, G. M. and Vardi, M. Y. "The Logical Data Model" ACM Transactions on Database Systems (TODS) 18, 3, 1993, pp. 379–413.

[Levene & Loizou, 1995] Levene, M. and Loizou, G. "A Graph-Based Data Model and its Ramifications", IEEE Transactions on Knowledge and Data Engineering (TKDE) 7, 5, 1995, pp. 809–823.

[Levene & Poulovassilis, 1990] Levene, M. and Poulovassilis, A. "The Hypernode Model and its Associated Query Language", In: Proc. of the 5th Jerusalem Conf. on Information technology. IEEE Computer Society Press, 1990, pp. 520–530.

[Levene & Poulovassilis, 1991] Levene, M. and Poulovassilis, A. "An Object-Oriented Data Model Formalised Through Hypergraphs", Data & Knowledge Engineering, (DKE) 6, 3, 1991, pp. 205 - 224.

[LTS, 2012] LargeTripleStores http://www.w3.org/wiki/LargeTripleStores (accessed: 29.08.2012)

[Markov et al, 2008] Markov, K., Ivanova, K., Mitov, I., & Karastanev, S. Advance of the access methods. International Journal of Information Technologies and Knowledge, 2(2), 2008, pp. 123–135.

[Markov, 1984] Markov Kr. A Multi-domain Access Method.//Proceedings of the International Conference on Computer Based Scientific Research, Plovdiv, 1984. pp. 558-563.

[Markov, 2004] Markov, K. Multi-domain information model, Int. J. Information Theories and Applications, 11/4, 2004, pp. 303-308.

[NG, 2013] W3C. Named Graphs http://www.w3.org/2004/03/trix/ (accessed: 21.02.2013).

[Pan & Pan, 2006] Pan D. and Y. Pan "Using Ontology Repository to Support Data Mining", Proceedings of the 6th, World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China, pp. 5947 – 5951.

[Paredaens et al, 1995] Paredaens, J., Peelman, P., and Tanca, L. 1995. G-Log: A Graph-Based Query Language. IEEE Transactions on Knowledge and Data Engineering (TKDE) 7, 3, 436–453.

[protégé, 2012] http://protege.stanford.edu (accessed: 25.05.2012).

[protege-owl, 2012] http://protege.stanford.edu/overview/protege-owl.html (accessed: 25.05.2012)

[RDF, 2013] http://www.w3.org/RDF/#specs (accessed: 21.02.2013).

[Shoch, 1978a] John F. Shoch "A note on Inter-Network Naming, Addressing and Routing", Xerox, Palo Alto, Research Center, Palo Alto - California 94305, USA, January 1978. http://www.postel.org/ien/pdf/ien019.pdf (accessed: 21.02.2013)

[Shoch, 1978b] John F. Shoch. Inter-Network Naming, Addressing, and Routing. In Proc. of the Seventeenth IEEE Conference on Computer Communication Networks, pages 72–79, Washington, D.C., 1978.

[Weisstein, 2013] Eric W. Weisstein Labeled Graph, From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/LabeledGraph.html (accessed: 21.02.2013)

[Yabloko, 2011] Yabloko L. OntoBase, Protégé 2011 http://protegewiki.stanford.edu/wiki/OntoBase (accessed: 02.08.2012)

Authors' Information

Ivanova Krassimira *– University of National and World Economy, Sofia, Bulgaria. Institute of Mathematics and Informatics, BAS, Sofia, Bulgaria.*
e-mail: krasy78@mail.bg
Major Fields of Scientific Research: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems

Vanhoof Koen - *Professor Dr.,*
Universiteit Hasselt; Campus Diepenbeek; Department of Applied Economic Sciences
Wetenschapspark 5; bus 6; BE-3590 Diepenbeek; Belgium
e-mail: koen.vanhoof@uhasselt.be
Main research areas: data mining, knowledge retrieval.

Markov Krassimir *– ITHEA ISS IJ, IBS and IRJ Editor in chief, P.O. Box: 775, Sofia-1090, Bulgaria; e-mail: markov@foibg.com*
Major Fields of Scientific Research: General theoretical information research, Multi-dimensional models and information systems

Velychko Vitalii – *Institute of Cybernetics, NASU, Kiev, Ukraine*
e-mail: Velychko@rambler.ru
Major Fields of Scientific Research: Data Mining, Natural Language Processing

COMMON BUILDING ISSUES OF KNOWLEDGE-BASED MANAGEMENT INFORMATION SYSTEM FOR SUPPORTING PERSONNEL MOTIVATION

Elena Antonova, Nikita Lyalyakin

Abstract: *The possibility of creating a knowledge-based management information systems (MIS) for support of solving personnel motivation tasks in principle is considered in this article, there is an emphasis and description of the main issues that need to be addressed when developing such systems.*

Keywords: *management information systems.*

Корректное функционирование любой системы управления зависит в первую очередь от правильности и своевременности принятия решений субъектом управления (лица, принимающего решения - ЛПР). Управление предприятием является многоцелевым, при этом правильность и своевременность принятия решений – это необходимое условие достижения целей организации. Среди целей разного уровня выделяют стратегические, перспективные и тактические, оперативные. Цели могут быть общими для всей организации и частными для ее различных подразделений, могут отличаться по содержанию (экономические, социальные, организационные, технические, производственные), носить комплексный характер, быть промежуточными и конечными [Управление организацией, 2001]. Ключевым объектом управления являются сотрудники организации, обладающие совокупностью психофизических свойств и ситуационных факторов, которые обуславливают случаи проявления т.н. «ненадёжности» – забывчивости, некачественного, медленного или неэффективного выполнения служебных обязанностей, негативно влияющих на достижение целей организации. Для достижения вышеперечисленных целей предприятия, предотвращения ненадёжности персонала и снижения негативного эффекта используют мотивации – воздействия на поведение сотрудников. Комплекс мотиваций представляет собой т.н. политику мотивации.

Для помощи ЛПР в выполнении их обязанностей существует класс систем поддержки принятия решений (СППР), в котором выделяют подкласс ИС менеджмента. ИС менеджмента - это интегрированная человеко-машинная система, обеспечивающая информацией, необходимой для операций, менеджмента и приятия решений в организациях. Системы используют компьютерную технику и программное обеспечение, модели управления и принятия решений, а также базу данных [Davis, 1974].

Автоматизация процессов в современной организации, как правило, подразумевает использование информационных систем управления персоналом, решающих следующие задачи:

1. управление личной информацией о сотрудниках;
2. отчётность и анализ информации о сотрудниках;
3. управление трудовой документацией организации, в т.ч. руководствами для сотрудников, регламентами, правилами техники безопасности, охраны труда и т.д.;
4. управление обучением;
5. управление зарплатой и премиями;

6. регистрация и управление резюме.

Задачи разработки политики мотивации сотрудников, выбора, применения и оценки эффективности мотиваций полностью выполняются менеджментом организаций и отделами кадров, и, несмотря на свою важность, на протяжении всей истории развития ИС менеджмента не рассматривались как объёкт частичной или полной автоматизации [Management Information Systems, 1995] [Power, 2011] . Решение подобных задач невозможно полностью автоматизировать в силу специфики предметной области – социально-психологического процесса на предприятии, - однако возможно предоставить автоматизированную поддержку менеджменту предприятия. Система, оказывающая таковую поддержку, должна быть интеллектуальной, т.е. объяснять мотивы тех или иных предлагаемых решений. В ходе дипломной работы и освоения полученного гранта [ДВГУ, 2010] был проведён анализ предметной области «предотвращение ненадёжности персонала», спроектирован и реализован прототип системы оценки мотиваций как составной части системы интеллектуальной поддержки при управлении персоналом. Проделанная работа и дальнейшие исследования класса ИС менеджмента привели к выделению следующих пяти принципиальных проблем, которые необходимо решить в ходе проектирования и реализации интеллектуальной ИС менеджмента поддержки решения задач мотивации персонала:

1. Выбор модели представления знаний о личностях сотрудников, их предпочтений и ожиданий от рабочего процесса.

В ИС менеджмента поддержки решения задач мотивации персонала недостаточно одной лишь базы данных, содержащей записи о сотрудниках и их психофизические характеристики. Наличие системы правил и зависимостей между характеристиками сотрудников и эффективностью применяемых к ним мотиваций, а также вновь разрабатываемых методик тестирования персонала на предмет удовлетворённости потребностей и ожиданий и оценки личного отношения к мотивациям с помощью опросников обуславливает необходимость использования пополняемой базы знаний.

Сложность представления знаний о сотрудниках и мотивациях, вероятно, не позволит использовать для этих целей единственную модель, но вынудит использовать сложную, смешанную модель. Выбранные модели должны обеспечивать всю полноту представления знаний о применимости воздействий к сотрудникам с определёнными психофизическими характеристиками пола, возраста, темперамента и некоторыми другими свойствами. Так, разрабатываемая ИС менеджмента должна, основываясь на указанных знаниях, всегда иметь чёткий ответ на вопросы: «какую мотивацию применить к данному сотруднику?», «будет ли данная мотивация наиболее эффективной на этом участке работ?» и «насколько снизится эффективность мотивации после следующего применения?».

2. Выбор теории мотивации.

До начала разработки ИС менеджмента необходимо выбрать наиболее подходящую с точки зрения формализации и автоматизации теорию мотивации, которая ляжет в основу всей системы, т.е. обеспечить основу для формальной модели мотиваций как социально-психологического процесса и непосредственно произвести процесс формализации.

В настоящее время выделяют две группы теорий мотивации: содержательные и процессуальные. Первые основаны на анализе потребностей человека; вторые – на оценке ситуаций, возникающих в процессе мотивации. Авторами наиболее известных содержательных теорий мотивации являются А. Маслоу, К. Альдерфер, Д. МакКелланд и Ф.Герцберг. Широко известной и часто цитируемой является теория А. Маслоу - «Пирамида Маслоу» или «Пирамида потребностей человека».

Из процессуальных теорий мотивации обычно отмечают теорию ожидания, теорию справедливости и модель Портера-Лаулера. В теории справедливости подчёркивается важность соотношений между усилиями членов коллектива и получаемыми вознаграждениями: если человек считает, что его недооценивают, то мотивация и результативность труда снижаются. Однако эти теории скорее носят характер гипотез или рекомендаций, и, как правило, не применяются менеджерами организаций. Современные системы мотивации ориентированы на развитие творческих способностей и инициативы сотрудников, гуманизацию труда, партнёрские отношения в коллективе, согласование интересов организации, подразделений и всех групп персонала [Самоукина, 2006].

При выборе теории мотивации необходимо учитывать влияние на применимость и эффективность мотивации психофизических характеристик сотрудников. А также возможность автоматизации измерения и хранения таковых характеристик. Так, некоторые авторы [7] приводят зависимость между разными типами темпераментов и применимыми типами мотиваций: в различных ситуациях применение ряда мотиваций к сотрудникам с определёнными типами темпераментов крайне нежелательно, т.к. может вызвать отрицательный эффект, и наоборот, применение других мотиваций к этим же сотрудникам может дать колоссальный прирост эффективности решения задач. Так как подобные характеристики, как правило, выявляются с помощью тестирования и анкетирования, т.е. являются вполне компьютеризируемыми, необходимо предусмотреть возможность их ввода в разрабатываемую систему.

3. Выбор каналов доставки мотивации.

В зависимости от характера работы сотрудника (офисная или полевая, постоянная или периодическая), характера мотивации (индивидуальная, коллективная) и психофизических характеристик сотрудника возможно применение различных каналов доставки мотиваций.

По способу доставки каналы можно разделить на:

физический: вручение премий, повышение зарплаты, вручение памятных подарков и дипломов;

виртуальный: отправка мотивирующих e-mail и SMS-сообщений, использование виртуальных досок почёта.

По направленности каналы можно разделить на:

1. персональный: применение мотивации к сотруднику без извещения об этом остальных членов коллектива;
2. коллективный: публичное применение мотивации, направленное на повышение ценности сотрудника в глазах его коллег и повышение самооценки сотрудника.

4. Выбор методов оценки эффективности мотиваций и прогнозирования эффективности мотиваций.

Наличие истории применения мотиваций к сотрудникам и корреляционных зависимостей между применяемыми мотивациями и эффективностью выполнения должностных обязанностей позволяет говорить о принципиальной возможности разработки методов оценки эффективности мотиваций. Кроме того, наличие формальной модели социально-психологического процесса (что было рассмотрено ранее) позволяет использовать методы прогнозирования величин эффективности мотивации. Однако разработка возможных методов оценки эффективности и прогнозирования этой величины, апробация и выбор наиболее качественного из этих методов является нерешённой на данный момент проблемой.

5. Определение места создаваемой системы среди информационных систем предприятия и её интеграция с ИС менеджмента, ИС управления персоналом и другими программными системами.

ИС менеджмента поддержки решения задач мотивации персонала должна создаваться с учётом внедрённых на предприятии прочих ИС менеджмента, автоматизированных систем управления, систем

управления персоналом и других программных систем, а также порядка их использования. Разрабатываемая система должна иметь механизмы интеграции с системами управления потоками работ (workflow management), проектами (project management), программными средствами планирования и отслеживания исполнения задач и автоматизированными системами опроса сотрудников. Для получения оценки эффективности мотивации разрабатываемая ИС менеджмента должна иметь доступ к расчётным и фактическим срокам выполнения заданий сотрудниками предприятия, а также к планам руководителей, сформированным в программных системах планирования работ, анализировать и обрабатывать их. Помимо этого, должны быть предусмотрены механизмы ввода результатов внешних (в т.ч. «физических») опросов сотрудников предприятия в систему.

Рассмотренные проблемы, определённо, отражают не все сложности разработки интеллектуальной ИС менеджмента поддержки решения задач мотивации персонала, и в ходе проектирования и разработки такой системы вскроется целый пласт побочных вопросов. Однако решение представленных проблем является необходимым условием успешной разработки подобной системы и обеспечит в достаточной степени универсальную автоматизацию.

Bibliography

[Davis, 1974] Davis G. Management Information Systems: Conceptual Foundations, Structure, and Development. — New York: McGraw-Hill, 1974.

[Management Information Systems, 1995] Management Information Systems. Comptroller's Handbook [Электронный ресурс] // Comptroller of the Currency Administrator of National Banks: [сайт]. [1995]. Систем. требования: Adobe AcrobatReader. URL: http://www.occ.gov/static/publications/handbook/mis.pdf

[Power, 2011] Power D.J. A Brief History of Decision Support Systems [Электронный ресурс] // DSSResources: [сайт]. URL: http://dssresources.com/history/dsshistory.html (дата обращения 03.04.2011)

[Генкин, 2007] Генкин Б.М. Экономика и социология труда: учеб. для вузов / Б.М. Генкин. – 7-е изд., доп. – М.: Норма, 2007. – 448 с.

[ДВГУ, 2010] Материалы научной конференции студентов и аспирантов ДВГУ – 2010. / Отв. ред. А.Н. Шушин. – Владивосток: Изд-во Дальневост. ун-та, 2010. – 332 с.

[Самоукина, 2006] Самоукина Н.В. Эффективная мотивация персонала при минимальных финансовых затратах / Н.В. Самоукина. – М.: Вершина, 2006. – 224 с.: ил.

[Управление организацией, 2001] Управление организацией. Энциклопедический словарь. М.: ИНФРА-М, 2001.

AUTHENTICATION BASED ON FINGERPRINTS WITH STEGANOGRAPHIC DATA PROTECTION

Narek Malkhasyan

Abstract: *This paper examines security problems of biometric based authentication. An authentication method is suggested, which is based on fingerprints with steganographic data protection in all stages of functioning. Suggested procedures of fingerprint based enrollment and authentication are also functionally described.*

Introduction

The increasing computerization of society, together with the prevalence of the internet and "cloud" technologies leads to the fact that both organizations and individuals increasingly rely on modern informational tools. The increasingly complex IT infrastructure of enterprises and organizations, along with the changing nature of the threats and risks make information security a vital issue. However, quite different and effective information security methods can be practically useless if they are not reinforced by convenient and reliable means of authentication (identity establishment) of consumers of information services.

Recent years are characterized by steady increase in interest in biometric authentication methods, which are based on physiological and behavioral characteristics of the user, and which are far better than traditional means, such as passwords, ID cards, etc.One main reason for this popularity is the ability of biometric technology to relatively simply and easily distinguish legitimate users from hackers attempting to fraudulently obtain access rights to information resources[1, 2]. Currently the most common are the technologies based on fingerprint,as these are the most convenient to use and the most cost-effective. Although, it should be noted that the results for fingerprints are mostly applicable in other biometric systems after undergoing some minor customizations.

At the same time, an analysis of possible attacks on the authentication system based on fingerprints shows that one of the major challenges is to ensure the security and integrity of biometric data[3]. It's obvious that the biometric data, having a high degree of uniqueness, in practice are poorly protected against copying, misuse or modification. Essentially a biometric authentication system can work properly only if it is able to ensure that duringenrollment and authentication data have been received from the relevant user and have not been subjected to external influence[4].Therefore, from the point of view of facilitating widespread usage of biometric methods,the task of protecting biometric data, in particular fingerprint data becomes critical.

Fingerprint protection

Fingerprint security can be achieved through the use of cryptographic techniques. Cryptography can be used for encrypting the fingerprint after scanning to ensure the safe transfer and storage. At the authentication stage both stored during the enrollment and received by the server fingerprint data are being deciphered for the matching procedure. The result is the security of fingerprint data, as it cannot be used or modified without the correct decryption with the corresponding private key. In general, cryptography can be used also to monitor the integrity of the fingerprint to confirm the authenticity of the source.

It should be mentioned that in many security applications biometric information is used to generate passwords for protecting cryptographic keys. A good survey on this problem can be found in [5]. In applications like the one described above it is critical to protect biometric information especially when it is transmitted remotely through an insecure channel.

The implementation of the mentioned protection of biometric data through the use of steganographic techniques [6] seems to befairly promising. While cryptography is primarily focused on techniques designed to make the encrypted information meaningless to outsiders, steganography is based on the concealment of the fact of the existence of secret information. As a result steganographic techniques can be used to protect the fingerprint with the same success providing both security and integrity of data transmitted from the client to the server or stored on the server. This significantly reduces the chance of unauthorized acquisition of biometric data, thus reducing the likelihood of misuse or alteration.

In order to monitor the integrity and authenticity of the fingerprint data source, steganography can be used to embed special labels in the image, often called digital watermarks. The mentioned labels can be embedded both assymbolic and graphic information. In the process of authentication labelsembedded in the fingerprint image can be extracted and used to verify the integrity and authenticity. Also the same steganographic techniques can be used for protecting fingerprint images by embedding them in other, not suspicious objects (such as images) often called containers. In other words, we can consider two main areas of steganographic protection of fingerprint data:

- embedding of special labels into fingerprint image for integrity monitoring,
- embeddingof fingerprint images into other images for more secure communication.

Authentication Method

Acombination of these approaches to the steganographic protection is proposed for comprehensive protection of fingerprint data. At first identification labels will be embedded into the fingerprint image, and thenit will be embedded into the container for secure communication. In the general case, the authentication is preceded by user enrollment process, during which the user data is recorded on the server side, and entered into the appropriate storage [7]. Functional scheme of the proposed enrollment process is shown in Figure 1.

User enrollment is carried out in the following order:

- Images of user's two different fingers are acquired using a suitable scanner, the first of which will then be used for authentication and the second one will be used for integrity control.
- An image is chosen from an archive stored in the user's computer, which will be used as a container.
- Both scanned fingerprint images are embedded into the container using appropriate steganographic algorithm and key.
- The filled container together with the user ID (name, nickname, password ...)is sent to the server side.
- The resulting filled container with the appropriate user ID is stored in the storage of fingerprints on the server side.

In this procedure, any image file (photo, drawing ...) stored on the computer and having a suitable format for the selected steganographic algorithm can be used as a container. Corresponding selection of steganographic algorithm and a secret key can provide a high level of protection of the fingerprint data during transmission and storage on the server. Other files supported by the selected steganographic algorithm, can also be used as a container in the mentioned scheme (audio, video ...) [6]. It should be noted that the enrollment procedure of a user is carried out not often, usually only once, therefore, to provide a high level of security it is quite permissible to use full-sized fingerprint images and large steganography container files.

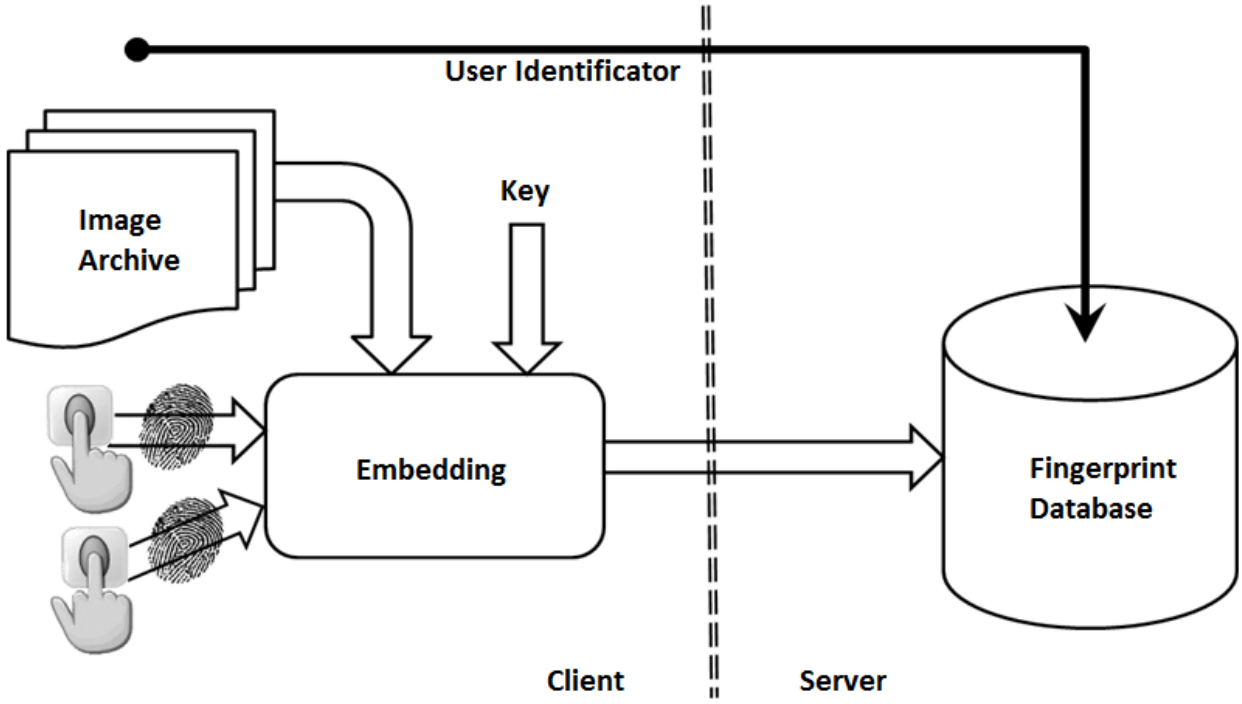


Figure 1. Functional Scheme of User Registration Procedure

Obviously, during the authentication stage the same fingers of the user should be scanned and securely transmitted to the server. To provide a higher level of secrecy of the data transmission it is possible to use synthetic or real image of some fingerprint as a container. This image is not really involved in the authentication and only distracts the attention of potential adversary who can intercept the information passed through an open channel. At the same time it is necessary to consider that from security perspective, the amount of embedded information should be much smaller than the volume of the container[8]. Therefore, it is proposed to extract a fragment from the first fingerprint image, which has suitable dimensions for embedding, and yet contains sufficient information for authentication. Additionally it is proposed to use minutiae points extracted from the second fingerprint image as a digital watermark, which must be embedded into the first fingerprint image to ensure the authenticity of the fingerprint data source. Based on the above an authentication procedure with steganographic data protection is proposed. The functional diagram of operations of the proposed procedure is shown in Figure 2.

Preparation of information required for authentication is carried out in the following order:

- Two fingers of the user, which have been used at the enrollment stage, are scanned using a suitable scanner.
- A fingerprint image is selected from an archive stored on the user's computer, which will be used as a container.
- A fragment of necessary size is extracted from the first fingerprint image in accordance with the volume of the selected container.
- Minutiae points are extracted from the second fingerprint image.
- The minutiae points' parameters extracted from the second fingerprint image are embedded into the extracted fragment of the first fingerprint image as a digital watermark using appropriate steganographic algorithm and key.
- The fragment of first fingerprint image is embedded into the container using appropriate steganographic algorithm and key.
- The filled container together with the user ID is passed to the server.

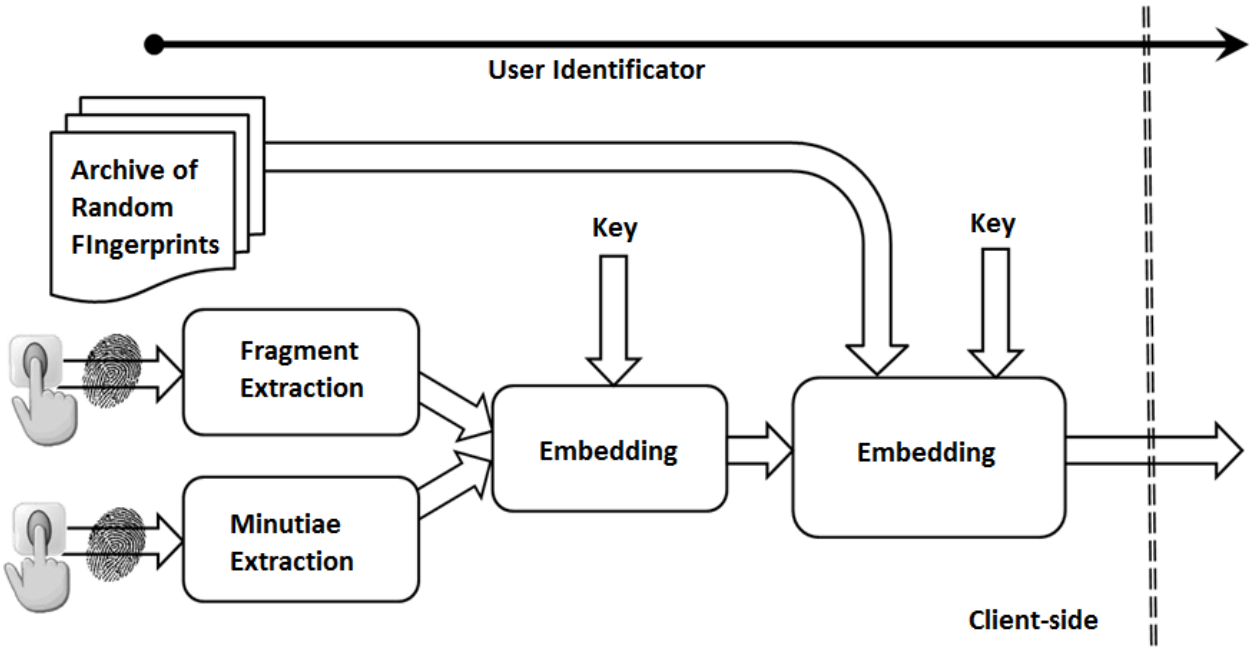


Figure 2. Functional Scheme of Client Side User Authentication Procedure

The functional scheme of the operations of the proposed authentication procedure on the server-side is shown in Figure 3.

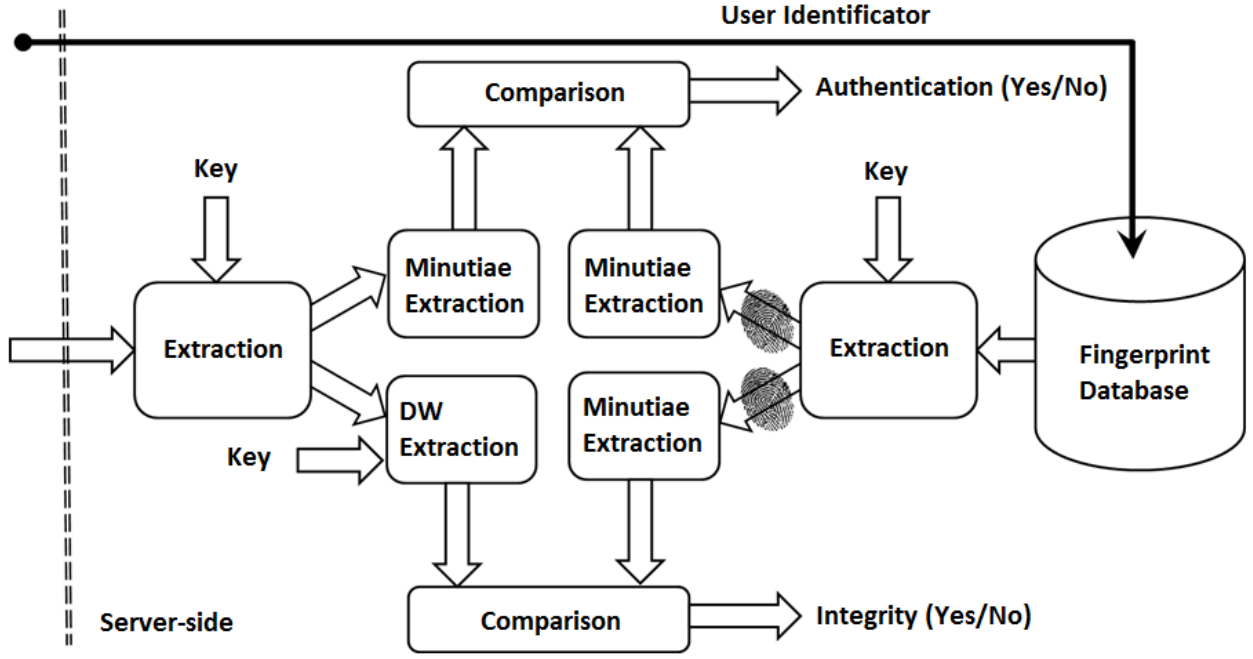


Figure 3. Functional Scheme of Server Side User Authentication

Processing of thereceived information needed for authentication on the server side is carried out in the following order:

- A filled container from the server side storage of fingerprints is retrieved according to the received user ID.

- Two embedded fingerprint images are extracted from the container using appropriate steganographic algorithm and key.
- Minutiae points are extracted from these fingerprint images.
- The fingerprint image fragment is extracted from the received container using appropriate steganographic algorithm and key.
- Using appropriate steganographic algorithm and key the digital watermark is extracted from the fingerprint image fragment, which is a set of parameters of minutiae points of the second fingerprint.
- Minutiae points' parameters extracted from the second fingerprint image, which has been retrieved from storage, are compared with the extracted digital watermark data and a decision regarding the integrity of the received fragment is made.
- Minutiae points are extracted from the extracted fingerprint image fragment.
- Minutiae points' parameters extracted from the first fingerprint image, which has been retrieved from storage, are compared with the minutiae points' parameters extracted from the received fragment and a decision regarding the user authenticity is made.

It should be noted that the same key should be used for embedding into the containers and extracting from them fingerprint images and digital watermarks in all phases of authentication. This key may be agreed between the user and the server as a part of the communication protocol. For safety reasons, different keys can be used for each embedding/extraction operation, which obviously will significantly increase the level of protection, but also will significantly complicate the protocols and procedures.

Security Considerations

Although the fingerprint recognition technique is the dominant technology in the biometric market, it may suffer attacks at different points during the authentication process. The most common attacks occur by the use of fake fingerprint images. These fake fingerprint images can be acquired from different surfaces touched by the legitimate user (such as glasses, doorknobs, glossy paper, etc.) The vulnerability to this kind of attacks has always been considered as a major drawback of fingerprint-based authentication systems.

The suggested method alleviates the mentioned vulnerability to some extent. Let us consider the security of the system from the perspective of an attacker who has access to the communication channel between the client and server modules of the proposed method and who has illegitimately acquired the fingerprint image of a legitimate user of the system. Also let us assume that the attacker has somehow managed to break the steganographic algorithm used to embed fingerprint images into a container, although this task is very hard to accomplish on it's own in the case of appropriate steganographic algorithm selection. In order to pass successful authentication on behalf of the legitimate user the attacker has to take the following additional steps:

- guess the necessity of embedded digital watermark in the fingerprint image,
- acquire the minutiae points of the fingerprint image of the legitimate user's second finger, which obviously is a harder task than acquiring just the fingerprint image from some surface touched by the user,
- break the digital watermark algorithm in order to embed the acquired minutiae points into the fingerprint image of the user's first finger, which, in the case of appropriate algorithm selection, is a very hard task also.

Thus, we can conclude that the proposed scheme adds an additional security layer to the fingerprint authentication process.

Conclusion

The proposed method represented in this paper provides security and integrity of transmitted and stored fingerprint data through the use of steganographic data protection methods. A slight modification of this method can provide its applicability to the problem of identification of registered users.

As a suggestion for a future research in this area the following directionscould be pointed out:

- Development of steganographic algorithms specifically adopted for effective embedding of fingerprint images into containers.
- Development of methodology for extracting fingerprint image fragments of needed size,sufficient for successful authentication.

References

1. A. K. Jain, A. A. Ross,K.Nandakumar, *Introduction to Biometrics*, Springer, 2011, 311 p.
2. L. O'Gorman, *Comparing Passwords, Tokens, and Biometrics for User Authentication*. Proc. IEEE, vol. 91, no. 12, 2003, pp. 2021–2040
3. J. Galbally, J. Fierrez, F. Alonso–Fernandez,M. Martinez–Diaz,*Evaluation of Direct Attacks to Fingerprint Verification Systems*, Telecommunication Systems (Springer), Volume 47, Numbers 3–4, 2011, pp. 243–254
4. B. Schneier, *The Uses and Abuses of Biometrics*, Communications of the ACM, vol. 42, no. 8, 1999, p. 136
5. G. Khachatrian, N. Malkhasyan, Overview of Methods of Biometric Based Key Protection, 2012
6. E. Cole, *Hiding in Plain Sight: Steganography and the Art of Covert Communication*,John Wiley & Sons, 2003. 360 p.
7. D. Maltoni, D. Maio, A. K. Jain, S.Prabhakar, *Handbook of Fingerprint Recognition (Second Edition)*, Springer, 2009, 348 p.
8. 8. G.Margarov, *Data Hiding on the Internet: Steganalysis against Steganography*, in "Terrorism and the Internet – Threats – Target groups – Deradicalisation strategies", IOS Press, 2010, pp. 167–182

Authors information

Narek Malkhasyan - *Institute for Informatics and Automation Problems, National Academy of Sciences of Armenia*

ПЬЕЗООПТИЧЕСКИЕ СКАНИРУЮЩИЕ КОММУТАТОРЫ

Рябцов А.В.

***Аннотация**: В статье рассмотрены оптоволоконные коммутационные устройства на основе трубчатых и монолитных пьезоэлектрических актуаторов.*

***Ключевые слова**: Волоконная оптика – Коммутационные устройства – Пьезоэлектрические актуаторы.*

***Ключевые слова классификации АСМ**: H.4.3 Communications Applications.*

Введение

В настоящее время в мире идет активная борьба за лидерство в информационной сфере. Особое внимание уделяется полностью оптическим сетям, как наиболее перспективным среди прочих информационно-коммуникационных технологий с точки зрения увеличения объема и скорости передачи информации [Скляров, 2004, Гринфильд, 2002]. Основные усилия большинства ведущих производителей сетевого оборудования направлены сейчас на создание простых и эффективных устройств для коммутации оптических сигналов. На сегодняшний день в мире разработано и используется большое число оптических коммутаторов, основанных на различных физических принципах, но, ни один из них не является лидирующим в данной области в силу имеющихся схемных ограничений или конструктивных особенностей [Гайворонская, 2011, Гайворонская, 2010]. Одним из перспективных направлений является разработка пространственных оптических коммутационных устройств на основе микро-электромеханических систем (MEMS) с различного вида актуаторами - устройствами, осуществляющими микроперемещения подвижных частей оптического коммутатора [Little, 2001, Dooyoung, 2004].

Сканирующие пьезоактуаторы

В связи с этим представляет интерес применение сканирующих пьезоэлектрических элементов в качестве актуаторов оптических переключателей. Так, например, хорошо отработанная технология производства трубчатых пьезоэлектрических элементов (ТПЭ), широко применявшихся в недалеком прошлом в качестве чувствительных элементов звукоснимателей [Хазе, 1975], и с успехом применяющаяся сейчас в

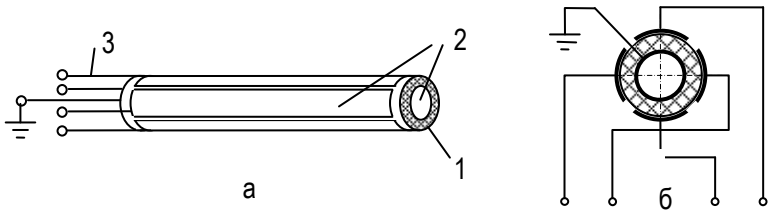


Рисунок 1. Трубчатый пьезоэлемент (а) и схема подключения его электродов (б):
1- трубчатый пьезоэлектрический элемент, 2 - плоские электроды, 3 - выводы

электронной микроскопии [Неволин, 1996], может быть использована в новом качестве в многоканальных оптических коммутаторах. Основным достоинством ТПЭ (рис. 1) является возможность получения изгибных пространственных деформаций в двух координатах при относительно небольших управляющих напряжениях. ТПЭ представляют собой тонкостенные полые цилиндры 1 из титаната бария или цирконата титаната свинца, на внешнюю и внутреннюю поверхности которых методом напыления нанесены электроды 2 с проволочными или ленточными выводами 3 для подключения ТПЭ к схеме управления. Электрод на внутренней поверхности ТПЭ обычно выполняется сплошным, а внешний электрод продольно разделен на четыре части по-квадрантно, как показано на рисунке 1.

Под действием напряжения, приложенного к электродам, пьезоэлемент изменяет свои геометрические размеры. Механическое напряжение, возникающее в ТПЭ вследствие обратного пьезоэффекта может быть описано в общем виде уравнением [Гайворонская, 2011]:

$$u_{ij} = d_{ijk} \cdot E_k \qquad (1)$$

где u_{ij} – тензор деформаций, $E_к$ – компоненты электрического поля, d_{ijk} – компоненты тензора пьезоэлектрических коэффициентов. При этом отличными от нуля являются только три коэффициента: d_{33}, d_{31} и d_{15}, характеризующие продольные, поперечные и сдвиговые деформации. Поперечными и сдвиговыми деформациями в ТПЭ в данном случае можно пренебречь. Тогда величина продольной деформации в ТПЭ под действием электрического поля остаточной поляризации может быть представлена в виде:

$$\delta = d_{33} \cdot l_1 \cdot E_{ост} \qquad (2)$$

где l_1 – длина ТПЭ в недеформированном состоянии, $E_{ост}$ – напряженность поля остаточной поляризации пьезоматериала.

Тогда абсолютное значение продольной деформации ТПЭ может быть представлено в виде:

$$\Delta = d_{33} \cdot \frac{l_1}{l_2} U_{упр} \qquad (3)$$

где l_2 – толщина стенки трубки, $U_{упр}$ – приложенное напряжение управления. Очевидно, что при одном и том же напряжении $U_{упр}$ удлинение трубки будет тем больше, чем больше ее длина и чем меньше толщина ее стенки. Так как вектор поля остаточной поляризации имеет в ТПЭ радиальный характер, то механические деформации на участках, находящихся под диаметрально противоположными электродами, будут иметь противоположные знаки. В этом случае деформация ТПЭ приобретает изгибный характер. Величина углового отклонения конца ТПЭ от продольной оси определяется целым рядом факторов, не всегда поддающихся аналитическому описанию, среди которых следует упомянуть толщину, материал и способ нанесения электродов, жесткость закрепления ТПЭ в корпусе и т.д. В большинстве практических применений ТПЭ величина их изгибного отклонения от продольной оси при вышеуказанных размерах не превышает 3…5% от линейного размера при напряженностях приложенного электрического поля $E \le 5 \cdot 10^4$ В/м.

Для использования ТПЭ в качестве мультиплексирующего оптического коммутатора один из концов такого трубчатого пьезоэлемента 1 должен быть консольно закреплен в неподвижном основании 2, как показано на рисунке 2.

Второй конец остается свободным и направляется в сторону пучка выходных световодов, торцы которых, снабженные коллимирующими линзами (на рисунке не показаны), образуют двухкоординатную матрицу 3.

Так как ТПЭ является полым стержнем, то через его центральную ось может быть пропущено одномодовое оптическое волокно 4 без защитных покровов, по которому в коммутатор будут поступать входные оптические сигналы.

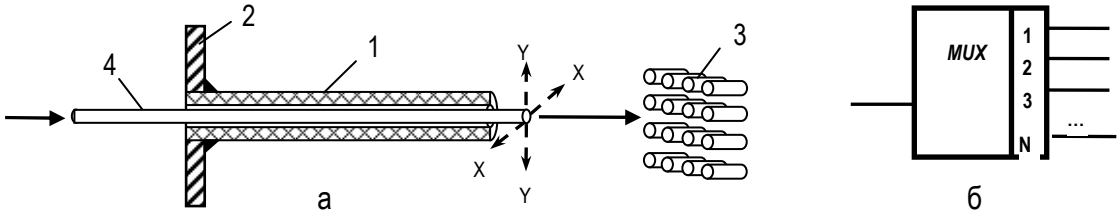


Рисунок 2. Конструкция оптического мультиплексора на основе ТПЭ (а) и его эквивалентная функциональная схема (б)

Подавая одновременно два управляющих напряжения $U_{упр}^{X}$ и $U_{упр}^{Y}$ на противоположные пары электродов ТПЭ, можно изгибать пьезоэлемент в двух координатах, осуществляя сканирование торца входного световода в плоскости выходных портов коммутатора, как показано на рисунке 2. Тем самым реализуется возможность пространственной коммутации оптического сигнала путем направления его с помощью ТПЭ в заданный порт. Очевидно, что данная конструкция на основе трубчатого пьезоэлектрического актуатора выполняет функцию мультиплексирующего (1xN) коммутатора оптических сигналов. Инвертируя оптические входы и выходы можно получить демультиплексирующий оптический коммутатор (Nx1) без изменения конструкции устройства.

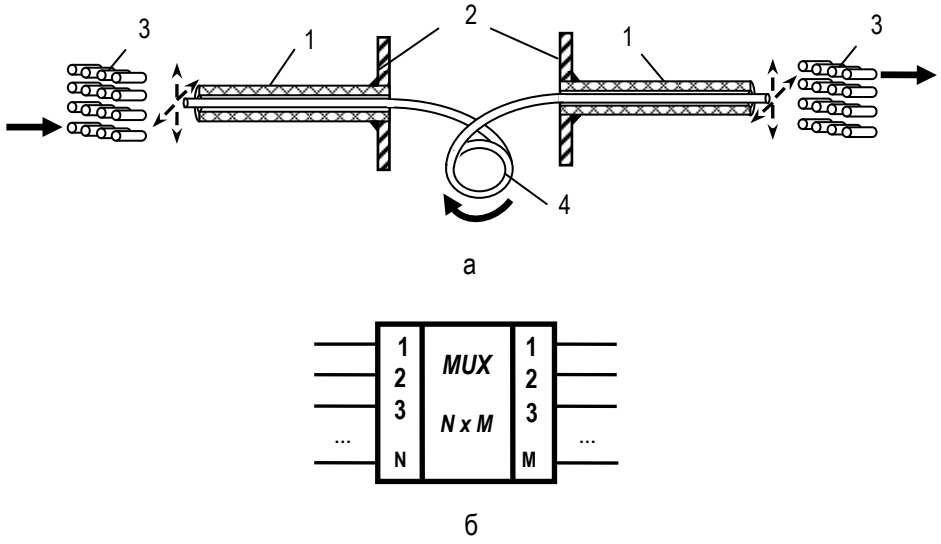


Рисунок 3. Конструкция полнофункционального оптоволоконного коммутатора на основе двух ТПЭ (а) и его эквивалентная функциональная схема (б)

Дальнейшее развитие предложенной конструкции позволяет реализовать полнофункциональный сканирующий оптический коммутатор (NxM), позволяющий обеспечить любой заданный алгоритм коммутации, на основе двух ТПЭ 1, закрепленных в основаниях 2, с отрезком световода 4, выполняющим функцию подвижного оптического патч-корда, как показано на рисунке 3.

Однако очевидным недостатком таких сборных конструкций является технологическая сложность в массовом производстве, обусловленная необходимостью обеспечения общей надежности и стабильности механических характеристик устройства в целом.

Гораздо эффективнее использовать в качестве сканирующего актуатора для оптического коммутатора не сборное изделие на основе ТПЭ, а монолитный пьезоэлемент из оптически прозрачного материала. Применяемые в технике кристаллы оптически чистого естественного кварца обладают великолепной прозрачностью, но весьма дороги. Поэтому, в настоящее время в электрооптических устройствах получили распространение такие искусственные прозрачные пьезоматериалы, как ниобат лития. Используя этот материал в качестве пьезоэлектрической основы актуатора можно получить новый тип пьезооптического сканирующего коммутатора, как предложено авторами в работе [Абрамов, 1984].

Основой такого монолитного сканирующего коммутатора является цилиндрический стержень 1 из прозрачного пьезоэлектрического материала, центрально закрепленный в неподвижном основании 2, как показано на рисунке 4.

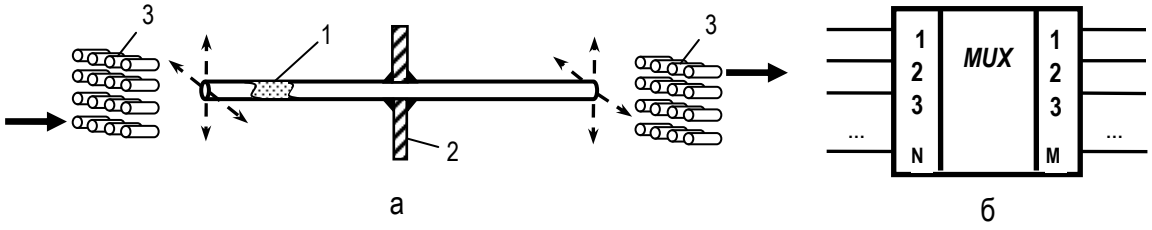


Рисунок 4. Конструкция полнофункционального оптоволоконного коммутатора на основе сканирующего пьезоэлемента из ниобата лития (а) и его эквивалентная функциональная схема (б)

Нанеся электроды на внешнюю поверхность прозрачного пьезоэлемента, и прикладывая к ним два управляющих напряжения $U_{упр}^{X}$ и $U_{упр}^{Y}$, можно осуществлять функцию пространственной оптической коммутации, аналогично устройствам на рисунках 2 и 3.

Заключение

Основными достоинствами сканирующего актуатора, выполненного из монолитного прозрачного пьезокристалла, являются простота, технологичность и надежность конструкции. Кроме того улучшаются точность и повторяемость позиционирования, временная и температурная стабильность параметров, уменьшаются значения гистерезиса и дрейфа вследствие остаточных пластических деформаций пьезоэлемента.

Уменьшение температурных флуктуаций угла отклонения актуатора обеспечивается выбором материалов с минимально возможным температурным коэффициент линейного расширения, который, в частности для пьезокерамики ЦТС-19, составляет $6\cdot10^{-6}$…$1{,}2\cdot10^{-7}$ K^{-1} [Джагупов, 1994]. Можно предложить также термостатирование всей конструкции коммутатора. Кроме того термостабильность актуатора можно повысить путем введения в конструкцию сканера или его держателя элементов, измеряющих и компенсирующих тепловые деформации, например как предложено авторами в работе [Джагупов Р.Г., 1991].

Быстродействие коммутатора в значительной мере зависит от фазочастотных характеристик актуатора. Консольное закрепление сканирующего пьезоэлемента предполагает наличие в характеристике резонансных пиков, что требует снижения рабочей частоты переключения коммутатора ниже частоты основного резонанса. Таким образом, быстродействие коммутатора будет определяться в основном линейными размерами актуатора и ограничиваться его низшей резонансной частотой.

Размеры актуатора определяют также долговременную стабильность параметров коммутатора. При больших (свыше $5\cdot10^{4}$ В/м) напряженностях приложенного электрического поля в пьезокерамике

появляются признаки переполяризации, что приводит к постепенному ухудшению точности позиционирования сканера из-за снижения коэффициента пьезочувствительности материала [Рябцов, 1994].

Простое снижение величины управляющего напряжения в данном случае неприемлемо, так как при этом уменьшается диапазон сканирования по обеим координатам, а, следовательно, и максимально достижимая размерность оптического коммутатора.

Решение этой проблемы, по мнению авторов, лежит в использовании сканирующих актуаторов, выполненных из недавно созданных композитных материалов, в частности, из неполярных полимеров на основе диблоков полистирена, в которых обратный пьезоэффект проявляется в десятки раз сильнее, чем во всех известных естественных пьезокристаллах и искусственных пьезокерамиках [Overton, 2011]. Это позволит в десятки раз снизить размеры актуаторов при сохранении прежних величин его угловых отклонений, тем самым на порядок увеличив быстродействие оптических коммутаторов.

Таким образом, сканирующие пьезоэлектрические актуаторы можно с успехом применить для создания надежных и долговечных оптических коммутационных устройств в полностью оптических сетях.

Библиография

[Dooyoung, 2004] Dooyoung H. et al. Low-Voltage, Large-Scan Angle MEMS Analog Micromirror Arrays with Hidden Vertical Comb-Drive Actuators // Journal of microelectromechanical systems, vol. 13. - №. 2. - 2004

[Little, 2001] M. Little, B. Jamison. All-optical MEMS continue to evolve //Solus Micro Technologies. - July. - 2001 - Интернет-ресурс: http://www.lightwaveonline.com/articles/print/volume-18/issue-8/features/all-optical-mems-continue-to-evolve-53471737.html.

[Overton, 2011] Gail Overton. New polymer from ORNL has 10 times the piezoelectric effect of crystals and ceramics. - Laser Focus World. – November 2011 - Volume 47 - Issue 11.

[Абрамов, 1984] Абрамов Г. А., Джагупов Р.Г., Ковригин В.А., Рябцов А.В., Якубов Р.А. Пьезооптическое сканирующее устройство. -Авторское свидетельство СССР № 1136636.- ДСП.- Зарегистрировано в Госреестре изобретений 22.09.84.

[Гайворонская, 2010] Гайворонская Г.С., Рябцов А.В. Методы и средства коммутации оптических сигналов в информационных сетях. Холодильна техніка і технологія. - Одеса: ОГАХ. - 2010. - №2 (124).

[Гайворонская, 2011] Гайворонская Г.С., Рябцов А.В. Особенности применения оптических коммутаторов в современных информационных сетях // Applicable Information Models. – Sofia: ITHEA, 2011. – № 22. – P. 169-181.

[Гринфильд, 2002] Гринфильд Д. Оптические сети. - К. - ТиД "ДС". - 2002. - 256 с.

[Джагупов, 1994] Пьезоэлектронные устройства вычислительной техники, систем контроля и управления: Справочник / Р.Г. Джагупов, А.А. Ерофеев. - СПб. - Политехника. - 1994.

[Неволин, 1996] Неволин В.К. Основы туннельно-зондовой нанотехнологии. - М.: МГИЭТ (ТУ). - 1996.

[Рябцов, 1994] Джагупов Р.Г., Рябцов А.В. Устройство для измерения перемещений.- Авторское свидетельство СССР № 1645807. - Опубл.30.04.91. - БИ № 16.

[Скляров, 2004] Скляров О.К. Волоконно-оптические сети и системы связи. - М. - Солон-Пресс. - 2004. — 272с.

[Хазе, 1975] Хаазе Г.И. Современные электропроигрыватели. - М. - Энергия. - 1975.

Информация об авторах

Рябцов Александр Васильевич *– Факультет Информационных технологий и кибербезопасности ОНАПТ, к.т.н., доцент кафедры информационно-коммуникационных технологий; Украина, Одесса, 65026, ул. Дворянская, 1/3; тел. (048)-720-91-48; e-mail: ryabtsov@usa.com*

Главные области научного исследования: информационные системы, коммутационная техника, оптические технологии, полностью оптические коммутаторы.

TABLE OF CONTENT