ITHEA

International Journal

INFORMATION THEORIES & APPLICATIONS

2011 Volume 18 Number 4

International Journal
INFORMATION THEORIES & APPLICATIONS
Volume 18 / 2011, Number 4

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 as well as the **subscription fees** are given on *www.foibg.com/ijita*.

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. 18, Number 4, 2011

Printed in Bulgaria

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Institute of Mathematics and Informatics, BAS, Bulgaria,
Universidad Politécnica de Madrid, Spain.

Publisher: **ITHEA**
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, www.foibg.com, e-mail: info@foibg.com

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

FUZZY SETS AS A MEAN FOR UNCERTAINTY HANDLING: MATH, APPLIED MATH, HEURISTICS

Volodymyr Donchenko

Abstract*: Number of Disciplines and Theories changed their status from status of Natural Science discipline to Mathematics. The Theory of Probability is the classical example of that kind. The main privilege of the new, Math, status is the conception of Math truth, which distinguish Math from other theories. Some disciplines, used in Applications, pretended to be Math, not being such. It's entirely true for Fuzzy Subsets Theory with its pretension to be Math and to be exclusive tools in uncertainty handling. Fundamental pretensions of classical Fuzzy subset theory including pretension to be math is discussed as well as the some gaps of the theory are discussed in the article. Statistical interpretation of membership functions is proposed. It is proved, that such the interpretation take place for practically all supporters with minimal constraint on it. Namely, a supporter must be the space with a measure. The interpretation proposed makes it clear the modifying of the classical fuzzy object to correct the gaps. It becomes possible to say about observations of fuzzy subset within the conception of modification and to extend Likelihood method on the new area. Fuzzy likelihood equation adduced as example of new possibilities within approach proposed. One more interpretation for the Fuzzy subset theory is proposed to discuss: multiset theory one.*

Keywords*: f Uncertainty, Plural model of uncertainty, Fuzzy subsets Theory, statistical interpretation of the membership function, modification of Fuzzy subsets, Fuzzy likelihood equation, Multiset theory.*

ACM Classification Keywords: *G.2.m. Discrete mathematics: miscellaneous,G.2.1 Combinatorics. G.3 Probability and statistics, G.1.6. Numerical analysis I.5.1.Pattern Recognition: Models Fuzzy sets; H.1.m. Models and Principles: miscellaneous:*

Introduction

Initially the conception of the Fuzziness planned to be the object of the proposed article. But it became clear after a while that the point of issue ought to be wider. Such expansion must first of all include the discussion about the role of the Fuzziness within the conception of uncertainty. What is the "uncertainty" by itself? Is it mathematics? If not, where ought one to look for the origin of the conception? How does the uncertainty sort with the Fuzziness?

What is primary between these two? There are some more pertinent questions related to place of Math and Applied Math in definitions and applications of the uncertainty and Fuzziness as well as to the role of the Heuristics in Applied researches. Uncertainty surely is the first in discussing about priority within the mentioned pair. For example, Pospelov and his school [Поспелов, 2001] share this opinion. They consider the Fuzziness to be the mean for the uncertainty handling but not vice versa. As to Math, Applied Math it is worth while mentioning in this connection that the Fuzzy subset theory (FzTh) coming into the world due to Lotfi Zadeh [Zadeh, 1965] (see also [Kaufmann, 1982]) was proclaimed to be the mathematical panacea for uncertainty modelling.

Mathematics

There are some principal consideration determined the relations between Math and Empiric experience.

As to the Mathematics by itself. For example Wikipedia [Wikipedia, Math] states the next: "Mathematics is the study of quantity, structure, space, and change. Mathematicians seek out patterns, formulate new conjectures, and establish truth by rigorous deduction from appropriately chosen axioms and definitions." Thus, the specific objects (Math structures) and conception of Math truth (rigorous deduction) are the essence of the Mathematics.

As to the Math structures (see? For example [Донченко, 2009]). When saying "math structure" we ought to understand it to be a set plus "bonds", "relations" between elements of the set. Correspondent "bonds" or "relations" in Math specified by: 1) Math relations (for example "$\leq$" in R^1); 2) functions; 3) operations (for example "+", "$\cdot$" in R^1); 4) collections of the subsets (for example, collection of open subsets, or collection closed subsets or collection of neighbours in R^1); 5) combinations of the four previous. All the Math structures initially have been established for the sets of a numbers of different kinds: integers, real, complex. Then they have been extended on abstract sets. So we have now, for example a structure of metric space (an abstract set plus real valued nonnegative function of two argument with certain properties), structure of group, including affine one (an abstract set plus binary operation with certain properties); structure of linear space (an abstract set with the structure of affine group plus product operations for each of real numbers); structure of Euclidean and Hilbert space (structure of linear space plus non- negative real valued function of two arguments: scalar product); topological space (an abstract set plus an appropriate collection of it subsets, which made possible to define the limit), measurable space (an abstract set plus collection of it subspace, named by σ-algebra), linear topological space and so on. More detailed structure may be considered within the base structure: linear subspace or hyper plane within linear or Euclidean space, subgroup within the group and so on.

Math truth

The fundament of the Math truth is the conception of deducibility. It means that the status of truth (proved statement) has the statement which is terminal in the specially constructed sequence of statements, which called its proof. The peculiarity in sequence constructing means, that a next one in it produced by previous by special admissible rules (deduction rules) from initial admissible statements (axioms and premises of a theorem). As a rule, corresponded admissible statements have the form of equations with the formulas in both its sides. So, each next statement in the sequence-proof of the terminal statement is produced by previous member of sequence (equation) by changing some part of formulas in left or right it side on another: from another side of equations-axioms or equations premises. The specification of the restrictions on admissible statements and the deduction rules are the object of math logic.

Applied Mathematics

The main aim of Applied Math (AppMa) is the description the real object under consideration by Math. This means that the object as "a structure" is represented by the means of Math structure. I.e. the main parts of the object under modelling and principal bonds, ties, relation between them are represented be the means of Math structurization. It is necessary condition for the math modelling to have apt interpretation for a correspondent Math objects and objects under observation. Such interpretations for example, for the function and its derivatives are correspondingly the path and speed for. Integration and differentiation are the means to represent the relation between speed and path. Likewise, a frequencies that or those groups of the result in a sequence of observations are interpreted as a probabilities and vice versa. Surely that or those interpretations can be applied under certain restrictions. So we can't investigate discrete systems by the means of differential equations or apply probabilistic method out of fulfilling stability frequencies low. The main aim of the Math description, the Math modelling of the real object, is to take the advantages establishing true statements for the apt Math object(target statement), which represents real object: for its Math model, with the following interpretation of the correspondent statements. So, if the model of the real object is equation, the target math statement is the statement about its decision and following interpretation of the decision for the real object. So, the next tree-step procedure is the essence of the AppMa. 1. Math decrypting the object on the base of the available knowledge about the real object under consideration and with the help of an apt interpretation. 2. Establishing of math true for the apt (target) statements within the Math model. 3) Interpretation of the target math statement for the real object. The first and the last steps are impossible without interpretation. The availability of interpretation is principal for applied math. Thus, interpretation plus math rigorous truth is the essence of the applied math. So, for example, numerology is not Math and AppMa, because it does not appeal to target (Math true) statements.

Heuristics

There some more means for the investigation, which uses Math on that or this stage or investigation but do not satisfy three stage procedure of AppMa. It may be designated by "intellectual calculus" as some people do. But it is reasonable on my mind to use an apt old good word "heuristic". Indeed, by [Wikipedia: heuristics] "heuristic or heuristics (from the Greek "Εὑρίσκω" for "find" or "discover") refers to experience-based techniques for problem solving, learning, and discovery. Heuristic methods are used to speed up the process of finding a good enough solution, where an exhaustive search is impractical. Examples of this method include using a "rule of thumb", an educated guess, an intuitive judgment, or common sense". We mention opportunely the authors of the "heuristic" from Polya [Polya,1945] through A.Newell&J.C. Shaw& H.A.Simon[Newell& Shaw& Simon,1962] to D. Kahneman [Wikipedia: Kahneman].

Uncertainty

It is common place for the investigators to say or to use the expression "modelling under uncertainties". It is also generally recognized that theory of probability is the classical mean for uncertainty handling when such uncertainty is shown as randomness. Determination of randomness appeals to the notion of experiment (observation, trail, test, sometime – stochastic experiment). So, understanding what the randomness is makes it necessary to look into the conception of "experiment".

Experiment

As the analysis of the numerous sources on Theory of Probability and Math Statistics [Донченко 2009], notion of experiment in them is associated with something, named conditions (condition of experiment), under which phenomena is investigated, and something, that appears under the conditions: named the results of experiment.

So, as in [Донченко 2009] "experiment" is proposed to be considered the pair *(c, y)*: *c*- conditions of experiment (observation, trail, test), y – result of experiment. Henceforth Y_c for the fixed condition *c* will denote the set of all possible that may appear in the experiments under conditions $c \in C$. Generally speaking Y_c is not singleton.

It is reasonable to mark out in a condition c variational, controlled, part x: $x \in R^p$ as a rule, and part f, which is invariable by default in a sequence of experiment. Condition c under such approach is denoted be the pair: c=(x, f), $x \in X \subseteq R^p$.

Sequence of Experiment and their registration

If there are n experiments, then their registration is the sequence

$$(c_i, y_i), c_i \in C, y_i \in Y_{c_i}, i = \overline{1,n}. \quad (1)$$

Different variants of (1) can be implemented in practice

$$((x_i, f), y_i), x_i \in X \subseteq R^p, y_i \in Y_{x_i}, i = \overline{1,n}, \quad (2)$$

$$(x_i, y_i), x_i \in X \subseteq R^p, y_i \in Y_{x_i}, i = \overline{1,n}, \quad (3)$$

$$y_i, y_i \in Y_{c_i}, i = \overline{1,n}, \quad (4)$$

$$y_i, y_i \in Y, i = \overline{1,n} \quad (5)$$

It is obvious, that (5) is equivalent to (1) when all conditions are the same:

$$c_i \equiv c, i = \overline{1,n}.$$

But if it is not so , then

$$Y = \bigcup_{i=1}^{n} Y_{c_i} \neq \text{Singleton}$$

Randomness as a classical example of Uncertainty

Randomness in introduced above designation means firstly, that the results of experiment do not determined by conditions $c \in C$ definitely, i.e.

$$Y_c \neq \text{Singleton}, c \in C. \quad (5)$$

And, secondly, the observations satisfy stability frequency low. This mean: 1) in a sequence of experiment with fixed conditions c frequency of each collection of possible results from Y_c turn to some limit value; 2) the limit value does not depend from the sequence of observations and characterize the phenomenon under consideration. In the Theory of Probability Y_c is called Space of elementary events and is denoted by Ω. Correspond experiment is often called stochastic experiment.

Plural model of Uncertainty

As the randomness is special kind of uncertainty, so the definition of uncertainty one ought to look for in the conception of experiment. Then the natural definition of uncertainty is coincides with the first part of randomness and described by equity (5). Thus, uncertainty is defined on the base of the experiment and classifies certain relation between conditions of experiments *c* and correspond results *y* of it. This relation is stated in (5). We will name such conception of uncertainty by Plural Model of Uncertainty (PluMoU).

Mathematical means for uncertainty handling

There are comparatively few math tools for uncertainty handling. Having no possibility to discuss the theme in detail, we note, that these, namely, are: 1)Theory of probability; 2)inverse problem; 3) maxmin method; 4) Hough Transform; 5) Multisets Theory; 6) Fuzzy(?) Theory; 7) combination of 1)-6) issues. The last point needs some additional explanation in order to embed FzTh in PluMoU. Such embedding become feasible on the base of two possible interpretation of FzTh: within Theory of Probability and Multisets theory.

Fuzzy Theory and statistical interpretation of membership function

Fuzzy set $\underline{A}$, subset to be more precise (Kaufmann, 1982), as the object in mathematics is nothing more, but the graphic of real valued function μ on an abstract crisp (usual) set *E* (henceforth - supporter of the Fuzzy subset). There is additional constraint on the value of this function, named membership function in Fuzzy theory: its values are bounded by the segment $[0,1]$:

$$\mu : E \to [0,1] , \underline{A} = \{(e, \mu_{\underline{A}}(e)) : \ e \in E)\} .$$

There are no objections. The definition is perfect but trivial. There great many functions in mathematics, there great many graphics and there are no pretensions of the Fuzzy theory.

Some lacks of the Fuzzy Theory

As it was mentioned about there are several Math tools for uncertainty handling. All of them are well-grounded Math. So, FzTh is not exclusive in pretension on uncertainty handling Also the attention was drawn earlier to the importance of the interpretation for the Applied Math unlike from fundamental. As to FzTh the lack objective interpretation is rather painful problem. The absence of its own set theory as well as a Fuzzy logic is the problem waiting for its solutions. There some steps relating logic (see, for example, [Hajek, 1998F], [Hajek, 1998]). But the problem of interpretation in this case must be solved also. The importance of apt interpretation may be brightly demonstrated on history of the modal logics.

There nothing like axiomatic set theory in FzTh even in naive, Kantor's sense. Particularly, such axiom of paramount importance, known as abstraction [Stoll, 1960] or separation [Kuratovski, Mostowski,1967] principle, is out of consideration. Implementation a variant of this axiom in FzTh would help to close the "object" problem. Indeed, as is well known, the axiom under consideration establishes the correspondence between classical (crisp) subsets and the properties of the elements of the universal set – namely, predicates on the universal crisp set. So, classical predicate have its object of characterization: the correspond set, determined by abstraction axiom. In FzTh changing binary predicates by membership functions forgot to define another element in the pair (predicate, set). Consequence the object of fuzzy characterization was lost. By the way, Multiset theory (some words below) with its technique could help in solving this problem.

It is interesting, that in obvious examples of membership functions out of the FzTh such objects are the intrinsic to the definition of the correspondent objects. Namely, such examples are the generalized variants of logit - and probit (GeLoPr) – regressions, transition matrix for the Markov's chains and Bayesian nets are the mentioned examples.

Natural examples of membership function: Generalized variants of logit - and probit regression

As to these examples, then GeLoPr describes the dependence of the frequencies (probabilities) of the certain event A from the real valued vector under certain parameterization:

$$P\{A \mid H_x\} = G(\beta^T \begin{pmatrix} 1 \\ x \end{pmatrix}),$$

$$\beta \in R^{n-1}, \beta^T = (\beta_{0,}..., \beta_{n-1,}), x \in R^n,$$

where G – distribution function F(z), $x \in R^1$ or correspond tail: 1-F(z) for the scalar distribution.

In this example GeLoPr $\mu(x)=P\{A \mid H_x\}, x \in R^{n-1} = E$ as a function of $x \in R^{n-1}$ is a membership function in the classical FzTh, which corresponds to the certain object, intrinsic for the theory: event A. We would remind, that the event A, mentioned above, describe the presence of certain property in an observation $(x, y), y \in \{0,1\}$. The value 1 for y means the fulfilling and 0 - not fulfilling the property in the observation.

Natural examples of membership function: Markov chain

A transition matrix for the Markov's chain $(\xi_n, n \in \mathbb{N})$, with states set $\wp = \{S_1, ..., S_M(...)\}$ is the $M \times M$ matrix $P = (p_{ij})$ of conditional probabilities:

$$p_{ij} = P\{\xi_{n+1} = S_j \mid \xi_n = S_i\}, i, j = \overline{1, M}$$

Each column with number $j = \overline{1, M}$ of the matrix defines membership function $\mu_j, j = \overline{1, M}$ on $E = \wp$:

$$\mu_j(S_i) = p_{ij} = P\{\xi_{n+1} = S_j \mid \xi_n = S_i\}, j = \overline{1, M}, \quad (6)$$

$$S_i \in \wp = E$$

In each of the M membership functions $\mu_j(S), S \in \wp = E, j = \overline{1, M}$ there are intrinsic objects of fuzzy characterization. Namely these are, correspondingly, $\{\xi_{n+1} = S_j\}, j = \overline{1, M}$.

It is interesting, that it is naturally to consider a (6) to be a "full system" of membership functions: a collection of functions $\mu_j, j = \overline{1, M}$ on E for which

for any $e \in E$

$$\sum_{j=1}^{M} \mu_j(e) = 1, e \in E$$

Natural examples of membership function: Bayesian nets

Any Bayesian net is in the essence a directed weighted graph associated with the probabilistic objects. But if in classic probabilistic graph the weights prescribed to the edges with one and the same head-nodes, in Bayesian – to the one with the same tail-nodes. Thus, the collection of the probabilities is associated with each node: the probabilities, which weight the nodes predecessors. So, correspond probabilities (conditional by its nature) define a membership function.

Probabilistic Interpretation membership function

This subsection deals with the probabilistic interpretation for the classical variant of the FzTh (Donchenko, 1998, 3). Two variant of a supporter E are considered below: discrete and non-discrete. Discrete case is the one which fully illustrates the situation. Namely, each membership function of a fuzzy subset is represented by a system of conditional probabilities of a certain events relatively complete collections of the sets $\mathrm{H_e}, \mathrm{e} \in \mathrm{E}$. Saying "complete collection" we consider the collection $\mathrm{H_e}, \mathrm{e} \in \mathrm{E}$ be the partition of the space of elementary events Ω for a basic probability space

Probabilistic Interpretation membership function: discrete supporter

The main result of the subsection is represented by theorem 1[Donchenko, 1998, 3].

Theorem 1. For any classical Fuzzy Set $(\mathrm{E}, \mu_{\underline{\mathrm{A}}}(\mathrm{e}))$ with discrete support E there exists such discrete probability space

$$(\Omega, \mathrm{B}_\Omega, \mathrm{P}),$$

event

$$\mathrm{A} \in \mathrm{B}_\Omega$$

and complete collection of events

$$\mathrm{H_e} : \mathrm{H_e} \in \mathrm{B}_\Omega, \mathrm{e} \in \mathrm{E}$$

within this probability space such that membership function $\mu_{\underline{A}}(e)$ is represented by the system of conditional probabilities in the next form:

$$\mu_{\underline{A}}(e) = P(A \mid H_e), e \in E. \qquad (7)$$

Theorem 2. For any complete collection of Fuzzy subsets $(\mathrm{E}, \mu_{\mathrm{A_i}}(\mathrm{e})), \mathrm{i} = \overline{1, \mathrm{n}}$, with the one and the same supporter E there exists:

discrete probability space $(\Omega, \mathrm{B}_\Omega, \mathrm{P})$;

collection of the evens $\mathrm{A_i} \in \mathrm{B}_\Omega, \mathrm{i} = \overline{1, \mathrm{n}}$;

complete collection of the events $\mathrm{H_e} : \mathrm{H_e} \in \mathrm{B}_\Omega, \mathrm{e} \in \mathrm{E}$,within the probability space $(\Omega, \mathrm{B}_\Omega, \mathrm{P})$,

such, that all of the membership functions $\mu_{A_i}(e), e \in E, i = \overline{1,n}$, are simultaneously represented as the systems of conditional probabilities in the next way:

$$\mu_{\underline{A_i}}(e) = P(A_i \mid H_e), e \in E, i = \overline{1,n}.$$

Probabilistic Interpretation membership function: non discrete supporter

The result of the previous subsection may be extended noticeably to non-discrete case if the supporter E possesses certain structure, namely, if it is the space with a measure [Donchenko, 1998, 3].

Theorem 3. Given the:

$(E,\Im,m)$ - is the space with a measure;

$(E, \mu_{\underline{A_i}}(e)), i = \overline{1,n}$, is the complete collection on Fuzzy subsets with the same supporter E;

all of the membership functions $\mu^{(A_i)}(e), i = \overline{1,n}$, are $\Im,£,$ - measurable ($£$ – Borel σ-algebra on R^1),

then, there exist:

probability space (Ω, B_Ω, P),

ξ - discrete random S_p – valued random variable on (Ω, B_Ω, P), where S_p is any n -element set with the elements, say, $S_i, i = \overline{1,n}$;

η random E – valued random variable on (Ω, B_Ω, P) such, that for any $i = \overline{1,n}$

$$\mu^{(A_i)}(e) = P\{\xi = S_i \mid \eta = e\},$$

where

$$P\{\xi = S_i \mid \eta\}$$

– conditional distribution of random variable(r.v) ξ respectively r.v. η.

The conditional distribution is regular: for any $e \in E$ $P\{B \mid \eta = e\}$ is a probability respectively B.

Remark on proof. The proof is the result of extending the ideas of the previous theorems but embodied by the application another technique: technique of conditional distribution. The proof being technically complicated is omitted.

Remark 1. There are obvious objects of uncertainty characterization within the theorems 1-3.

Modified Definition of Fuzzy Sets

The way for the solving the problem of constructing the analogue of the separation principle may be on the author opinion the straight reference on the object or property described uncertainly. This reference ought to be reflected evidently in the definition of the membership function:

$$\mu^{(T)}(e), e \in E,$$

where T – correspondent property (predicate) on certain set U. The last is the set of "uncertain characterization". It may coincides with the E. So $\mu^{\{T\}}(e)$ would be "uncertain characterization"of the property T or corresponding crisp subset $P_T \subseteq U$. The last transition is possible due the separation principle for the crisp sets. Two membership functions $\mu^{\{T_1\}}(e)$ and $\mu^{\{T_2\}}(e)$ with $T_1 \neq T_2$, would specify two different Fuzzy sets, even if they are equal as the function of $e, e \in E$.

Definition. The pair

$$(E, \mu^{(T)}(e))$$

or

$$(E, \mu^{(P_T)}(e))$$

is called the modified Fuzzy subset (MoF) with E as a supporter, which uncertainly describe crisp T on U (or correspondent crisp subset, $P_T \subseteq U$, where U- the "universal" crisp set of "uncertain characterization"), if :

E – is the abstract crisp set, which is referenced to as a supporter;

T - is a crisp predicate onU , correspondingly, P_T - crisp subset ofU, which corresponds to T;

$\mu^{(T)}(e) \in [0,1]$ – function of two arguments: $e, e \in E$ and T from the set of all crisp predicates on universal crisp U.

The function

$$\mu^{(T)}(e), e \in E$$

just as in classical theory of Fuzzy sets will be referenced to as membership function, with adding that it characterizes uncertainly property T (or correspondent subset P_T).

Remark 2. Obviously, statistical interpretation of the theorems 1-3 is applicable to MoF.

Observations of the Modified Fuzzy Sets

The modification of the definition of Fuzzy set introduced earlier in the paper imparts the objectivity to the Fuzzy sets and it is possible now to say about observations of Fuzzy sets for modified ones (Donchenko, 2004). It's very important ontological aspect for mathematical modeling using Fuzzy sets. The observation of modified Fuzzy sets is the pair $(e, T(e)) - e, e \in E$ – element from the supporter and T(e) is the predicate value on this element. Namely, e is the element, displayed in observation and $T(e)$ is the fixed information about fulfilling the property T in the observation, specified by $e \in E$. It is just in such a way the observations are interpreting in the logit- and probit – regressions and in its generalizations.

So the observation sample is $(e_i, t_i), t_i = T(e_i), i = \overline{1,n}$. One can say about independent observation within statistical interpretation.

Likelihood method for the Modified Fuzzy Sets

Statistical interpretation a membership function grant to say about extension of statistical MLM for estimating fuzzy parameter just as it takes place in the regressions mentioned above.

Indeed, let

$$\mu^{(T)}(e), e \in E$$

-MoF with membership function from parametric collection of membership functions

$$\mu^{(T)}(e) = \mu(e, \beta), \beta \in R^p .$$

Let $(e_i, t_i), i = \overline{1,n}$ independent observation of MoF. We determine "Fuzzy Likelihood function" $FL(\beta)$) by the relation

$$FL(\beta) = \prod_{i=1}^{n} \mu^{t_i}(e_i, \beta)(1 - \mu(e_i, \beta))^{1-t_i} .$$

Correspondingly, we denote by

$$fl(\beta) = \ln FL(\beta) = \sum_{i=1}^{n} t_i \ln \mu(e_i, \beta) +$$

$$+\sum_{i=1}^{n} (1 - t_i) \ln(1 - \mu(e_i, \beta))$$

- logarithmic "Fuzzy Likelihood function".

Just as it is in statistic likelihood estimation

$$\hat{\mu}^{(T)}(e) = \mu(e, \hat{\beta}),$$

where

$$\hat{\beta} = \arg \max_{\beta \in R^p} FL(\beta) .$$

Just as in Statistics if $\mu^{(T)}(e) = \mu(e, \beta)$ necessary conditions is the

$$\frac{\partial FL(\beta)}{\partial \beta} = 0$$

or

$$\frac{\partial \mathrm{fl}(\beta)}{\partial \beta} = 0 .$$

The last equation is equivalent the fist one under additional restriction, that the set of zeroes of $\mu(\mathrm{e}, \beta), \beta \in \mathrm{R}^{\mathrm{p}}$ respectively $\beta \in \mathrm{R}^{\mathrm{p}}$ is the same for all $\mathrm{e} \in \mathrm{E}$.

The equations of necessary conditions it is naturally to reference to as "fuzzy likelihood equations".

Theorem 4. Under all necessary restrictions "fuzzy likelihood equations" are of the next form

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{\mathrm{t_i} - \mu(\mathrm{e_i}, \beta)}{\mu(\mathrm{e_i}, \beta)((1 - \mu(\mathrm{e_i}, \beta)))} \frac{\partial \mu(\mathrm{e_i}, \beta)}{\partial \beta_{\mathrm{j}}} = 0,$$

$$\mathrm{j} = \overline{\mathrm{l,p}}, \beta = \begin{pmatrix} \beta_{\mathrm{l}} \\ \cdots \\ \beta_{\mathrm{p}} \end{pmatrix} \in \mathrm{R}^{\mathrm{p}} .$$

Experts estimating can be used too by combining LSM and MLM.

Multisets Theory

Multisets (see,for example, reviews: [Blizard, 1989], [Буй, Богатирьова 2010]), is the Math answer for necessity to describe sets which elements with may "repeat". Thus originally conception of multiset implement the idea of repetition rep(u)for elements u from subset D of certain universal set U. Which are the sets D and U , and, correspondingly, rep(u) , depends on peculiarities of applied problem. So, for example, D can be a set of the answers for this or that call in the Internet, відповідей and rep(u) – number of repetition for each record. There is natural way to implement the idea of repetition: to provide each $\mathrm{u} \in \mathrm{D}$ with number or repetition $\mathrm{n_u} : \mathrm{n_u} \in \{1, 2, \ldots, \mathrm{n}, \ldots\} \equiv \mathrm{N}^{+}$.

So, we got the first variant for multiset determining 1.We will call by multiset the set of the pairs $\bigcap_{u \in D} \{(u, n_u)\}, n_u \in N^{+}, u \in D \subseteq U$ for any subset D of certain universal set U . We will call D to be the

base of multiset and n_u -multiplicity or repetition factor. This terms will use in all variants of multiset definitins below in evident way. We will denote multiset with base D by $D^{(ms)}$.

Thus, multiset $D^{(ms)}$ is the usual set D with "comments" n_u to its elements.

2. Within the frame of the second definition multiset for any subset D of certain universal set U is the transformation $\alpha : D \to N^+$, defined for any $u \in D$ (see, for example, [Петровский, 2002], [Редько, 2001]) . Equivalence of the first and second determination is evident: $\alpha(u) = n_u, u \in U$. One ought to remark that in second variant the relation function substitutes the set.

3. Third variant: $D^{(ms)}$ for $D \subseteq U$ is the pair $D^{(ms)} \equiv (D, \alpha) : \forall D \subseteq U, \forall \alpha : D \to N^+$, α is defined on all elements of D. Thus, in this variant multiset is the pair: set D –"comment" α .

When it necessary we will refer on the components of the multiset-pair $D_{ms} = (D, \alpha)$ n evident way correspondingly by D_α ,and α_D as well as by $D_{D^{(ms)}}, \alpha_{D^{(ms)}} : D = D_{D^{(ms)}}, \alpha_{D^{(ms)}}(u) = \alpha(u), u \in U$.

Natural set terminology take place for the multisets: for the standard operations ("$\cup$","$\cap$") and for standard relation: "$\subseteq$" . We will denote them for multiset correspondingly "$\cup_{ms}$ ", "$\cap_{ms}$ "" $\subseteq_{ms}$ " .

We will define them

$$\forall D_1^{(ms)} = (D_1, \alpha_1), D_2^{(ms)} = (D_2, \alpha_2) : D_i \subseteq U, i = 1, 2$$

by the relations, correspondingly:

1. $$D_1^{(ms)} \subseteq D_2^{(ms)} \Leftrightarrow (D_1 \subseteq D_2 \ \& \ \alpha_1 \le \alpha_2)$$

2. $$D_1^{(ms)} \bigcup_{ms} D_2^{(ms)} \equiv \ (D_1 \bigcup D_2, \max(\alpha_1, \alpha_2))$$

3. $$D_1^{(ms)}{}_{ms} \bigcap D_2^{(ms)} \equiv \ (D_1 \bigcap D_2, \min(\alpha_1, \alpha_2))$$

As to operation "$\bar{}$", then it is necessary to "cut" N^+ to $N_M^+ = \{1,2,\ldots,M\}$ leaving all the rest of the determinations unchangeable. Then $\overline{D^{(ms)}} = \overline{(D,\alpha)}$ is determined by the relation

$$\overline{D^{(ms)}} = (D, M - \alpha)$$

Characteristic function $\chi_{D^{(ms)}}(u)$ (see, for example, [Buy, Bogatyreva, 2010]) is convenient in multiset handling. It is determined by the relation

$$\chi_{D^{(ms)}}(u) = \begin{cases} \alpha(u), u \in D \\ 0, \quad u \notin D \end{cases}$$

Namely, characteristic function is extension of repetition factor or multiplicity on the universal set U.

The role of characteristic functions in multiset theory is fixed by the equivalency in the determination of set operations and order described by the next relations

1. $(D_1^{(ms)} \subseteq D_2^{(ms)}) \Leftrightarrow (\chi_{D_1^{(ms)}} \leq \chi_{D_2^{(ms)}})$

2. $\chi_{D_1^{(ms)} \cup D_2^{(ms)}} = \max(\chi_{D_1^{(ms)}}, \chi_{D_2^{(ms)}})$,

3. $\chi_{D_1^{(ms)} \cap D_2^{(ms)}} = \min(\chi_{D_1^{(ms)}}, \chi_{D_2^{(ms)}})$.

Multisets Theory and Fuzziness

It is evidently that in multiset theory repetition factor is the "absolute" variant of membership function. Saying so, we mean absolute and relative frequency. Even more, in the variant of using N_M^+ we get pure membership function by dividing repetition factor α by M. But there are essential differences between these two theories: all membership functions in FzTh are referenced to one and the same E (U in the designations of the multiset theory) and are referenced to particular $D \subseteq U$ in multiset theory. Simple substitution: subsets D instead one and the same universal set in Fzth solve the problem of the object characterization: D is the object. All the rest lacks of the Fzth are also immediately solved the problems: 1)of own set theory with correspond set operations and order; 2) own logic: commonly used mathematical logic; 3) interpretation ($\alpha(u)$ as rep(u)); 4) abstraction axiom: for each $D \subseteq U$ there many possible correspondent α : any of them.

Conclusion

General approach to describing uncertainty was expounded in the paper within conception plurality in understanding uncertainty. The uncertainty is the quality of interaction between researcher and phenomenon within an observation (experiment, trial, and test). Obviously, some formalization for the "observation" is proposed and discussed in the text. The conception of uncertainty proposed make it possible to give for all math means used for uncertainty handling. It is entirely true for then Fuzzy approach after proving principal theorems about statistical interpretation of membership function. Some lacks of the Fuzzy Theory were discussed and some examples and directions of its overcoming were demonstrated. Namely, these were modification, proposed for the membership function and Multiset Theory.

Bibliography

[Blizzard,1989] Blizzard W.D. The Development of Multiset Theory. In *Notre Dame Journal of Formal Logic. - Vol.30, No.1. -* 1989.-*P. 36-66*.

[Buy, Bogatyreva, 2010] Buy D., Bogatyreva Ju. Multiset Bibliography. In *Papers of 9th International Conference on Applied Mathematics, February 2-5,2010.– Bratislava.- 2010.- P.407 -413.*

[Donchenko, 1998,3] Donchenko V. Conditional distributions and Fuzzy sets. In *Bulletin of Kiev University. Series: Physics and Mathematics, №3, 1998 (In Ukrainian)*.

[Donchenko, 1998,4] Donchenko V. Probability and Fuzzy sets. In *Bulletin of Kiev University. Series: Physics and Mathematics, №4,1998. (In Ukrainian)*

[Donchenko, 2004.] Donchenko V. Statistical models of observations and Fuzzy sets. In *Bulletin of Kiev University. Series: Physics and Mathematics, №1, 2004(In Ukrainian)*.

[Донченко, 2009.] Донченко В.С., 2009. Неопределённость и математические структуры в прикладных исследованиях (Uncertainty and math structures in applied investigations) *Human aspects of Artificial Intelligence* International Book Series Information science & Computing.– Number 12.– Supplement to International Journal "Information technologies and Knowledge". –Volume 3.–2009. – P. 9-18. (*In Russian*)

[Hajek, 1998F] Hajek P. Ten questions and one problem on fuzzy logic. In *Preprint submitted to Elsevier Science.- February 1998.- 10 p*.

[Hajek, 1998] Hajek P. *Metamathematics of Fuzzy Logic.*- Kluwer:- 1998.

[-*Wikipedia,* Heuristics] Heuristics. [*Electronic resourse*] -*Wikipedia:* http://en.wikipedia.org/wiki/Heuristic.

[*Wikipedia* ,Kahneman:] Kahneman.- [*Electronic resourse*] -*Wikipedia:* http://en.wikipedia.org/wiki/Daniel_Kahneman

[Kaufmann, 1982] Kaufmann A. *Introduction in the Theory of Fuzzy Sets,* - Moscow.- 1982. (in Russian).

[Kuratovski, Mostowski,1967] Kuratovski K., Mostowski A. . *Set theory* – North Holland Publishing Company, Amsterdam.- 1967

[Newell et al, 1962] Newell A., Shaw J. C., Simon H. A. Empirical Explorations of the Logic Theory Machine: A Case Study in Heuristic. In *J. Symbolic Logic.- 1962. -Volume 27, Issue 1- P. 102-103*.

[Polya, 1945] Polya G. *How To Solve It: A New Aspect of Mathematical Method*.- Princeton, NJ: Princeton University Press. -1945.-ISBN 0-691-02356-5 ISBN 0-691-08097-6

[Петровский, 2002] Петровский А.Б. Основные понятия теории мультимножеств. – Москва: Едиториал УРСС. – 2002.- 80 с.

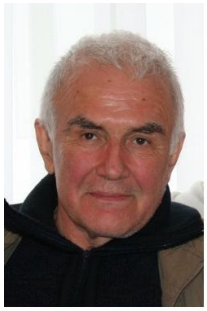
[Поспелов, 2001] Поспелов Д. Из истории развития нечетких множеств и мягких вычислений в России.- In *Новости искусственного интеллекта. – 2001. – №2-3*.

[Редько et al,2001] Редько В.Н., Брона Ю.Й., Буй Д. Б. , Поляков С.А. Реляційні бази даних: табличні алгебри та SQL-подібні мови. – Київ: Видавничий дім "Академперіодика". – 2001- 198 с.

[Stoll, 1960] Stoll R. *Sets, Logic and Axiomatic Theories.* Freeman and Company, San Francisco.- 1960.

[Zadeh, 1965] Zadeh L. Fuzzy Sets. In *Information and Control, 8(3). June 1965. pp. 338-53*.

Authors' Information



Volodymyr Donchenko *– Professor, National Taras Shevchenko University of Kyiv. Volodymyrs'ka street, Kyiv, 03680, Ukraine; e-mail: voldon@bigmir.net.*

DISTURBANCE OF STATISTICAL STABILITY
(PART II)

Igor Gorban

Abstract: *The revue of results obtained by the author this year in the area of disturbance of statistical stability of physical quantities and processes is presented. Two new sensitive parameters characterizing their statistical instability in finite observation interval are proposed. For known and new parameters of statistical instability unit measures that give possibility to characterize the level of disturbance are introduced. It is shown by modeling that STD corridors of unit measures weakly depend from the type of the distribution. It is confirmed that important role in disturbance of statistical stability plays specific fluctuation of expectation of the process that generates changes of average expectation. Dependence of parameters of statistical instability from normalized sample variance of expectation average is found. It is shown by modeling that positive correlation calls increasing of statistical instability and negative correlation – decreasing of it. Under any type correlation increasing of sample size calls increasing of statistical stability and the process trends to statistically stable state. It is found that changes of variance influence to statistical stability of the process. Under some relations of parameters unpredictable component of the process fades-out and under other ones – fades-in. It is demonstrated high sensitivity of new parameter of statistical instability. It is shown that fluctuations of maximum day temperature and minimum day temperature in two cities (Moscow and Kiev) are instable and fluctuation of day precipitation is near stable. Disturbance of statistical stability of temperature begins from some observation weeks.*

Keywords: *statistical instability, theory of hyper-random phenomena, uncertainty, probability.*

ACM Classification Keywords: *G.3 Probability and Statistics*

1. Introduction

In the former conference [Gorban, 2010 (2)], the survey of author's researches in the theory of hyper-random phenomena, in particular in the area devoted to disturbance of statistical stability of physical quantities and processes was introduced. In the current article, the revue of the results obtained by the author in this area during last year is presented.

In the article, the main consideration was given to three directions:

- unit measures for parameters of statistical instability,
- new sensitive parameters of statistical instability,
- statistical instability of real processes.

2. Unit measure for parameters of statistical instability

Measuring of any physical magnitude beginning from it's comparing with some unit measure.

In [Gorban, 2010 (2)] and [Gorban, 2011 (1)] two parameters of statistical instability of a sequence of N physical units $X_1, X_2, \ldots, X_N$ were introduced ad learned. They are

$$\gamma_N = \frac{\mathrm{M}\left[\bar{D}_{Y_N}\right]}{ND_{y_N}} \tag{1}$$

and

$$\mu_N = \sqrt{\frac{\gamma_N}{1+\gamma_N}} \tag{2}$$

where $\mathrm{M}[\cdot]$ is expectation operator,

$$\bar{D}_{Y_N} = \frac{1}{N-1}\sum_{n=1}^{N}(Y_n - \bar{m}_{Y_N})^2 \tag{3}$$

is a sample variance of the average

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (n = \overline{1,N}), \tag{4}$$

$$\bar{m}_{Y_N} = \frac{1}{N}\sum_{n=1}^{N} Y_n \tag{5}$$

is it's sample mean,

$$D_{y_N} = \frac{1}{N^2}\sum_{n=1}^{N} D_{x_n}$$

is a variance of the average, and D_{x_n} is a variance of the unit X_n.

For parameter γ_N the role of unit measure may play the variable γ_{0N} calculated according to formulae (1), (3)–(5) for sequence of sample uncorrelated units with constant variance $D_{x_n} = D_x$, zero expectation, and the same number of units; for parameter μ_N it may play the variable $\mu_{0N} = \sqrt{\gamma_{0N}/(1+\gamma_{0N})}$.

Mark, that in this approach distribution low is not essential and therefore is not stipulated.

Variable γ_{0N} for pointed sample sequence was calculated analytically in the following form [Gorban, 2011 (2)]:

$$\gamma_{0N} = \frac{N+1}{(N-1)N} C_N - \frac{2}{N-1}, \tag{6}$$

where $C_N = \sum_{n=1}^{N} \frac{1}{n}$.

It was impossible to obtain analytically STD for the variable $\tilde{\gamma}_{0N} = \bar{D}_{Y_N} / N D_{y_N}$ without any addition assumptions. For Gaussian sample sequence it was calculated in the following form:

$$\sigma_{\tilde{\gamma}_{0N}} = \frac{1}{N-1}\sqrt{\frac{2C_N^2}{N^2} + \frac{4(N+1)C_N}{N} + A_N\left(\frac{4}{N} - 2\right) + \frac{8B_N}{N} - 12}, \tag{7}$$

where $A_N = \sum_{n=1}^{N} \frac{1}{n^2}$, $B_N = \sum_{n=1}^{N} \frac{C_{n-1}}{n}$.

It was shown [Gorban, 2011 (2)] by modeling that the corridors $\gamma_{0N}^{\pm} = \gamma_{0N} \pm \sigma_{\tilde{\gamma}_{0N}}$ and $\mu_{0N}^{\pm} = \sqrt{\gamma_{0N}^{\pm}/(1+\gamma_{0N}^{\pm})}$ are practically identical for Gaussian and uniform distributions. Therefore it is possible to assume that deviation of the variables $\tilde{\gamma}_{0N}$ and $\tilde{\mu}_{0N}$ weakly depend from the type of the distribution.

3. New parameters of statistical instability

By using unit measure γ_{0N} it is possible to introduce relative variable

$$h_N = \gamma_N / \gamma_{0N} \tag{8}$$

that characterizes the absolute level of statistical instability in the units γ_{0N}.

Relative level of statistical instability characterizes not only parameter μ_N but also parameter

$$I_N = \frac{\gamma_N - \gamma_{0N}}{\gamma_N} = \frac{h_N - 1}{h_N} \tag{9}$$

that is ratio of the parameter of statistical instability calculated for unpredictable part of the process to the parameter of statistical instability calculated for whole process.

Parameter I_N is determined by the parameter μ_N and the unit measure γ_{0N}:

$$I_N = (1 + \gamma_{0N}) - \frac{\gamma_{0N}}{\mu_N^2}.$$

Turndown of the parameter h_N is $[0, \infty)$ and of the parameter I_N is $(-\infty, 1]$.

Dependencies of parameter γ_{0N} and its corridors $h_{0N}^{\pm} = \gamma_{0N}^{\pm} / \gamma_{0N}$ from sample size N are presented in fig. 1 by solid and dotted lines accordingly. Dependencies of parameter μ_{0N} and its corridors $I_{0N}^{\pm} = (\gamma_{0N}^{\pm} - \gamma_{0N}) / \gamma_{0N}^{\pm}$ from sample size N are presented in fig. 2 by solid and dotted lines too.

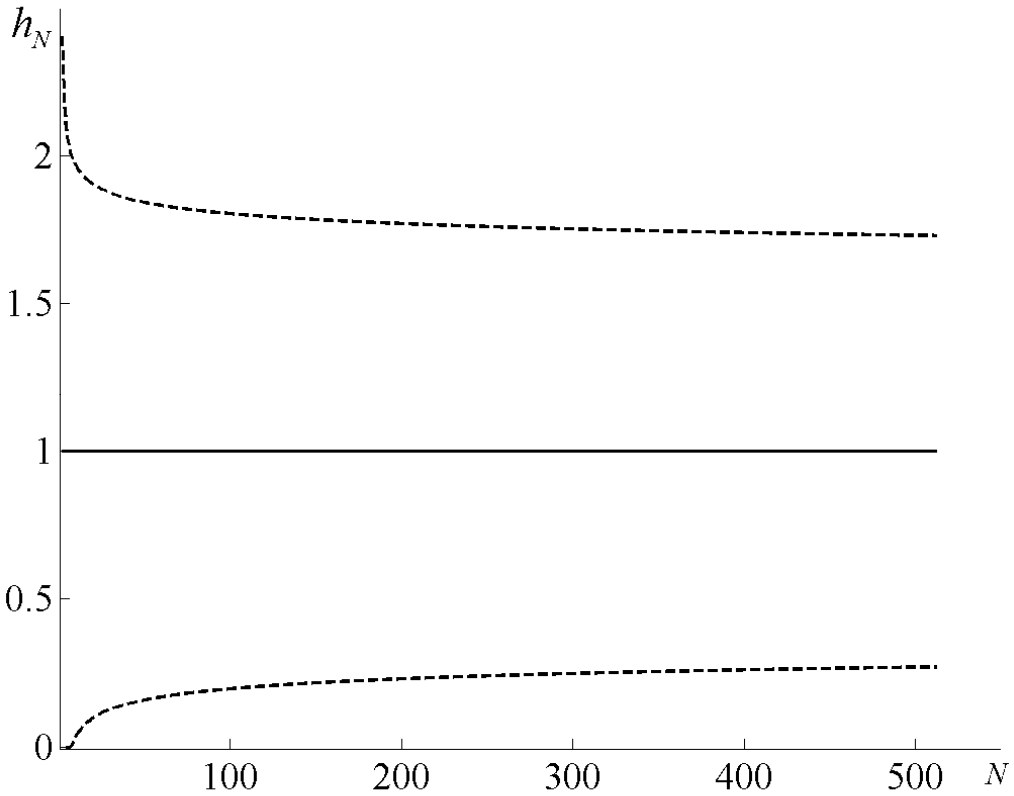


Fig. 1

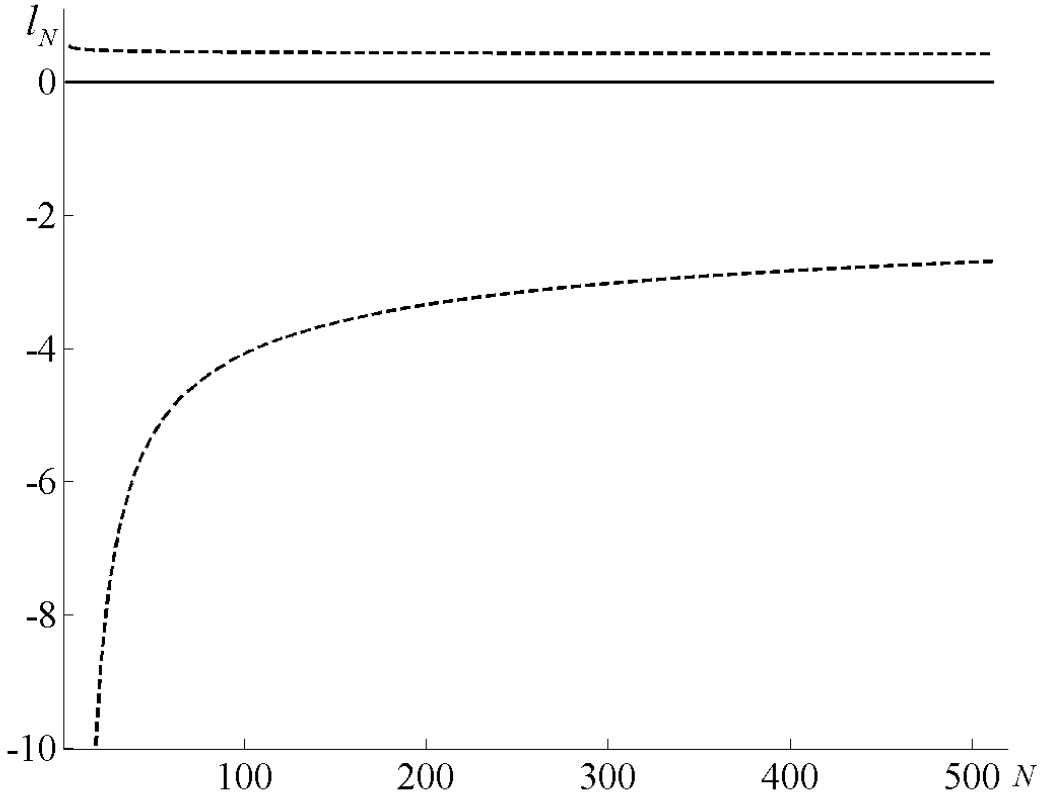


Fig. 2

It follows from the figures that with rising of sample size N the corridors $h_{0N}^{\pm}$ and $I_{0N}^{\pm}$ are converged.

4. Particularities of statistical instability parameters

Statistical instability parameters γ_N, h_N, μ_N, and I_N are physical magnitudes that characterize processes. In contrast to a lot of different other unit measures, correspondent units γ_{0N}, h_{0N}, μ_{0N}, and I_{0N} not request physical standard samples because are mathematical functions defined by sample size N. For fixed N they can be calculated without any error.

In physics, little physical constants such as light velocity, gravity constant, and others [Fundamental physical constants] are defined on definition with zero error. Statistical instability parameters γ_N, h_N, μ_N, and I_N have zero error too. In this case, zero errors is a result of that the parameters are mathematical functions.

Possibility using mathematical functions as unit measures of physical magnitudes is a result of that parameters γ_N, h_N, μ_N, and I_N have not dimension (are relative variables).

In [Gorban, 2010 (1)], [Gorban, 2010 (2)], and [Gorban, 2011 (1)] reasons which led to disturbance of statistical stability was researched. It was marked that important role plays fluctuations of some types of expectation. Periodical fluctuation components of expectation with low period and short-time steps do not disturb stability but fluctuation components with period that comparable with observation interval and also prolonged aperiodic processes lead to disturbance of stability.

Some results of more detail researches of reasons [Gorban, 2011 (2)] led to disturbance of statistical stability are presented downstream.

5. Fluctuation of expectation

Let the process described by sequence $X_1, X_2, \ldots$ has constant variance D_x. Such process may be presented by sum of centered process described by sequence $\overset{\circ}{X}_1, \overset{\circ}{X}_2, \ldots$ and deterministic process described by sequence of expectations $m_{x_1}, m_{x_2}, \ldots.$

So average $Y_n = \bar{m}_{y_n} + \overset{\circ}{Y}_n$, where $\bar{m}_{y_n} = \frac{1}{n}\sum_{i=1}^{n} m_{x_i}$ is current average of expectations and $\overset{\circ}{Y}_n = \frac{1}{n}\sum_{i=1}^{n} \overset{\circ}{X}_i$ is average of sequence $\overset{\circ}{X}_1, \overset{\circ}{X}_2, \ldots.$

Statistical instability parameter γ_N may be obtained in the following form:

$$\gamma_N = q_N + \gamma_{0N}, \tag{10}$$

where $q_N = \bar{D}_{\bar{m}_{yN}} \big/ D_x$ is normalized sample variance of expectation average, $\bar{D}_{\bar{m}_{yN}}$ is sample variance of expectation average:

$$\bar{D}_{\bar{m}_{yN}} = \frac{1}{N}\sum_{n=1}^{N} (\bar{m}_{y_n} - \bar{m}_{\bar{m}_{yN}})^2,$$

$\bar{m}_{\bar{m}_{yN}} = \frac{1}{N}\sum_{n=1}^{N} \bar{m}_{y_n}$ is average of averages of expectations of the sequence $X_1, X_2, \ldots.$

In these designations, statistical instability parameter

$$\mu_N = \sqrt{\frac{q_N + \gamma_{0N}}{q_N + (1 + \gamma_{0N})}}, \tag{11}$$

parameter

$$h_N = \frac{q_N}{\gamma_{0N}} + 1, \tag{12}$$

and parameter

$$I_N = \frac{q_N}{q_N + \gamma_{0N}}. \tag{13}$$

It is followed from expressions (10)–(13) that changes of average of expectations $\bar{m}_{y_n}$ lead to increasing of statistical instability parameters. Parameters γ_N, h_N, μ_N, and I_N rise with rising of variance of these changes.

It must be marked that not any fluctuation of expectation of initial process leads to disturbance of statistical stability but only special one that calls fluctuation of $\bar{m}_{y_n}$.

Image about dependence of parameters μ_N and I_N from normalized variance of fluctuations of average of expectations q_N gives fig. 3. Curves 1 – 3 accords to parameter μ_N and $1' - 3'$ – to parameter I_N. Curves $1,\ 1'$ are obtained for $N = 16,$ curves $2,\ 2'$ – for $N = 256$ and $3,\ 3'$ – for $N = 4096$.

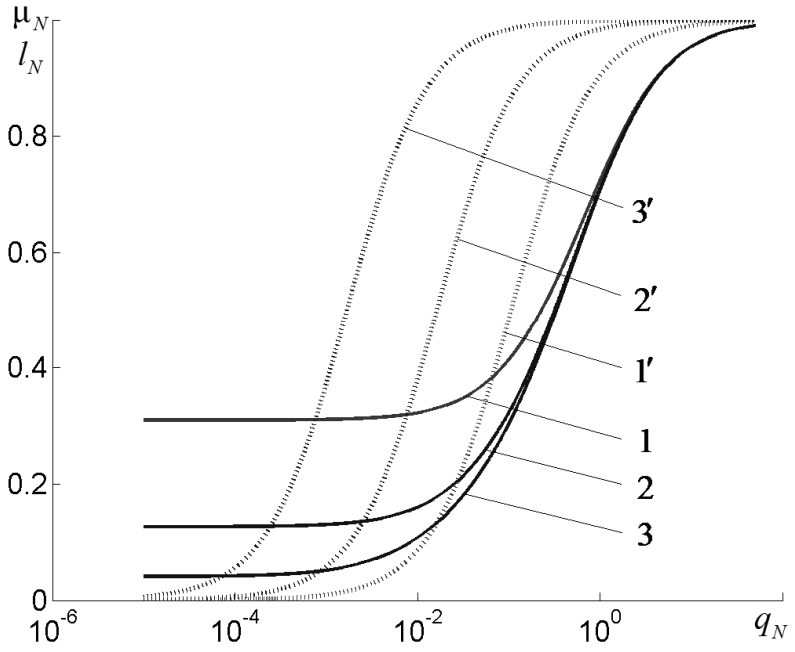


Fig. 3

Apparently from figure, when $q_N < 1$ more sensitive to changes of q_N is parameter l_N and when $q_N > 1$ – parameter μ_N. So under low disturbances of statistical stability more preferable use parameter l_N and under high disturbances – parameter μ_N.

With rising of sample size N parameter μ_N decreases and parameter l_N, contrary, increases.

When there is low disturbances of statistical stability ($q_N \ll \gamma_{0N}$) parameter l_N depends from q_N practically in line low (see expression (13)).

Not only variances of expectation may lead to change of statistical stability but another deviations from sample conditions, in particular correlation of sample units and variations of variance.

6. Correlation

Parameters of statistical instability calculated by modeling for different correlation conditions are presented in fig. 4 and 5. Solid bold curves correspond to sample conditions (uncorrelated Gaussian sequence with constant variance and zero expectation), solid extra bold curves – to sequence with positive correlated units, and dotted extra bold curves – to sequence with negative correlated ones.

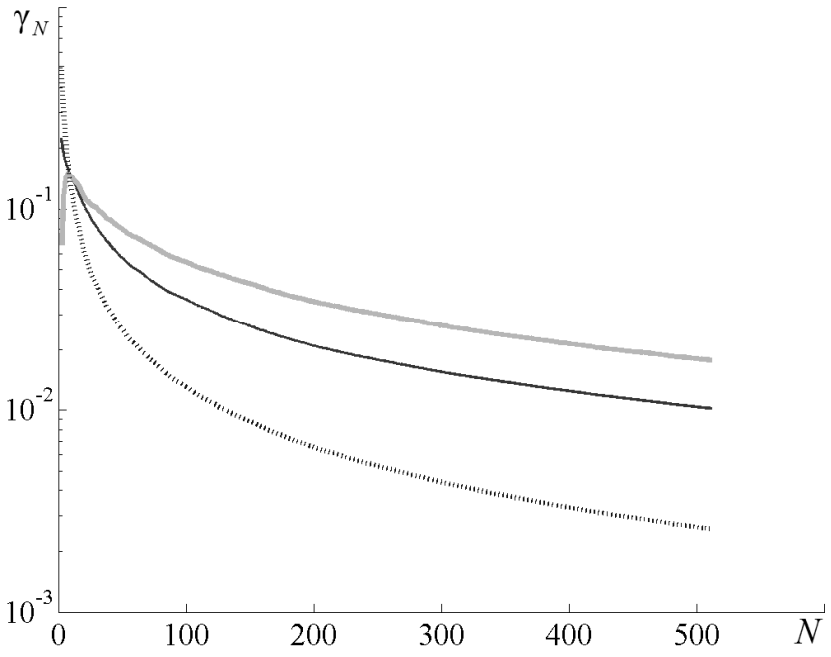


Fig. 4

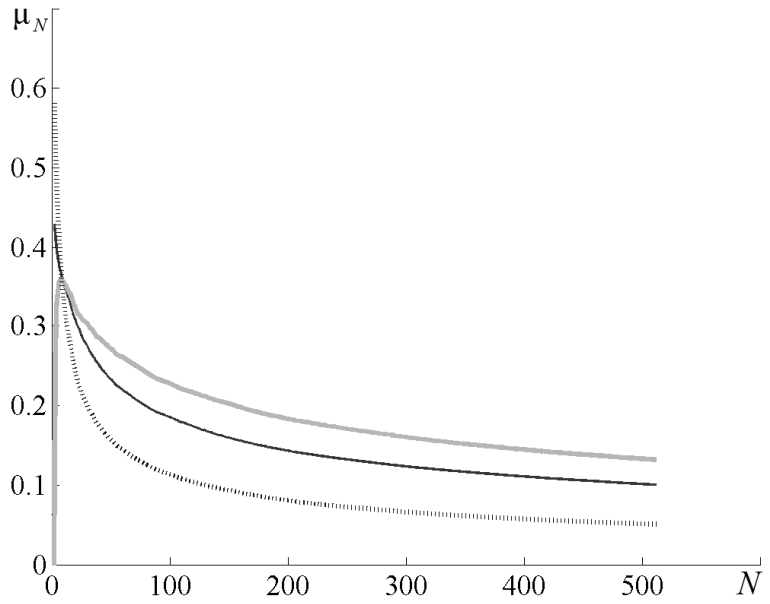


Fig. 5

It follows from figures that positive correlation calls increasing of statistical instability and negative correlation – decreasing of it.

Under any type correlation increasing of sample size N calls increasing of statistical stability and the process tends to statistically stable state.

7. Fluctuation of variance

Fluctuation of variance can call statistical instability.

Let us research sequence $X_1, X_2, \ldots$ with uncorrelated units, variance D_{x_n}, and zero expectation that can be presented by sum of two uncorrelated sequences, one of which ($\hat{X}_1, \hat{X}_2, \ldots$) has constant variance $D_{\hat{x}} = \min_n D_{x_n}$ and zero expectation and another one ($\breve{X}_1, \breve{X}_2, \ldots$) – fluent variance $D_{\breve{x}_n}$ and zero expectation.

In this case, average $Y_n = \hat{Y}_n + \breve{Y}_n$, variance of average

$$D_{y_N} = D_{\hat{y}_N} + D_{\breve{y}_N} + 2R_{\hat{y}_N\breve{y}_N}, \tag{14}$$

and sample variance of average

$$\bar{D}_{Y_N} = \bar{D}_{\hat{Y}_N} + \bar{D}_{\breve{Y}_N} + 2\bar{R}_{\hat{Y}\breve{Y}_N}, \tag{15}$$

were $\hat{Y}_n$ and $\breve{Y}_n$ are averages of sequences $\hat{X}_1, \hat{X}_2, \ldots$ and $\breve{X}_1, \breve{X}_2, \ldots$ accordingly, $D_{\hat{y}_N} = \frac{1}{N^2}\sum_{n=1}^{N} D_{\hat{x}_n}$ and $D_{\breve{y}_N} = \frac{1}{N^2}\sum_{n=1}^{N} D_{\breve{x}_n}$ are variances of $\hat{Y}_n$ and $\breve{Y}_n$, $R_{\hat{y}_N\breve{y}_N} = \frac{1}{N^2}\sum_{n=1}^{N} R_{\hat{x}_n\breve{x}_n}$ is their correlated moment, $D_{\hat{x}_n}$, $D_{\breve{x}_n}$ are variances of sequences $\hat{X}_1, \hat{X}_2, \ldots$ and $\breve{X}_1, \breve{X}_2, \ldots$, $R_{\hat{x}_n\breve{x}_m}$ is correlated moment of $\hat{X}_n$ and $\breve{X}_m$, $\bar{D}_{\hat{Y}_N}$ and $\bar{D}_{\breve{Y}_N}$ are sample variances of averages $\hat{Y}_n$ and $\breve{Y}_n$: $\bar{D}_{\hat{Y}_N} = \frac{1}{N-1}\sum_{n=1}^{N} \hat{Y}_n^2$ and $\bar{D}_{\breve{Y}_N} = \frac{1}{N-1}\sum_{n=1}^{N} \breve{Y}_n^2$, $\bar{R}_{\hat{Y}\breve{Y}_N} = \frac{1}{N-1}\sum_{n=1}^{N} \hat{Y}_n\breve{Y}_n$ is their sample correlated moment.

Taking into account that variables $\hat{X}_n$, $\breve{X}_m$ are uncorrelated ($R_{\hat{x}_n\breve{x}_m} = 0 \quad \forall n = \overline{1,N}$ и $\forall m = \overline{1,N}$) and there are relations (14), (15), expression for parameter γ_N can be obtained in the following form:

$$\gamma_N = \frac{\gamma_{0N} + \dfrac{M[\bar{D}_{\breve{Y}_N}]}{ND_{\hat{y}_N}}}{1 + \dfrac{D_{\breve{y}_N}}{D_{\hat{y}_N}}}. \tag{16}$$

It follows from this expression that if expectation of sample variance $\bar{D}_{\breve{Y}_N}$ of the sequence $\breve{X}_1, \breve{X}_2, \ldots$ normalized to variance of average $D_{\breve{y}_N}$ and sample size N is more then sample variable γ_{0N}:

$$\left(\frac{M[\bar{D}_{\breve{Y}_N}]}{ND_{\breve{y}_N}} > \gamma_{0N}\right), \tag{17}$$

parameter of statistical instability $\gamma_N > \gamma_{0N}$. If inequality (17) is not true then $\gamma_N \le \gamma_{0N}$.

It follows from expression (16) that if variable $\mathrm{M}[\bar{D}_{\bar{Y}_N}]/ND_{\bar{y}_N}$ tends to zero the process is statistically stable and in contrary case is unstable.

Results of computer modeling illustrating this resume are presented in fig. 6 and 7. Bold curves accord to sample sequence, extra bold ones – to sequence that consist of sum of sample sequence and sequence in which variance increases in line low with increasing of sample size N, and dotted extra bold ones – to sequence that consist of sum of sample sequence and sequence in which variance decreases in line low with increasing of sample size N.

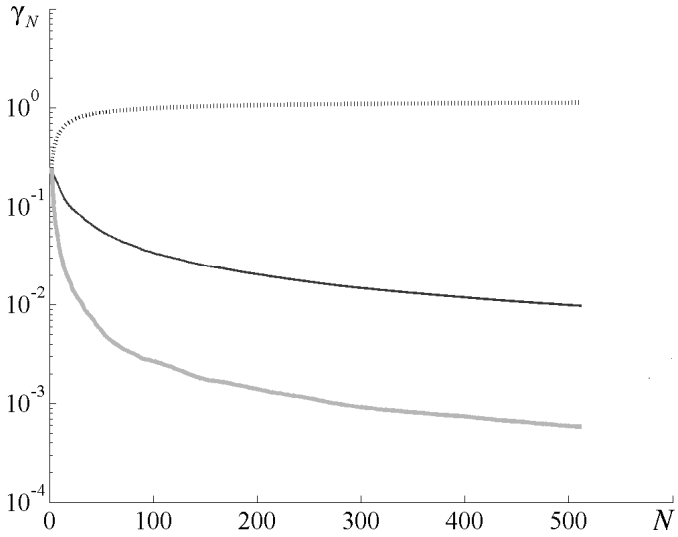


Fig. 6

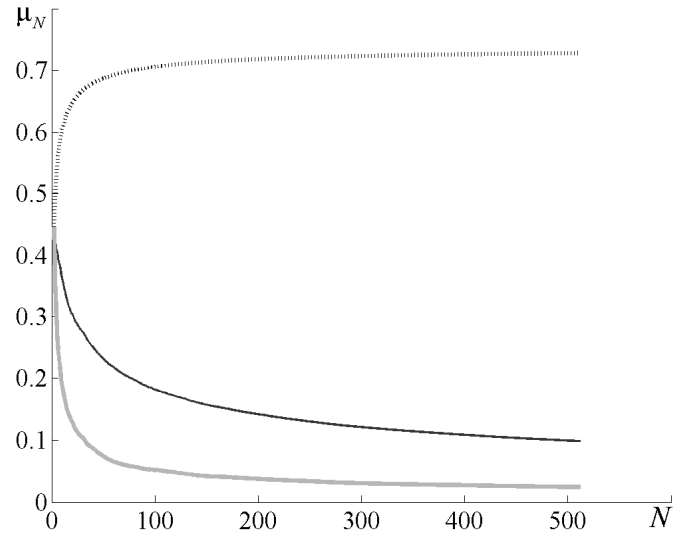


Fig. 7

It follows from figures that in the second case the process more stable and in the third case less stable than in the first one. In the second case the process explicitly tends to statistically stable state and in the third one – to statistically instable state.

So changes of variance influence to statistical stability of the process. Under some relations of parameters unpredictable component of the process fades-out and under other ones – fades-in.

8. Example of statistically instable process

Statistical stability of maximum day temperature, minimum day temperature, and day precipitation was researched with using developed methodology. Initial data was received from [Weather] for Moscow (1949 – 1992) and for Kiev (1881 – 1992).

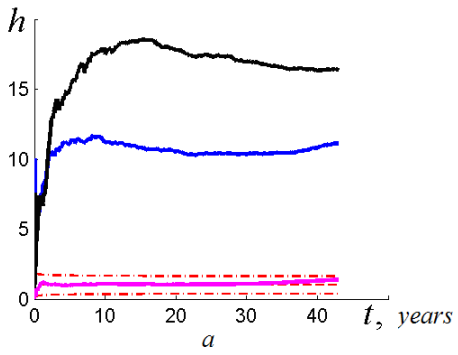
Dependences of parameters h_N and μ_N from time t calculated according to this methodology are presented in fig. 8 for Moscow and in fig. 9 – for Kiev.

Curves in fig. 8a, 8b, 9a, 9b are obtained without ensemble averaging of sample variance $\bar{D}_{Y_N}$ and in fig. 8c, 8d and 9c, 9d – with averaging for 43 and 112 years correspondently. Overhead solid curves accord to maximum day temperature and minimum day temperature. Bottom solid curves correspond to day precipitation. For comparing, unit measures of parameters of statistical instability are presented by dash lines and abnormality from them on STD (STD corridors) – by dash-dot lines.

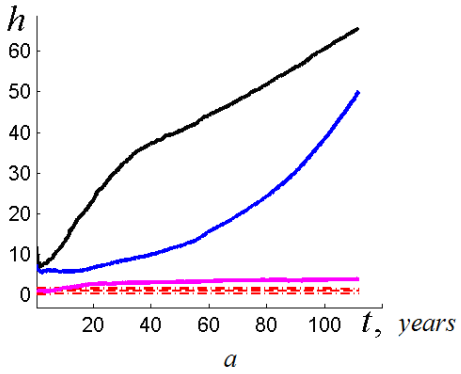
Curves were calculated with preliminary seasonal correction data realized by subtraction of expectation estimates and then normalizing obtained magnitudes on according STD estimates.

By analyzing parameter h_N (see fig. 8b, 8d, 9b, 9d) it is impossible to answer on the question: are researched processes statistically stable or not? Although it is clear from curves in these figures that fluctuations of maximum day temperature and minimum day temperature are less stable than fluctuation of day precipitation.

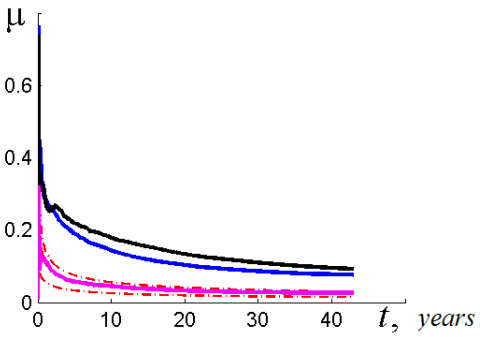
Answer on pointed question gives curves describing parameter μ_N (see fig. 8a, 8c, 9a, 9c). It follows from the figures that fluctuations of maximum day temperature and minimum day temperature are instable and fluctuation of day precipitation is near stable. Disturbance of statistical stability of temperature begins from some observation weeks.



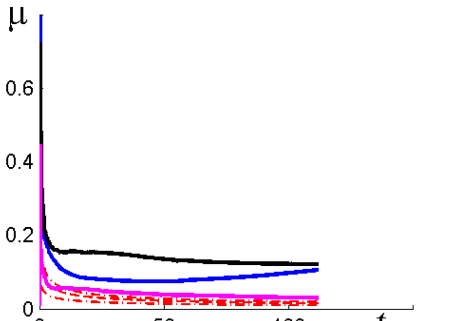
a



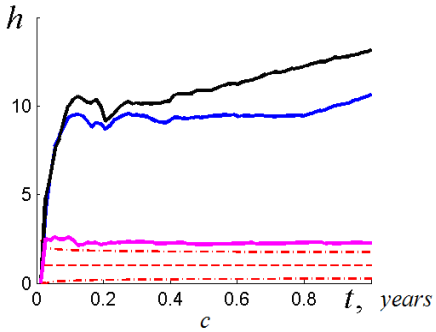
a



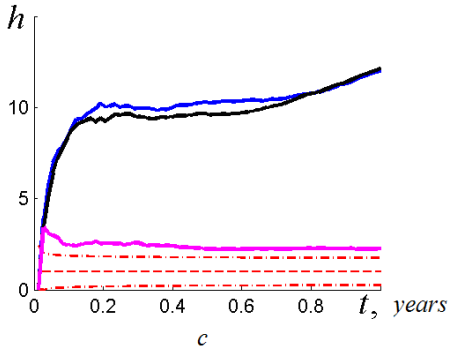
b



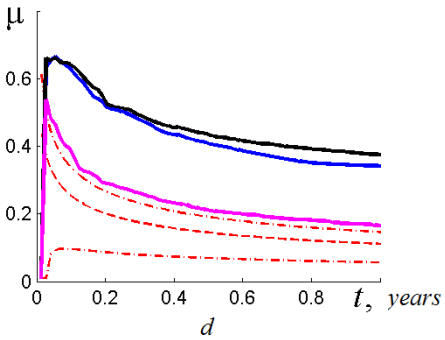
b



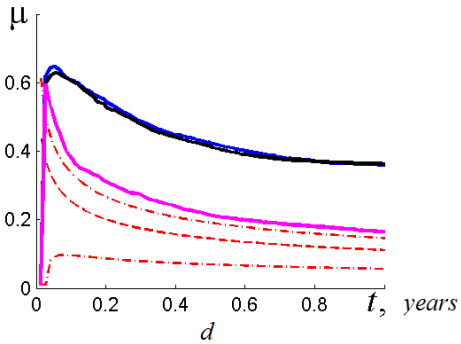
c



c



d



d

Fig. 8

Fig.9

9. Conclusion

1. To describe disturbance of statistical stability on the finite observation interval two sensitive parameters (h_N and I_N) of statistical instability has been proposed.

2. For known (γ_N , μ_N) and new (h_N , I_N) parameters of statistical instability unit measures that give possibility to characterize the level of disturbance in the finite observation interval have been introduced.

3. Modeling has shown that STD corridors of unit measures weakly depend from the type of the distribution.

4. It has been confirmed that important role in disturbance of statistical stability plays specific fluctuation of expectation of the process that generates changes of average expectation. Dependence of parameters of statistical instability from normalized sample variance of expectation average has been found.

5. It has been shown by modeling that positive correlation calls increasing of statistical instability and negative correlation – decreasing of it. Under any type correlation increasing of sample size calls increasing of statistical stability and the process trends to statistically stable state.

6. It has been found that changes of variance influence to statistical stability of the process. Under some relations of parameters unpredictable component of the process fades-out and under other ones – fades-in.

7. It has been shown that fluctuations of maximum day temperature and minimum day temperature in two cities (Moscow and Kiev) are instable and fluctuation of day precipitation is near stable. Disturbance of statistical stability of temperature begins from some observation weeks.

Bibliography

[Gorban, 2010 (1)] I.I. Gorban. Distraction of statistical stability // Mathematical Machines and Systems. – №1, 2010. – P. 171 – 184. (In Russion).

[Gorban, 2010 (2)] I.I. Gorban. Disturbance of statistical stability // Information Models of Knowledge 2010. – 2010. P. 398 – 410.

[Gorban, 2011 (1)] I.I. Gorban. Theory of hyper-random phenomenon. Kiev: Naukova dumka. – 2011. – 318 p. – ISBN 978-966-00-1093-2. – http://www.immsp.kiev.ua/perspages/gorban_i_i/index.html (In Russion).

[Gorban, 2011 (2)] I.I. Gorban. Statistical instability of physical processes // Radioelectronics and Communications Systems. – 2011. (In printing).

[Gorban, 2011 (3)] I.I. Gorban. Statistical stability of temperature and precipitation in the region of Moscow // Mathematical Machines and Systems. – №3, 2011. (In Russion).

[Fundamental physical constants] Fundamental physical constants. – http://physics.nist.gov/constants.

[Weather] Weather history for CIS towns. – http://thermo.karelia.ru/weather/w_history.php.

Authors' Information

Igor Gorban *– Principal scientist of the Institute of Mathematical Machines and Systems Problem, National Academy of Sciences of Ukraine, Glushkov ave., 42, Kiev, 03187, Ukraine; e-mail: igor.gorban@yahoo.com*

Major Fields of Scientific Research: Theory of hyper-random phenomena, Space-time signal processing in complicated dynamic conditions, Fast multichannel space-time signal processing

HYBRID CASCADE NEURAL NETWORK BASED ON WAVELET-NEURON

Yevgeniy Bodyanskiy, Oleksandra Kharchenko, Olena Vynokurova

***Abstract**: In the paper new hybrid cascade wavelet-neural network and its learning algorithm in batch and on-line mode are proposed. Such architecture can be used for solving prediction and emulation non-stationary non-linear time series under current and a-priori uncertenity. The computational experiments confirm the effectiveness of developed approach.*

***Keywords**: cascade neural network, wavelet-neuron, learning algorithm, prediction, emulation.*

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

Introduction

As part of an evolutionary approach to the synthesis of neural networks architecture it can be provided such direction as the cascade neural networks [Avedjan, 1999; Bodyanskiy, 2004a; Bodyanskiy, 2006; Bodyanskiy, 2007; Bodyanskiy, 2008a; Fahlman, 1990; Schalkoff, 1997]. The cascade-correlation neural network, which proposed S. Fahlman and C. Lebiere [Fahlman, 1990], is the most typical and effective representative of such neural networks. The main feature of this network type is the ability to add new nodes during learning process.

Thus the cascade neural networks are the flexible and effective approach for solving of the wide type tasks related to data mining under full and partial, a-priori and current uncertainty. The cascade neural networks are fully adaptive evolutionary architectures, because they tune not only synaptic weight but and cascade number during learning process. Fitting cascade number allows to select the architecture with complexity which is most suitable for the solving problem without the experts intervention in the object domain.

The main advantages of cascade-correlation networks are following ones:

- such networks do not demand a-priori defining both architecture network and neuron number in the cascades;
- neurons are added to the network as necessary, creating no hidden layers, but cascades, each of them uses the input of network and previous cascade outputs as the own input signals;

- the learning process is not associated with concept of back-propagation, that allows significantly to reduce the learning process time;
- the computational burden on the learning process is reduced by "freezing" of synaptic weights that formed previous cascades.

The main disadvantage of such networks is impossibility of their learning process in on-line mode [Bodyanskiy, 2004a], due to the type of used artificial neurons – elementary Rosenblatt perceptrons [Rosenblatt, 1964; Rosenblatt, 1966]. As is known in such neurons the sigmoidal or hyperbolic tangent functions are used as activation functions and as a result of the output signal of each neuron depends nonlinearly from the synaptic weights. So learning process should be performed using the delta-rule and its modifications, which are gradient optimization algorithms. Obviously, it is difficult to talk about optimizing of the learning rate and learning process in on-line mode in this case.

In the connection with that it seems appropriate to synthesize the hybrid cascade architecture, where the cascades use neurons, in which the output signal linearly depends of the synaptic weights that permits to optimize learning speed and reduce the size of the training set.

Wavelet-neuron and its learning algorithm

Let us consider the structure of the wavelet-neuron [Bodyanskiy, 2003; Bodyanskiy, 2004b; Bodyanskiy, 2005] shown in fig. 1. Evidently, wavelet-neuron is sufficiently close to structure of conventional n-input formal neuron, but instead of tuning synaptic weights contains wavelet synapses WS_i, $i = 1,2,\ldots,n$, whose adjustable parameters are not only weights w_{ji}, but also the center and width of wavelet activation function $\varphi_{ji}(x_i(k))$.

When vector signal $x(k) = (x_1(k), x_2(k),\ldots, x_n(k))^T$ (where $k = 0,1,2,\ldots$ is discrete current time) is fed to the input of the wavelet-neuron, the output is determined by both the tunable weights $w_{ji}(k)$ and wavelet functions:

$$y(k) = \sum_{i=1}^{n} f_i(x(k)) = \sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}(k)\varphi_{ji}(x_i(k)). \qquad (1)$$

Notice that wavelet-neuron architecture coincides with the neo-fuzzy neuron of T. Yamakawa [Miki, 1999; Uchino, 1997; Yamakawa, 1992], but differs in that, instead of triangular membership functions are used even wavelets in nonlinear synapses. However, as shown by B. Kosko [Mitaim, 1997], the use of even wavelets does not contradict the ideas of fuzzy inference, and the specific values of wavelet functions can be given a sense of membership levels.

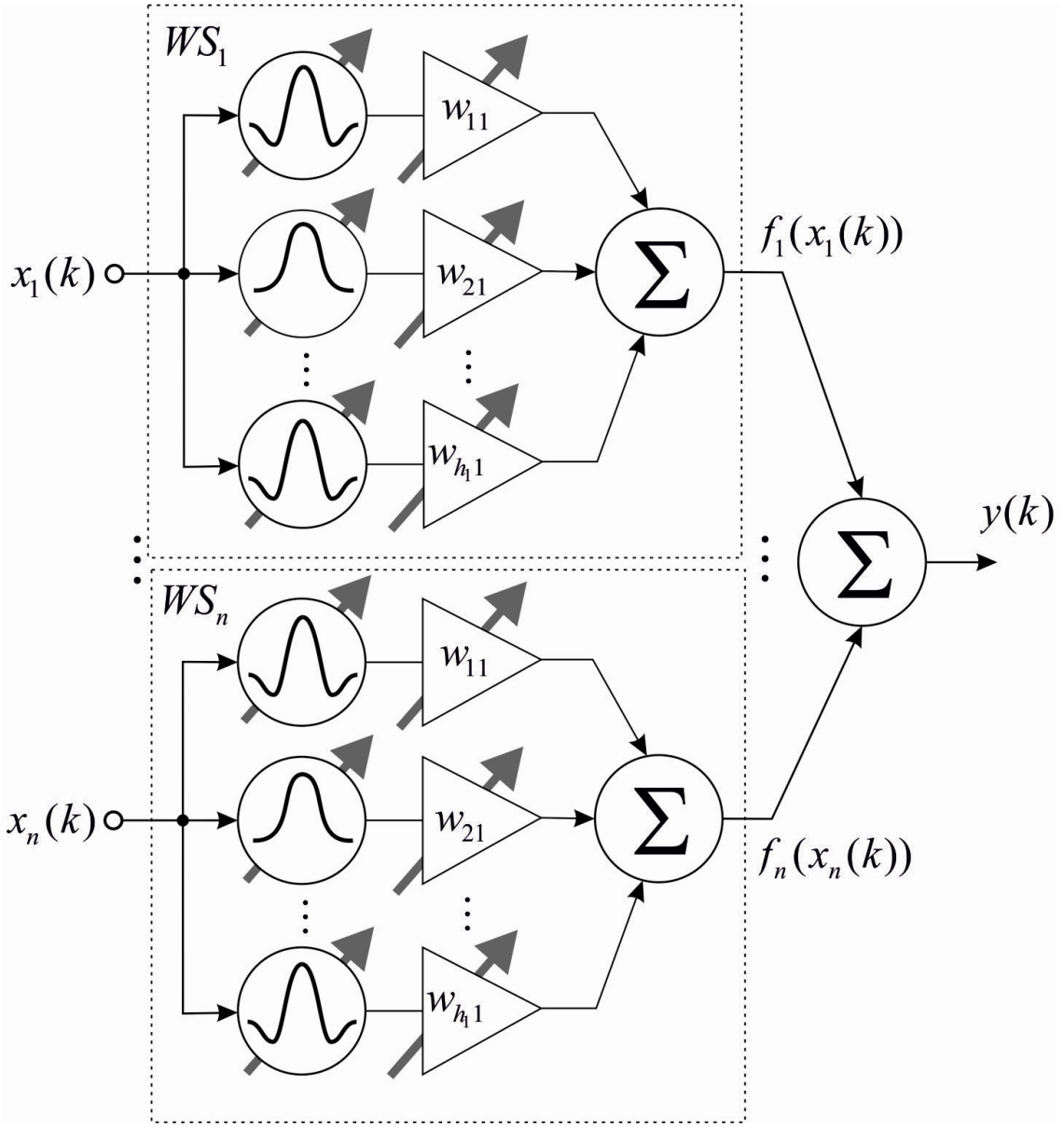


Figure 1. Wavelet-neuron architecture

The different wavelet-function type can be used as the activation function of wavelet-neuron. The most suitable function is proposed by us - adaptive wavelet activation-membership function [Bodyanskiy, 2008b] which has the form

$$\varphi_{ji}(x_i(k)) = (1 - \alpha_{ji}(k)\tau_{ji}^2)\exp\left(-\tau_{ji}^2(k)/2\right), \tag{2}$$

where $\tau_{ji}(k) = (x_i(k) - c_{ji}(k))\sigma_{ji}^{-1}(k)$, $c_{ji}(k), \sigma_{ji}(k)$ are parameters, which define the center, width and $\alpha_{ji}(k)$ is parameter of function shape.

In [Bodyanskiy, 2003] the enough simple and effective learning algorithm of wavelet-neuron is proposed, which has the form

$$w_{ji}(k+1) = w_{ji}(k) + \eta^w(k)e(k)(1-\alpha_{ji}(k)\tau_{ji}(k))\exp\left(-\tau_{ji}^2(k)/2\right), \tag{3}$$

where scalar coefficient $\eta^w(k)$ defines the step in the tuning parameters space.

The rate of convergence of the learning algorithm (3) can be increased via using of the second-order procedures, such as the Levenberg-Marquardt algorithm [Shepherd, 1997] which are widely used to train neural networks.

Introducing $(h_i \times 1)$-vectors of variables $\varphi_i(x_i(k)) = (\varphi_{1i}(x_i(k)),\ldots,\varphi_{h_i i}(x_i(k)))^T$, $w_i(k) = (w_{1i}(k),\ldots,w_{h_i i}(k))^T$ and $\tau_i(k) = (\tau_{1i}(k), \ldots, \tau_{h_i i}(k))^T$, we can obtain the gradient-based learning algorithm of the *i*-th wavelet synapse WS_i [Bodyanskiy, 2005; Otto, 2003]:

$$\begin{cases} w_i(k+1) = w_i(k) + \dfrac{e(k)\varphi_i(x_i(k))}{\eta^w + \left\|\varphi_i(x_i(k))\right\|^2} = w_i(k) + \dfrac{e(k)\varphi_i(x_i(k))}{\eta_j^w(k)}, \\ \eta_i^w(k+1) = \beta\eta_i^w(k) + \left\|\varphi_i(x_i(k))\right\|^2, \end{cases} \tag{4}$$

which has both tracking and filtering properties, where β is a forgetting factor ($0 \le \beta \le 1$).

Hybrid cascade wavelet neural network

Replacing neurons in the nodes of the cascade network by the elements, whose output signals are linearly dependent of the synaptic weight we can escape disadvantages and even get some useful properties due to the choice of artificial neuron type.

Replacing Rosenblatt perceptrons in nodes of cascade correlation neural networks which proposed Fahlman and Lebier on the wavelet-neurons we can to introduce hybrid wavelet cascade architecture shown on the fig. 2.

Such cascade wavelet-neural network is realized mapping of the following form

– wavelet-neuron of first cascade

$$\hat{y}^{[1]} = \sum_{i=1}^{n}\sum_{j=1}^{h}\varphi_{ji}^{[1]}\mu_{ji}(x_i), \tag{5}$$

– wavelet-neuron of second cascade

$$\hat{y}^{[2]} = \sum_{i=1}^{n}\sum_{j=1}^{h}w_{ji}^{[2]}\varphi_{ji}(x_i) + \sum_{j=1}^{h}w_{j,n+1}^{[2]}\varphi_{j,n+1}(\hat{y}^{[1]}), \tag{6}$$

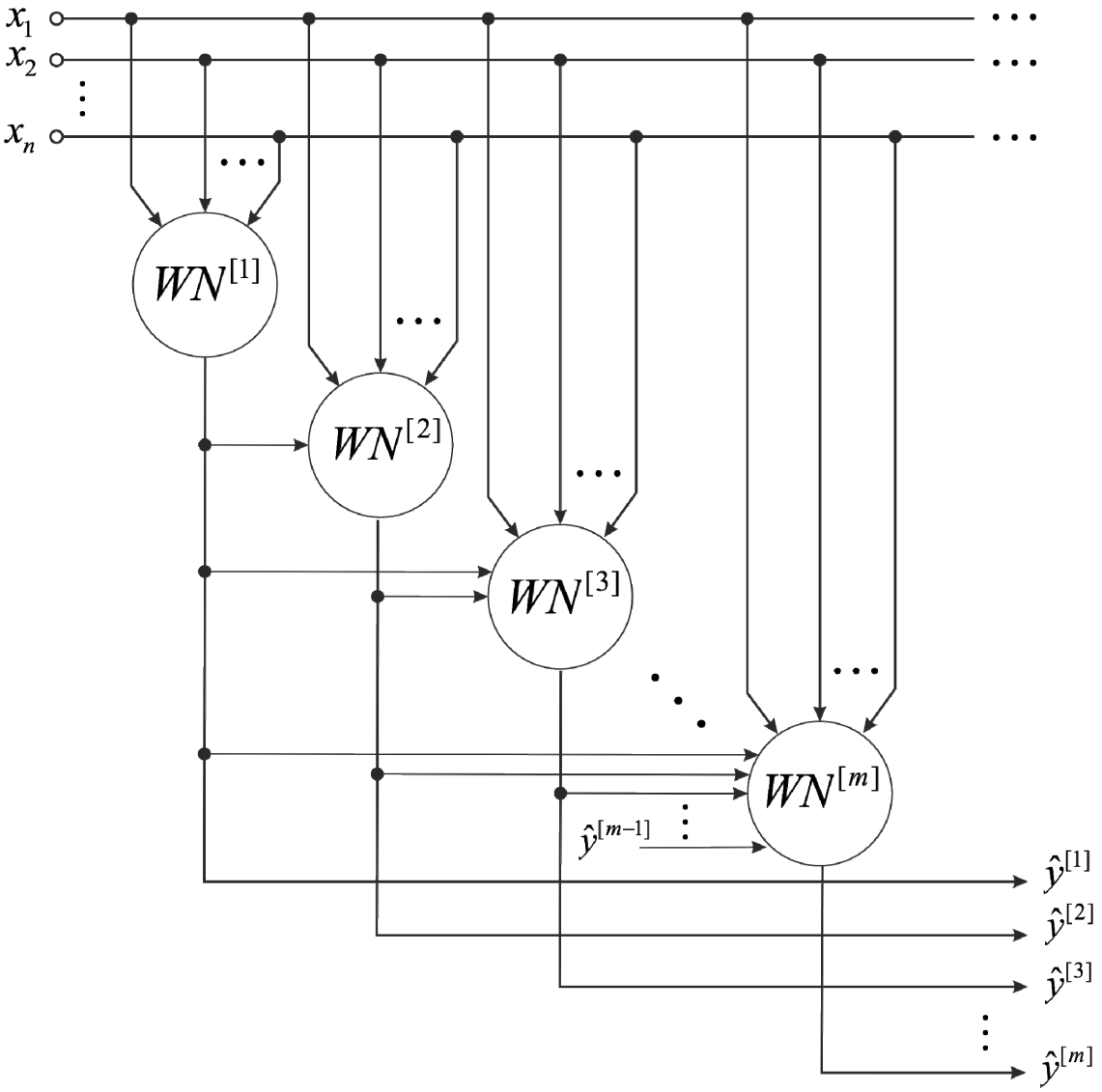


Figure 2. The cascade wavelet neural network

– wavelet-neuron of third cascade

$$\hat{y}^{[3]} = \sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}^{[3]}\varphi_{ji}(x_i) + \sum_{j=1}^{h} w_{j,n+1}^{[3]}\varphi_{j,n+1}(\hat{y}^{[1]}) + \sum_{j=1}^{h} w_{j,n+2}^{[3]}\varphi_{j,n+2}(\hat{y}^{[2]}), \tag{7}$$

– wavelet-neuron of m-th cascade

$$\hat{y}^{[m]} = \sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}^{[m]}\varphi_{ji}(x_i) + \sum_{l=n+1}^{n+m-1}\sum_{j=1}^{h} w_{jl}^{[m]}\varphi_{jl}(\hat{y}^{[l-n]}). \tag{8}$$

Leaning algorithms for hybrid cascade wavelet–neural network

The cascade wavelet neural network learning is performed in the batch mode using full training set $\{x(1), y(1); x(2), y(2); ...; x(k), y(k); ...; x(N), y(N)\}$.

At the beginning a set of wavelet-functions values (2) $\varphi^{[1]}(1), \varphi^{[1]}(2), ..., \varphi^{[1]}(N)$ is calculated for each training sample. For the learning process of cascade wavelet-neural network the parameters of wavelet activation-membership function $c_{ji}(k), \sigma_{ji}(k)$ and $\alpha_{ji}(k)$ are defined by clustering procedure or can be defined based on uniform grid.

Then using direct minimization of the learning criterion

$$E_N^{[1]} = \frac{1}{2}\sum_{k=1}^{N} e_1(k)^2 = \frac{1}{2}\sum_{k=1}^{N}(y(k) - \hat{y}_1(k))^2, \tag{9}$$

the vector of synaptic weights can be computed as

$$w^{[1]}(N) = \left(\sum_{k=1}^{N}\varphi^{[1]}(k)\varphi^{[1]T}(k)\right)^{+}\sum_{k=1}^{N}\varphi^{[1]}(k)y(k) = P^{[1]}(N)\sum_{k=1}^{N}\varphi^{[1]}(k)y(k), \tag{10}$$

where $(\bullet)^{+}$ is symbol of Moore-Penrose pseudoinverse.

If dimension of this vector is sufficiently large it is suitable to use procedure based on recursive least squares method with sequential training samples processing:

$$\begin{cases} w^{[1]}(k+1) = w^{[1]}(k) + \dfrac{P^{[1]}(k)(y(k+1) - w^{[1]T}(k)\varphi^{[1]}(k+1))}{1 + \varphi^{[1]T}(k+1)P^{[1]}(k)\varphi^{[1]}(k+1)}\varphi^{[1]}(k+1), \\ P^{[1]}(k+1) = P^{[1]}(k) - \dfrac{P^{[1]}(k)\varphi^{[1]}(k+1)\varphi^{[1]T}(k+1)P^{[1]}(k)}{1 + \varphi^{[1]T}(k+1)P^{[1]}(k)\varphi^{[1]}(k+1)},\ P^{[1]}(0) = \gamma I, \end{cases} \tag{11}$$

where γ is sufficiently large positive number which defined empirically, I is unity matrix (of appropriate dimensionality).

It is necessary to notice that using procedures (10), (11) for adjusting weight coefficients essentially reduces learning time in comparison with gradient algorithms underlying delta-rule.

After first cascade learning completion, synaptic weights of the neuron WN_1 become 'frozen' and second cascade of network consisting from a single neuron WN_2 is generated. It has one additional input for the output signal of the first cascade. Then procedures (10), (11) again are applied for adjusting of weight coefficients vector $w^{[2]}$, with dimensionality $(h+1)(n+1)\times 1$.

The neural network growing process (increasing cascades number) continues until we obtain required precision of the solved problem's solution, and for the adjusting weight coefficients of the last *m*-th cascade following expression are used:

$$w^{[m]}(N)=\left(\sum_{k=1}^{N}\varphi^{[m]}(k)\varphi^{[m]T}(k)\right)^{+}\sum_{k=1}^{N}\varphi^{[m]}(k)y(k)=P^{[m]}(N)\sum_{k=1}^{N}\varphi^{[m]}(k)y(k) \tag{12}$$

in batch mode,

$$\begin{cases} w^{[m]}(k+1)=w^{[m]}(k)+\dfrac{P^{[m]}(k)(y(k+1)-w^{[m]T}(k)\varphi^{[m]}(k+1))}{1+\varphi^{[m]T}(k+1)P^{[m]}(k)\varphi^{[m]}(k+1)}\varphi^{[m]}(k+1), \\ P^{[m]}(k+1)=P^{[m]}(k)-\dfrac{P^{[m]}(k)\varphi^{[m]}(k+1)\varphi^{[m]T}(k+1)P^{[m]}(k)}{1+\varphi^{[m]T}(k+1)P^{[m]}(k)\varphi^{[m]}(k+1)} \end{cases} \tag{13}$$

or

$$\begin{cases} w^{[m]}(k+1)=w^{[m]}(k)+\dfrac{e(k)\varphi^{[m]}(x(k))}{\eta^{[m]}(k)}, \\ \eta^{[m]}(k+1)=\beta\eta^{[m]}(k)+\left\|\varphi^{[m]}(x(k))\right\|^{2},\ 0\le\beta\le 1 \end{cases} \tag{14}$$

in on-line mode of information processing.

The main disadvantage of conventional cascade-correlation network is their ability of the batch mode learning usage, when all training set should be given a-priori. Cascade wavelet neural network can be trained in on-line mode, because of algorithm (13), (14) possess maximal possible squared rate of convergence. In this case at the first step architecture consisting of *m* cascades is generated. Each cascade trains using proper algorithm. Since outputs of the previous wavelet-neurons become additional inputs for the *m*-th cascade, algorithm realizes recurrent method of the prediction error, well known in the theory of adaptive identification. Changing cascades number during learning process also can be easily performed.

Conclusion

In the paper hybrid cascade wavelet neural network is proposed. It differs from its prototype, cascade-correlation learning architecture, in increased speed of operation, numerical stability and real-time processing possibility. Theoretical justification and experiment results confirm the efficiency of developed approach.

Acknowledgement

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

[Avedjan, 1999] E.D. Avedjan, G.V. Barkan, I.K. Levin Cascade Neural Networks. Avtomatika i Telemekhanika, 3, 1999. P.38-55. (In Russian)

[Bodyanskiy, 2000] Ye. Bodyanskiy, O. Mykhalev, I. Pliss Adaptive fault detection in control plants using artificial neural networks. Dnipropetrovsk: Systemni tekhnologiy, 2000. (In Ukrainian)

[Bodyanskiy, 2003] Ye. Bodyanskiy, O. Vynokurova Adaptive wavelet-neuro-predictor. Problemy bioniki, 58, 2003, P. 10-17. (In Russian)

[Bodyanskiy, 2004a] Ye. Bodyanskiy, O. Rudenko Artificial neural networks: architectures, learning and application. Kharkiv:TELETEKH, 2004. (In Russian)

[Bodyanskiy, 2004b] Ye. Bodyanskiy, V. Kolodyazhniy, I. Pliss, O. Vynokurova Learning wavelet neuron based on the RASP-function // Radioelectronics. Informatics. Control, 1 (11), 2004, P. 118-122.

[Bodyanskiy, 2005] Ye. Bodyanskiy, N. Lamonova, I. Pliss, O. Vynokurova An adaptive learning algorithm for a wavelet neural network. Blackwell Synergy: Expert Systems, 22(5), 2005, P. 235-240.

[Bodyanskiy, 2006] Ye. Bodyanskiy, I. Pliss, O. Slipchenko Growing neural networks based on orthogonal activation functions. Proc. XII-th Int. Conf. ITA KDS. Varna, 2006. P84-89.

[Bodyanskiy, 2007] Ye. Bodyanskiy, I. Pliss, O. Slipchenko Growing neural network using nonconventional activation functions. Int. J. Information Theories & Applications, 14, 2007. P.275-281.

[Bodyanskiy, 2008a] Ye. Bodyanskiy, A. Dolotov, I. Pliss, Ye. Viktorov The cascade orthogonal neural network. Advanced Research in Artificial Intelligence. Bulgaria, Sofia: Institute of Informational Theories and Applications FOI ITHEA., 2, 2008, P. 13-20.

[Bodyanskiy, 2008b] Ye. Bodyanskiy, O. Vynokurova, E. Yegorova Radial-basis-fuzzy-wavelet-neural network with adaptive activation-membership function. Int. J. on Artificial Intelligence and Machine Learning, 8, 2008, P. 9-15.

[Fahlman, 1990] S.E. Fahlman, C. Lebiere The cascade-correlation learning architecture. Advances in Neural Information Processing Systems. Ed. D.S. Touretzky. San Mateo, CA: Morgan Kaufman, 1990. P.524-532.

[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.

[Mitaim, 1997] S. Mitaim, B. Kosko Adaptive joint fuzzy sets for function approximation. Proc. Int. Conf. on Neural Networks, 1997, P. 537-542.

[Otto, 2003] P. Otto, Ye. Bodyanskiy, V. Kolodyazhniy A new learning algorithm for a forecasting neuro-fuzzy network. Integrated Computer-Aided Engineering. Amsterdam: IOS Press, 10 (4), 2003, P. 399-409.

[Rosenblatt, 1964] F. Rosenblatt The perceptron - a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

[Rosenblatt, 1966] F. Rosenblatt The perceptron: A probabilistic model for Information storage and organization in the brain, Cornel aeronautical laboratory. Psychological Review, 65 (6), 1958, P. 386-408.

[Schalkoff, 1997] R.J. Schalkoff Artificial Neural Networks. N.Y.: The McGraw-Hill Comp., Inc., 1997.

[Shepherd, 1997] A. J. Shepherd Second-Order Methods for Neural Networks. London: Springer–Verlag, 1997.

[Uchino, 1997] E. Uchino E., 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

Bodyanskiy Yevgeniy - *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, Ukraine 61166, Tel +380577021890, e-mail: bodya@kture.kharkov.ua*

Major Fields of Scientific Research: Hybrid Systems of Computational Intelligence

Kharchenko Oleksandra *– Student, Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine 61166, Tel +380577021890*

Major Fields of Scientific Research: Hybrid Wavelet-Fuzzy-Neuro-Techniques, Computational Intelligence.

Vynokurova Olena *– Candidate of Technical Sciences, Leading Researcher of the Control Systems Research Laboratory, Associate Professor of Information Technologies Security Department, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine, 61166, Tel +380577021890, e-mail: vinokurova@kture.kharkov.ua*

Major Fields of Scientific Research: Hybrid Wavelet-Fuzzy-Neuro-Techniques, Computational Intelligence.

EFFECTIVE ENERGY RECOMPUTATION FOR LOW AUTOCORRELATION BINARY SEQUENCE PROBLEM

Leonid Hulianytskyi, Vladyslav Sokol

Abstract:*This paper deals with finding low autocorrelation binary sequences which is a hard combinatorial optimization problem. Recent developments in this area analyzed, in order to understand characteristics of a problem. Emphasis is put on effective energy recomputation operators. Different types of these operators are tested to achieve full picture of LABS solvers development process. It is shown that latest state-of-the-art metaheuristics in fact all based on simplest Tabu Search framework, achieving performance boost by means of energy recomputation operators' optimization. In this paper we construct variation of memetic algorithmincorporating latest developments to reach higher performance than it's original. Comparison to a state-of-the-art TSv7 approach completed on instances with known optimums as well as on some unsolved larger ones. It is concluded that these approaches shows similar performance as they both have the same built-in heuristic. As further research proposed a comparison of different metaheuristic frameworks applied to this problem.*

Keywords: *combinatorial optimization, low autocorrelation binary sequences, memetic algorithm,stochastic local search, tabu search.*

ACM Classification Keywords: *G. 1. 6. Mathematics of Computing, Numerical Analysis, Optimization.*

Introduction

Findinglow autocorrelation binary sequence (LABS)is a hard combinatorial optimization problem withmany applications in different areas such as telecommunications, physics and chemistry (see Ref.[Gallardo, Cotta, Fernandez, 2009]). It has been deeply studied since the 1960sby both the communities of Physics and Artificial Intelligence.

Binary sequence $S$of length Lrepresented as$s_1 s_2 \dots s_L$, with $s_i \in \{-1,1\}$for $1 \leq i \leq L$, i.e.,$S \in \{-1,1\}^L$. The aperiodic autocorrelation of elements in sequence $S$with distance $k$is defined as:

$$C_k(S) = \sum_{i=1}^{L-k} s_i s_{i+k}. \quad (1)$$

The energy function associated with sequence S is the quadratic sum of its correlations:

$$E(S) = \sum_{k=1}^{L-1} C_k^2(S). \quad (2)$$

The LABS problem with length L,or$LABS(L)$, liesin finding a binary sequence of length L with associated minimum energy.

Related work

For the last 3 decades the LABS problem has been constantly tackled in the literature using exact and heuristic methods (Ref. [Gallardo, Cotta, Fernandez, 2009]).

In [Golay, 1982]for the first time, were published optimal solutions for$L \in [6,32]$ that author computed by performing an exhaustive search enumeration.

Mertens in[Mertens, 1996]used a parallel branch and bound algorithm with symmetry breakingprocedures to solve instances up to $L = 60$.

Presented in [Dotu, van Hentenryck, 2006]stochastic local search (SLS) - Tabu Search (TS) algorithm, was capable of reaching global optimums (GO) for$6 \leq L \leq 48$ faster than exact enumeration approaches.

Gallardo et al.([Gallardo, Cotta, Fernandez, 2007],[Gallardo, Cotta, Fernandez, 2009]) developed Memetic Algorithm (MA) endowed with TS mechanism inspired by [Dotu, van Hentenryck, 2006]. Results shows that proposed algorithm worked as state-of-the-art at time, finding optimal solutions for instances with known optima faster than predecessors overall by at least an order.

In[Halim, Yap, Halim, 2008] authors adopt and significantly improve TS framework of [Dotu, van Hentenryck, 2006] so it performs actually even better than recent state-of-the-art MA. According to authors, such major improvements became available after deep analysis of search trajectory, and as a result more effective diversification part of their metaheuristic.

In this paper we will try to understand nature of such performance boost in both algorithms of MA and TSv7. Moreover we will use gathered data to construct effective framework that outperforms previous approaches.

Memetic Algorithm

MA presented in [Gallardo, Cotta, Fernandez, 2007], [Gallardo, Cotta, Fernandez, 2009] is a metaheuristic consisting of GeneticAlgorithm and TSbuilt inside it. TS in this approach is quite similar to one proposed earlier in [Dotu, van Hentenryck, 2006] and have basic structure as one can find in simplestTS implementation. At each step algorithm moves to the best (in terms of energy) solution in neighborhood, even if its energy is higher, using a short-term memory (tabu list) to avoid cycling.

Authors of MA performed calibration of framework parameters to hit better performance, but key improvement of their approach was lying in effective energy recomputation at each step. Instead of naive full energy recalculation (as in (2), $O(L^2)$) every time algorithm wanted to estimate move to another solution, authors proposed more efficient incremental operator ($O(L)$). Tradeoff asalways in such cases in memory (which we actually have enough): solution besides actual sequence and energy value contains two additional data structures: matrix $T \in (L-1) \times (L-1)$, and vector $C \in (L-1)$. $T(S)$stores all computed products such that $T(S)_{ij} = s_j s_{j+i}$for $j \leq L-1$, when $C(S)$containsvalues of the different correlationsC_k (as in (1))(Figure1).

	1	2	3	4	
1	s_1s_2	s_2s_3	s_3s_4	s_4s_5	$s_1s_2 + s_2s_3 + s_3s_4 + s_4s_5$
2	s_1s_3	s_2s_4	s_3s_4		$s_1s_3 + s_2s_4 + s_3s_4$
3	s_1s_4	s_2s_5			$s_1s_4 + s_2s_5$
4	s_1s_5				s_1s_5

Figure 1.Example of additional data structures $T(S)$and $C(S)$ for $L = 5$ instance.

Regarding this, an efficient operator for computing energy of solution, 1 bit different from current, can be used (Figure2).

```
        FlipedEnergy(L, T, C, i)
        {
flipedEnergy = 0
fork = 1toL - 1 do
            {
                c = C[k]
if (i - k>= 1)
                    c = c - 2 * matrix[k, i –k]
if (i + k<=L)
                    c = c - 2 * matrix[k, i]
          flipedEnergy = flipedEnergy + c * c
            }
return flipedEnergy
        }
```

Figure 2.Pseudocode of an efficient energy recomputation operator.

After selecting best move in neighborhood algorithm as well updates additional data structures using similar $O(L)$-performance operator (Figure 3).

```
        Update(L, T, C, i)
        {
fork = 1toL - 1do
            {
if (i - k>= 1)
                {
C[k] = C[k] - 2 * T[k, i - k]
T[k, i - k] = -T[k, i - k]
                }
if (i + k<= L)
                {
C[k] = C[k] - 2 * T[k, i]
T[k, i] = -T[k, i]
                }
            }
        }
```

Figure 3.Pseudocode of a selected move update operator.

Data structures $T(S)$ and $C(S)$ initialized once, at the beginning of each TS procedure.

We conducted several experiments to see impact of such energy recalculation on overall performance of this metaheuristic. Here and further experiments were performed on instances of sizes $L \in [31,60]$ for which global optimums (GO) are known (http://www-e.uni-magdeburg.de/mertens/research/labs/open.dat), on a Core i7-860 2.8 GHz PC, programmed in C# (source code available upon email request).Termination criteria for each algorithm set as finding a GO. Results are - timings in seconds, showing how long it takes to reach GO. Each experiment was repeated 10 times for statistical estimation.

Table 1 and Figure 4 show results obtained by Memetic Algorithm with full energy recomputation (MAf) and original MA with effective incremental energy recomputation.

Table 1. Average performance of MAf, MA in seconds for each instance size L.

L	MAf	MA	L	MAf	MA	L	MAf	MA
31	0.33	0.07	**41**	74.38	15.33	**51**	3889.45	478.4
32	1.17	0.19	**42**	74.86	5.34	**52**	1602.03	241.28
33	1.59	0.52	**43**	429.34	35.55	**53**	1998.13	108.38
34	10.3	0.36	**44**	117.05	13.92	**54**	8536.41	234.62
35	18.27	1.97	**45**	73.79	23.24	**55**	4738.36	600.05
36	6.61	0.89	**46**	110.5	9.72	**56**	5207.26	901.57
37	12.25	1.17	**47**	110.17	12.49	**57**	25598.61	1235.28
38	17.16	1.75	**48**	650.68	44.76	**58**	26376.22	1231.7
39	67.83	5.64	**49**	468.51	34.22	**59**	12105.19	966.71
40	32.17	3.95	**50**	204.1	41.21	**60**	15240.15	1813.36

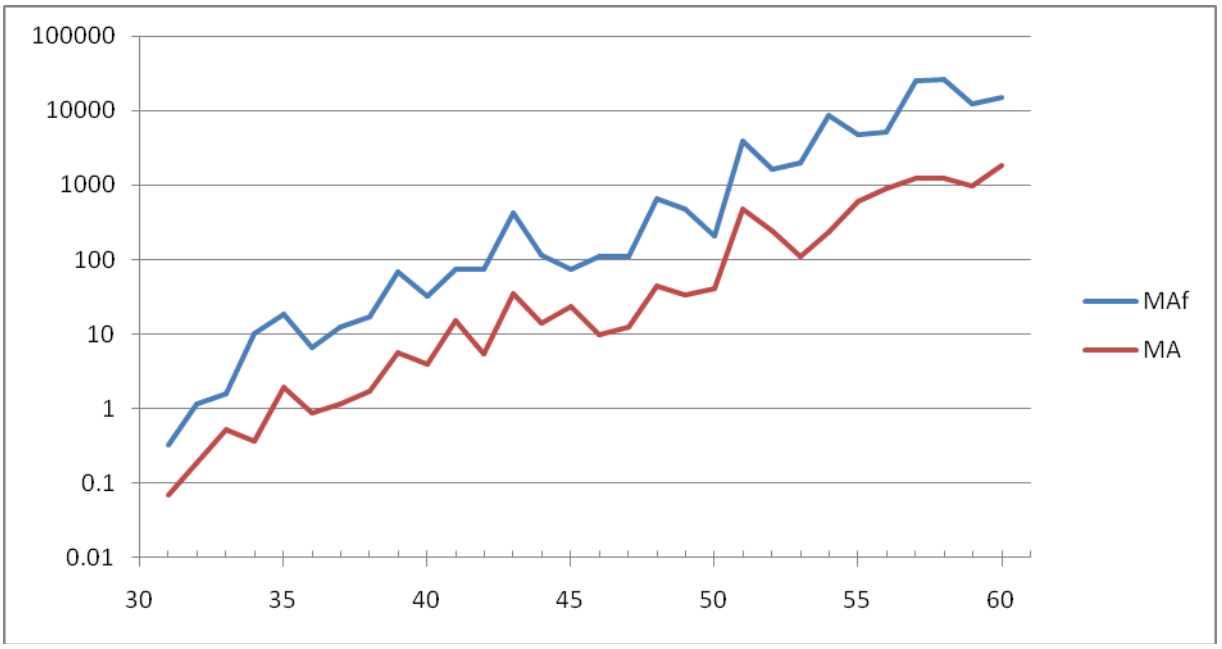


Figure 4. Average performance of MAf, MA in seconds for each instance size L, logarithmic scale.

As we can see changing only the energy recomputation results in a major speed boost of about 10 times in average for this set of instances. This result confirms obvious conclusion that energy calculation takes most of CPU time used by heuristics.

Tabu Search

In [Halim, Yap, Halim, 2008] proposed TS similar to [Dotu, van Hentenryck, 2006] with some major improvements (called TSv7).First of all,an efficient incremental recomputation of energy, inspired by MA were used. Next, according to authors such simple diversification as random restarts used in [Dotu, van Hentenryck, 2006]was not efficient due to some features of a search space. As we can see binary sequence have the same energy as negated or inversed one, making each instance of problem to have at least 4 GO. Authors of [Halim, Yap, Halim, 2008] provided some empirical evidence, achieved with exact enumeration for smaller instances, that current solution of TS is always much closer(in terms of Hamming distance) to one GO than to others. Thus, it was suggested to make small local restarts instead of full random reconfiguration, each time heuristic completes intensification phase. Experiments showed that such approach could find optimal solutions approximately 2 times faster than previous state-of-the-art MA. Note that original TS realization in [Halim, Yap, Halim, 2008] called TSv1 actually showed similar effectiveness as TSv7, outperforming MA in about an order of 2 in average (yet slightly inferior comparing to TSv7).

When programming TSv7 approach on our platform we examine original source code (thanks to authors for sharing it). It is come to our attention that energy recomputation used in this approach is slightly different from the way proposed in MA. Authors put into use one more additional data structure $T2(S) \in (L) \times (L)$ that stores some temporary calculations used in FlipedEnergy() operator. Such manipulations helped to optimize FlipedEnergy() operator by transferring some computational complexity to another operator Update(). Note that Tabu Search approach on each step uses approximately L times FlipedEnergy() operator and only once Update() operator when selected best solution from neighborhood, thus making this optimization useful. Initialization of $T2(S)$ conducted once after every reconfiguration of sequence (Figure 5).

```
for i = 0toL -1 do
            {
k = 1
do
                {
T2[k, i] = 0
if (i -k>= 1)
T2[k, i] = T2[k, i] + 2 * T2[k, i - k]
if (i + k<=L)
T2[k, i] = T2[k, i] + 2 * T2[k, i]
k = k + 1
                }
whileNOT(i - k< 0 ANDi + k >= L)
            }
```

Figure5.Pseudocode of $T2(S)$ structure initialization.

FlipedEnergy() and Update() operators then can be successfully changed to achieve maximal performance of algorithm (Figure 6, Figure 7).

```
        FlipedEnergy(L, T2, C, i)
        {
flipedEnergy = 0
fork = 1toL- 1do
            {
                c = C[k] - T2[k, i]
                flipedEnergy = flipedEnergy + c * c
            }
return flipedEnergy
        }
```

Figure 6. Pseudocode of an efficient energy recomputation operator, optimized with usage of an additional data structure $T2(S)$, instead of $T(S)$.

```
        Update(L, T, T2, C, i)
        {
forkk = 1to L - 1do
            {
C[k] = C[k] - T2[k, i]
if (i - k>= 1)
                {
T2[k, i] = T2[k, i] - 4 * T[k, i - k]
T2[k, i - lag] = T2[k, i - k] - 4 * T[k, i - k]
T[k, i - lag] = -T[k, i - k]
                }
if (i + k<= L)
                {
T2[k, i] -= 4 * T[k, i]
T2[k, i + k] -= 4 * T[k, i]
T[k, i] = -T[k, i]
                }
            }
        }
```

Figure 7. Pseudocode of update operator for data structures after successful move selection.

We conducted some experiments to show importance of such optimization. For this we programmed 3 versions of TSv7 respectively: with full energy recomputation on each step (TSv7f), with efficient incremental energy recomputation (TSv7e) as in MA, and original with effective optimized energy recomputationTSv7 (Table 2).

Table 2. Average performance of TSv7f, TSv7e,TSv7 in seconds for each instance size L.

L	TSv7f	TSv7e	TSv7	L	TSv7f	TSv7e	TSv7	L	TSv7f	TSv7e	TSv7
31	0.25	0.06	0.04	**41**	113.85	10.05	5.05	**51**	3110.59	550.44	147.01
32	1.06	0.1	0.04	**42**	94.06	8.44	4.81	**52**	5172.52	408.45	245.52
33	5.22	0.37	0.16	**43**	614.64	44.48	39.48	**53**	4147.09	223.11	163.5
34	2.41	0.42	0.15	**44**	120.75	7.82	8.07	**54**	3028.22	169.24	102.18
35	5.77	0.62	0.56	**45**	292.2	11.37	5.21	**55**	5122.86	370.79	173.83
36	5.5	0.39	0.48	**46**	26.88	9.03	3.15	**56**	12113.13	523.82	656.46
37	18.87	0.98	0.64	**47**	128.52	17.72	5.35	**57**	26773.11	1817.93	746.44

38	20.72	1.31	1.47	**48**	487.03	39.84	14.95	**58**	21412.45	1853.58	890.86
39	51.5	4.14	2.8	**49**	402.59	43.98	26.9	**59**	3289.51	1021.54	632.7
40	55.96	2.97	1.85	**50**	313.61	20.29	19.69	**60**	4535.86	1535.78	753.22

Figure5 shows performance of both sets of algorithms (MAs' and TSv7s').

Figure 5. Average performance of MAf, MA,TSv7f, TSv7e,TSv7 in seconds for each instance size L, logarithmic scale.

We can split algorithms into 3 visual groups by performance. The weakest two approaches are both algorithms with full energy recalculation showing approximately same average time to find GO. The next group contains of MA and TSv7e both of which got efficient recomputation of energy. These approaches are as expected much more efficient then the first two, and again showing same performance. Last algorithm - TSv7 indeed outperforms all opponents, having more optimized effective recomputation comparing to MA and TSv7e.

Optimized Memetic Algorithm

Considering results obtained above we decided to improve performance of MA approach (we will call it OMA). We added optimized energy recomputation as in TSv7. Then we reconfigured some parameters aiming to a better performance. GA used in MA was under next parameter settings: population size – 100, crossover rate 0.9, mutation rate – 1/L, binary tournament selection, uniform crossover. After empirical adjustments we decided to change mutation rate to 0.01 so mutation frequency does not degrade with increase of L. Next step was reconfiguration of embedded TS heuristic. We changed iterations parameter from 3*l/2 in MA to 4*L in our OMA, as experiments shows that it is worth to spend additional time to intensification. Tabu tenure parameters set as following: minimal tabu tenure set as 1+L/20, random additional tabu tenure set as L/10. Exact same parameter values were used in TSv7 which is another empirical evidence of effectiveness of such calibration. Table 3 shows results of experiments on instances with known optima.

Table 3. Average performance of OMA in seconds for each instance size L.

L	OMA	L	OMA	L	OMA
31	0.04	**41**	5.09	**51**	314.26
32	0.2	**42**	5.51	**52**	139.15
33	0.22	**43**	11.19	**53**	67.38
34	0.3	**44**	6.86	**54**	72.03
35	0.38	**45**	5.22	**55**	169.76
36	0.38	**46**	2.43	**56**	471.47
37	1.13	**47**	6.61	**57**	534.22
38	0.59	**48**	25.79	**58**	952.94
39	3.64	**49**	16.5	**59**	634.33
40	1.38	**50**	32.5	**60**	638.54

Figure6 shows performance ofOMA in comparison to original TSv7 which is current state-of-the-art approach.OMA performance is comparable to TSv7 but slightly better in average.

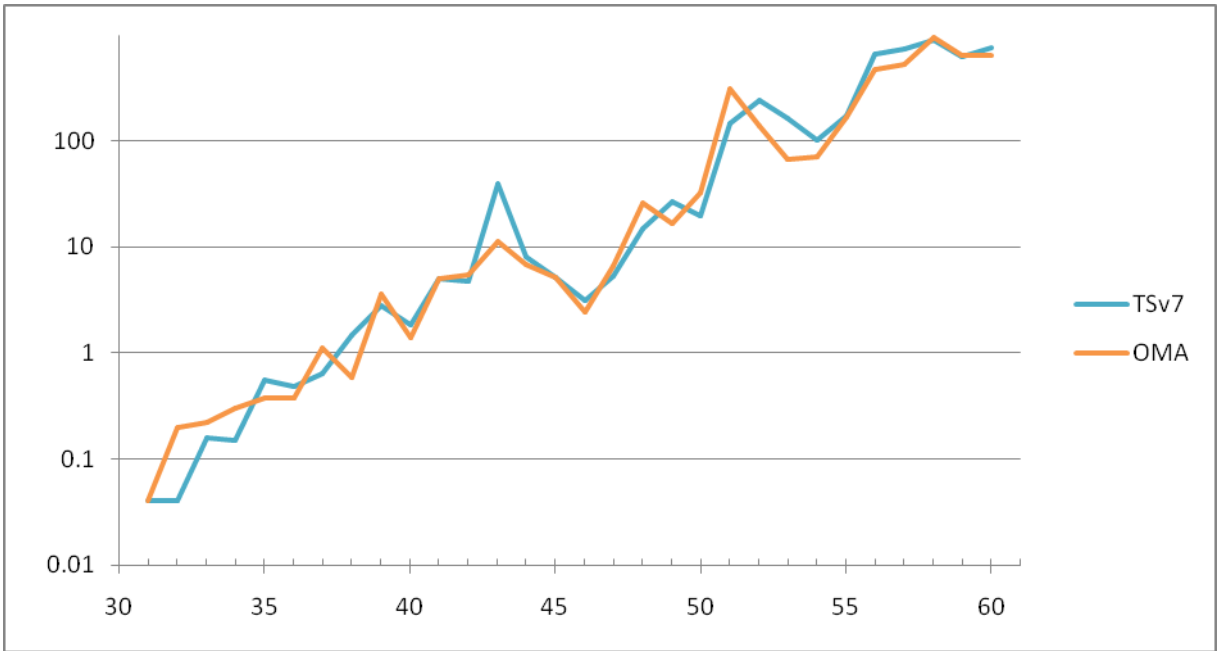


Figure 6. Average performance of TSv7, OMA in seconds for each instance size L, logarithmic scale.

Experiments on larger instances

Another important part of OMA and TSv7 effectiveness comparison would be experiments on instances for which GO are not found yet or not proven. For this analysis we selected 10 next instances $L \in [61,70]$. Termination criteria for both approaches was set as time limit and defined as follows. Previous experiment showed us average time needed by metaheuristics to reach GO at specified L. We can approximate time with linear trend and make forecast to achieve time limit formula we need. For this purpose we used results for instances $L \in [51,60]$.

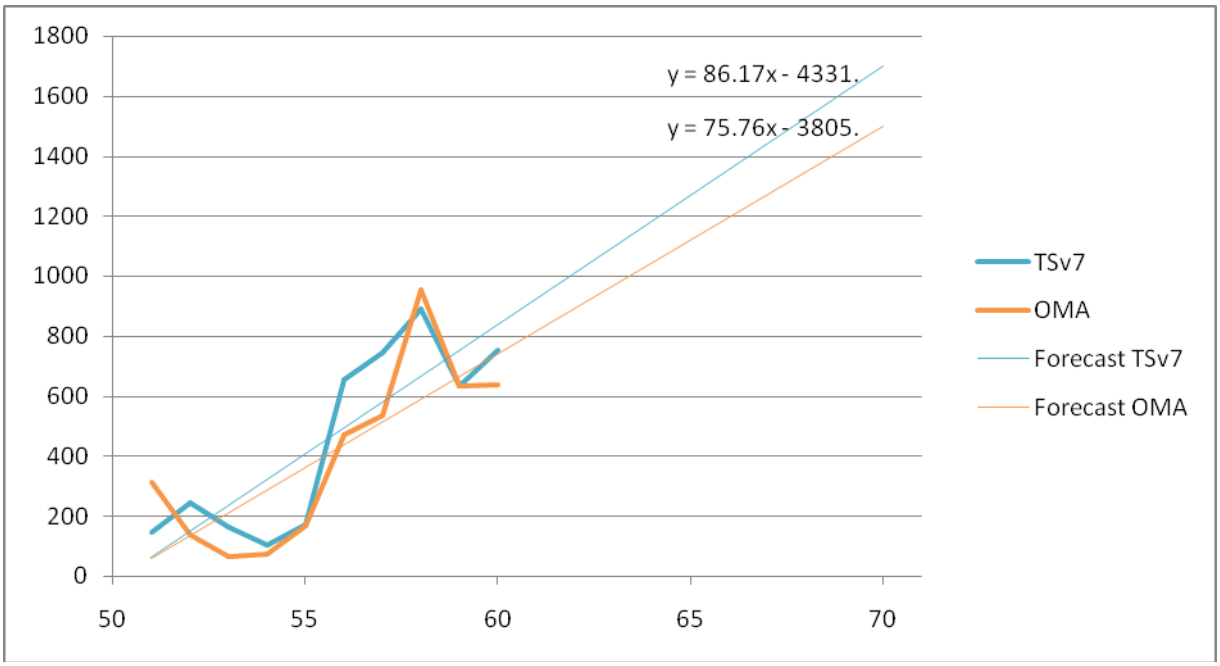


Figure 7. Average performance of TSv7, OMA and their forecasted values in seconds for each instance size L, normal scale.

We take higher forecast (achieved from TSv7 trend) as our time limit:

$$\mathrm{timeLimit}(L) = 86.172L - 4331.4\ (s). \quad (3)$$

Thus, time limit varies from 925 seconds for $L = 61$, to 1700 seconds for $L = 70$.

Table 4 and Figure 8 show results of this type of experiments.

Table 4. Average energy of best solution achieved by TSv7 and OMA for each instance size L.

L	TSv7	OMA
61	229.6	229.2
62	239	239.8
63	228.6	221
64	250.4	228
65	266.4	267.6

66	267.4	267.4
67	275	271.4
68	276	281.2
69	301.6	290.8
70	327	311

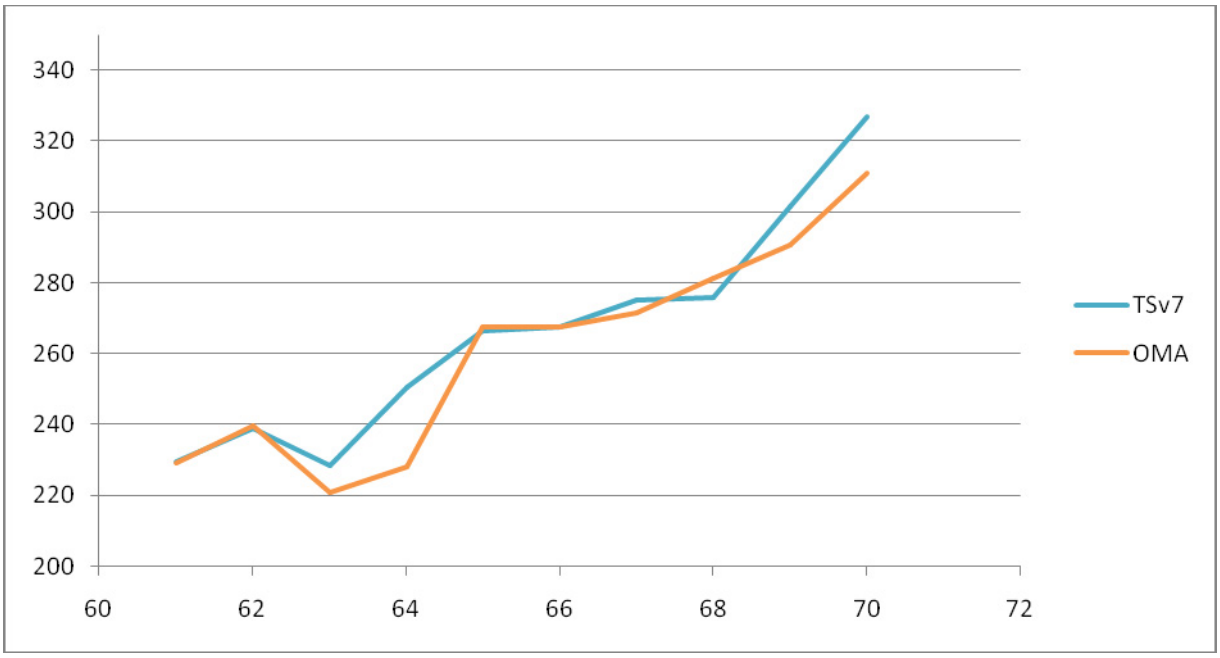


Figure 8. Average energy of best solution achieved by TSv7 and OMA for each instance size L.

As we can see performance of both approaches on larger instances also quite similar with some advantage of OMA in average. This shows that performance of both TSv7 and OMA changes identically with growth of complexity of problem. This is partially due to usage of TS local search mechanism as main component of bothmetaheuristics.

Conclusion

We showed importance of effective energy recomputation during stochastic local search applied to LABS problem.Progress in reducing time cost of these operator made by authors [6,7,8] allowed to bring LABS solvers to a new level of effectiveness. Though, metaheuristic strategy staid the same throughout these improvements – TS mechanism played key role in most of the approaches.

We developed new variant of memetic algorithm which incorporates best from both recent state-of-the-art approaches. As it performs on same level of effectiveness with latest TSv7 solver, we conclude that new LABS problem heuristics should use developments in energy recomputation to at least reach current state-of-the-art performance. Yet, comparison of different metaheuristic strategies applied to LABS problem is an open question.

Bibliography

[Gallardo, Cotta, Fernandez, 2009] J.E.Gallardo, C.Cotta, A.J.Fernandez. Finding Low Autocorrelation BinarySequences with Memetic Algorithms. In: Applied Soft Computing, 2009,vol. 9, pp. 1252–1256.

[Golay, 1982] M.J.E.Golay. The merit factor of long low autocorrelation binary sequences.In: IEEE Transactions on Information Theory, 1982,vol. 28(3), pp. 543–549.

[Mertens, 1996] S.Mertens. Exhaustive search for low-autocorrelation binary sequences.In: Journal of Physics A: Mathematical and General, 1996,vol. 29, pp. 473–481.

[Dotu, van Hentenryck, 2006] I.Dotu, P.vanHentenryck.A Note on Low Autocorrelation Binary Sequences. In:Constraint Programming, 2006, pp. 685–689.

[Gallardo, Cotta, Fernandez, 2007] J.E.Gallardo, C.Cotta, A.J.Fernandez.A Memetic Algorithm for the Low Autocorrelation Binary Sequence Problem. In: GECCO, 2007, pp. 1226–1233.

[Halim, Yap, Halim, 2008] S.Halim, R.Yap, F.Halim. EngineeringStochastic Local Search for the Low Autocorrelation Binary Sequence Problem. In: Principles and Practice of ConstraintProgramming, Australia, Sydney, 2008.

Authors' Information

Leonid Hulianytskyi– *Dr.Sc.(Technology), head of Department of combinatorial optimization and intelligent information technology of V.M.Glushkov Instituteof Cybernetics of National Academy of Sciences of Ukraine; Prof. of NTUU "KPI" and* Taras Shevchenko National University of Kyiv*; e-mail:*lh_dar@hotmail.com

Major Fields of Scientific Research: Combinatorial Optimization, Decision Making,Mathematical Modeling and practical applications

Vladyslav Sokol *– Master student in National Technical University of Ukraine "Kyiv Polytechnic Institute"; e-mail:* sokol.vladyslav@gmail.com

Major Fields of Scientific Research:General Computer Science, Combinatorial Optimization, Artificial Intelligence

OPTIMIZING PROGRAMMABLE LOGIC ARRAYS USING THE SIMULATED ANNEALING ALGORITHM

Liudmila Cheremisinova, Irina Loginova

Abstract: *In the paper the programmable logic array (PLA) topological optimization problem is dealt with using folding techniques. A PLA folding algorithm based on the method of simulated annealing is presented. A simulated-annealing PLA folding algorithm is presented for multiple unconstrained folding. Then, the algorithm is extended to handle constrained folding. In this way, simple folding is considered as a case of multiple constrained folding. Some experimental results of computer investigation of the suggested algorithms are given.*

Keywords: *design automation, area optimization, VLSI structure folding, simulated annealing.*

ACM Classification Keywords: *B.6.3 [Logic Design]: Design Aids – Optimization; B.7.2 [Integrated circuits]: Design Aids –Layout.*

Introduction

Structured logic refers to logic forms that exhibit a high degree of regularity in their layout and interconnections. Such layout is referred to a regular layout style that is constructed according to some definite architecture and established in advance for some structured logic. Typical widely used regular structures are Programmable Logic Arrays (PLA's), gate matrix arrays, Weinberger Arrays [Ullman, 1984]. All these forms have a two-dimensional structure consisting of a matrix of rows and columns. There are transistors in intersections of some rows and columns. The use of such regular structures makes it possible to automatically generate the layout from its functional description. The price paid for the structural regularity is a larger chip area because the layouts obtained are sparse: a large percentage of the row-column intersections are not personalized. In order one to have a clear idea of degree of structural rarity, it can be said that previous research has shown that on average, about half of the entries in the personality matrices of large structures in real circuits are not personalized. Several techniques have been proposed for reducing the area required.

Since matrix is the central part of any regular structure mentioned above, we consider further just it and then we show how to expand the obtained results to real regular structures taking into account the peculiarities of their layout and some other constraints resulted from the implementation circuit on the base of these array structures and the chosen type of topological minimization.

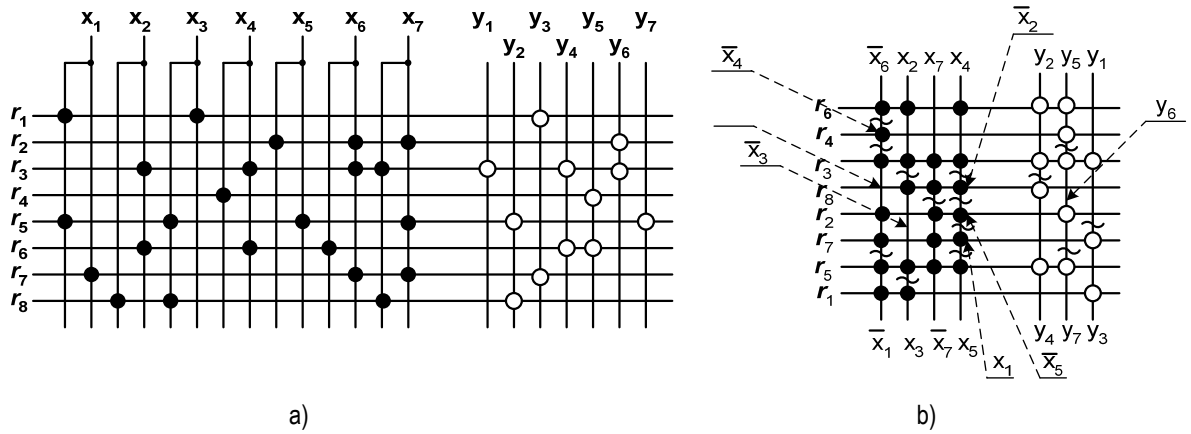
Techniques of topological minimization reduce the number of physical columns and/or rows, they change the PLA structure by using a procedure called folding [Hachtel, 1982], [DeMicheli, 1983]. Folding is a technology-independent transformation, it is developed for array structures and attempts to place two or more columns (and/or rows) together in the same physical vertical (and/or horizontal) line (signal bus) so that they can share this line. Column folding is said to be simple if utmost two columns (rows) share a single physical vertical (and/or horizontal) line. It is called multiple if more than two columns can share a single column.

Folding does not change the implemented logic in any manner, but reduces the number of columns (and/or rows), and thus reduces the area. In the paper we deal with array structure folding and its effects on its square.

[Wong, 88] proposes the use of simulated-annealing to solve the PLA folding problem for the case of multiple unconstrained folding. In this approach, entire solutions are analyzed one after another by producing different permutations of rows. In the paper we also propose the solution of multiple unconstrained and constrained folding problem and extend the suggested simulated-annealing algorithm to solve a special case of multiple constrained folding – simple folding. Then we give some experimental results of computer investigations of the suggested simulated-annealing algorithms based on simple and multiple PLA folding and comparative evaluation of their effects on PLA layout reduction.

Array structures style and their folding

Any array structure can be represented by specifying the positions of of their elements (transistors) in its plane. In the folding problem it can be described in symbolic form by a Boolean matrix $\boldsymbol{B}$ having sets $C(\boldsymbol{B})$ and $R(\boldsymbol{B})$ of their columns (where uncomplemented and complemented modes of a variable have their own distinct column) and rows. A 1 in the position i,j of the matrix $\boldsymbol{B}$ means that there is an appropriate crosspoint (transistor) between the i-row and the j-column in the matrix. Each column $c_i \in C(\boldsymbol{B})$ implies a set $R(c_i) \subseteq R(\boldsymbol{B})$ of rows, which are populated on it: $r_j \in R(c_i) \leftrightarrow b_j^i = 1$. For instance, for PLA on Figure 1 Boolean matrix $\boldsymbol{B}$ corresponding to AND-plane is depicted in Figure 2.



a) b)

Figure 1. An example of array-based structure: a) PLA; b) PLA multiple folded form

	$\bar{x}_1$ c_1	x_1 c_2	$\bar{x}_2$ c_3	x_2 c_4	$\bar{x}_3$ c_5	x_3 c_6	$\bar{x}_4$ c_7	x_4 c_8	$\bar{x}_5$ c_9	x_5 c_{10}	$\bar{x}_6$ c_{11}	x_6 c_{12}	$\bar{x}_7$ c_{13}	x_7 c_{14}
r_1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
r_2	0	0	0	0	0	0	0	0	1	0	0	1	0	1
r_3	0	0	0	1	0	0	0	1	0	0	0	1	1	0
r_4	0	0	0	0	0	0	1	0	0	0	0	0	0	0
r_5	1	0	0	0	1	0	0	0	0	1	0	0	0	1
r_6	0	0	0	1	0	0	0	1	0	0	1	0	0	0
r_7	0	1	0	0	0	0	0	0	0	0	0	1	0	1
r_8	0	0	1	0	1	0	0	0	0	0	0	0	1	0

Figure 2. Boolean matrix ***B*** reflecting the structure of PLA AND plane of Figure 1, a

The goal of the array structure folding is to find the maximum number of columns/rows that can be folded simultaneously.

Any two columns c_i and c_j are disjoint if $R(c_i) \cap R(c_j) = \varnothing$. Two disjoint columns both do not have transistors on any particular row of the array-based structure. A column folding list $l^c_k = (c_{k1}, c_{k2},\ldots, c_{km})$ is a set of pairwise disjoint columns c_{ij}, it is unordered in general case. An ordered column folding list $l^{co}_k = <c_{k1}, c_{k2},\ldots, c_{km}>$ is a column folding list l^c_k whose elements c_{ki} are ordered. An ordered column folding list (OCFL) l^{co}_k of cardinality two is an ordered column folding pair. Any OCFL $l^{co}_k = <c_{k1}, c_{k2},\ldots, c_{km}>$ can be actually implemented in the same vertical line of array-based structure moving c_{k1} above c_{k2}, c_{k2} above c_{k3}, an so on, $c_{k,m-1}$ above c_{km}. So OCFL l^{co}_k results to permutation on the set of rows: $R(c_{k1}) > R(c_{k2})$ – the rows of $R(c_{k1})$ are all above those of $R(c_{k2})$, $R(c_{k2}) > R(c_{k3})$ – $R(c_{k2})$ are all above those of $R(c_{k3})$, and so on, $R(c_{k,m-1}) > R(c_{k,m})$ – the rows of $R(c_{k,m-1})$ are all above those of $R(c_{km})$ inducing the relation on row set $R(\boldsymbol{B})$:

$$P^r(l^{co}_k) = \bigcup_{i,j} (R(c_{ki}) \times R(c_{kj})), \quad \text{i.e.} \quad P^r(l^{co}_k) = \{ r_p \times r_q / r_p \in R(c_{ki}), r_q \in R(c_{kj}), i < j \}.$$

This relation is partial because it is irreflexive, asymmetric and transitive by its definition.

An ordered column folding set (OCFS) $L^{co} = \{l^{co}_1, l^{co}_2,\ldots, l^{co}_k\}$ is a set of disjoint ordered column folding lists. The number k of columns entering into all OCFLs $l^{co}_i \in L^{co}$ is called the size of OCFS L^{co}. OCFS L^{co} induces a set of ordering relations among the rows that is the union of ordering relations induced by OCFLs l^{co}_k belonging to the OCFS $P^r(l^{co})$:

$$P^r(L^c_k) = \bigcup_{i=1}^{k} (P^r(l^{co}_i)).$$

This relation $P^r(L^{co})$ is irreflexive, asymmetric but not transitive in general case. The transitive closure $R^t(P^r)$ of $P^r(L^{co})$ is irreflexive, transitive but can be not asymmetric.

It is proven [DeMicheli, 1983] that an OCFS L^{co} is implementable topologically (by a folded array-based structure) if the transitive closure $R^t(P^r)$ of the relation $P^r(L^{co})$ is a partial ordering on $R(\boldsymbol{B})$, that is $R^t(P^r)$ is asymmetric. In other

words, an OCFS $L^{co}{}_k$ is implementable topologically if there exist linear order of the rows $R(\boldsymbol{B})$ extending the row ordering $P^r(L^{co})$.

An implementable OCFS L^{co} specifies the structure of the folded array, and its size is referred to as the size of the folding: the number of OCFLs in L^{co} corresponds to the number of columns that will replace $\sum_{i=1}^{k} |l^c_i|$ columns of the initial array-based structure.

An example of a PLA structure (in pictorial symbol) and its multiple folded form is depicted in Figure 1. Here the "breaks" of the lines that occur on the folded columns are designated by the symbol "~". The size of implementable OCFS L^{co} = {< $\bar{x}_6$, $\bar{x}_4$, x_6, $\bar{x}_1$>, <x_2, $\bar{x}_3$, x_3>, <x_7, $\bar{x}_7$>, <x_4, $\bar{x}_2$, $\bar{x}_5$, x_1, x_5>, <y_2, y_4>, <y_5, y_6, y_7>, <y_1, y_3>} (corresponding to the folded PLA) is equal seven and thus 7 columns of the folded regular structure replace 21 columns of the initial structure.

So the formal statement of optimal folding problem is as follows: given a Boolean matrix representing array-based structure, find an implementable ordered folding set of maximum size.

Simulated annealing formulation

Simulated annealing is a computer approach widely used to solve difficult optimization problems. Such an approach is applied to a wide variety of applications where the search for optimal solution is needed. Any problem requiring optimal or near optimal solution over a space formed from the combination of several variables is considered to be a combinatorial approximation problem of the following type [Lee, 1995]:

Minimize (maximize) $f(x)$, subject to $g_j(\boldsymbol{x}) = 0$, $j = 1, 2, \ldots,$ (1)

where $f(x)$ is the cost (or object) function over the vector of configuration variables x_i, and $g_j(x)$ are some constraints.

The first widely available publication on simulated annealing belongs to Kirkpatrick et al. [Kirkpatrick, 1983]. Simulated annealing, developed by Kirkpatrick et al. treats combinatorial approximation analogously to the annealing of metals. As in the actual annealing process (in which a good crystal structure is desired as the final result), simulated annealing requires a carefully controlled cooling schedule to avoid a bad final solution. The basic algorithm has been thoroughly discussed over the past several years (their overview could be found in [Lee, 1995], [Greening, 1995]).

In general case the combinatorial optimization problems could be presented as follows: there is a finite set G of states $g_i(\boldsymbol{x})$ where each state $g_i(\boldsymbol{x}) \in G$ is represented by n state variables: $g_i(\boldsymbol{x}) = (x_1, x_2, \ldots, x_n)$. Then a cost function Cost($g_i(\boldsymbol{x})$) is given. The goal is to find out the state $g_i(\boldsymbol{x})$ with minimum cost. Overwhelming majority of such problems are NP-complete, so the algorithms to solve such problems require exponential time relative to n. A near optimal state is often good enough in practice. One of the polynomial time heuristics for these problems is

the greedy algorithm. It does not always produce a satisfactory outcome but it is good algorithmic basis for simulated annealing.

Simulated annealing augments the greedy algorithm with a random escape from local minima converting it to a probabilistic hill-climbing algorithm. The escape is controlled through a value called as "temperature". Higher temperature makes the algorithm more likely to increase cost when selecting a trial state. In this way simulated annealing can climb out of a local minimum. Figure 3 [Greening, 1995] shows how a simulated annealing algorithm works.

$t \leftarrow t_s$;
$g \leftarrow$ starting state;
$Cost \leftarrow Cost(g)$;
while not stopping criteria( )
    $g^* \leftarrow$ generate(g) with probability G_{ss^*};
    $Cost^* \leftarrow Cost(g^*)$;
    $\triangle \leftarrow Cost^* - Cost$;
    if $(\triangle \leq 0) \vee (\text{random}() < e^{-\triangle/t})$
        $g \leftarrow g^*$;
        $Cost \leftarrow Cost^*$;
    $t \leftarrow$ reduce temperature(t);
end while

Figure 3. Simulated annealing procedure

The first three operators of the simulated annealing procedure (Figure 3) set the initial temperature t, the current state g and its cost $Cost$. The loop generates a trial state g^*, evaluates its cost $Cost^*$ and change of cost $\triangle$. Then if the condition is satisfied the algorithm selects this new state g^* as the next current state g and reduces the temperature until the stopping criteria is met. The condition "if $(\triangle \leq 0) \vee (\text{random}() < e^{-\triangle/t})$" shows how simulated annealing accepts a trial state. The term $(\triangle \leq 0)$ expresses greedy strategy (it chooses a lower cost trial state). The function random() returns a uniformly distributed random value between 0 and 1. The term $(\text{random}() < e^{-\triangle/t})$ evaluates the likelihood of permitting a costlier trial state, the probability of accepting a costlier trial state decreases exponentially with the increase in cost and the decrease in temperature t.

The function "reduce temperature (t)" decreases the temperature according to a cooling schedule, which provides fulfilling four major tasks in the proper way:

1) to set high enough starting temperature to accept most of the moves;

2) to determine when the present temperature is to be decreased;

3) to determine the next temperature;

4) to finish the process.

Together, the methods of generating moves and the cooling schedule form the foundation of simulated annealing procedure. A starting temperature t_0 must be assigned or calculated and a series of random alterations (iterations) is then made. In accordance with the accepted condition of moves the sequence of states at one temperature forms a Markov chain. When the chain reaches equilibrium at a particular temperature, the temperature is decreased according to some cooling schedule and the procedure is repeated. The algorithm terminates when a specified stopping condition is reached.

The number of iterations in an annealing run appears to be the most critical parameter in annealing, many methods have been suggested (their review cold be found in Lee, 1995]). The simplest of them is the geometric cooling schedule, where the starting temperature t_s, chain length L, temperature decrement, and stopping temperature T_e are all predefined.

Multiple folded regular structure realization

Column folding can be obtained by permuting the rows of an array structure, so it introduces a restriction on the order of the rows. That is, if column c_i is folded with column c_j and placed above it, then all the rows that have cross points with the column c_i must be placed above those rows that have cross points with the column c_j. For example, in Figure 1, to fold column x_6 above $\bar{x}_1$, thus rows r_2, r_3 and r_7 should be placed above r_1 and r_5.

The restrictions imposed on the order of rows by a list of folded columns can conflict with the row ordering desired by another list of folded columns. An unconflicting ordering of rows is decided as optimal if it needs to the maximal folding of columns or the minimal number of physical vertical lines. So we should find out such an unconflicting ordering of rows.

	$\bar{x}_1$ c_1	x_1 c_2	$\bar{x}_2$ c_3	x_2 c_4	$\bar{x}_3$ c_5	x_3 c_6	$\bar{x}_4$ c_7	x_4 c_8	$\bar{x}_5$ c_9	x_5 c_{10}	$\bar{x}_6$ c_{11}	x_6 c_{12}	$\bar{x}_7$ c_{13}	x_7 c_{14}
r_6	0	0	0	1	0	0	0	1	0	0	1	0	0	0
r_4	0	0	0	0	0	0	1	0	0	0	0	0	0	0
r_3	0	0	0	1	0	0	0	1	0	0	0	1	1	0
r_8	0	0	1	0	1	0	0	0	0	0	0	0	1	0
r_2	0	0	0	0	0	0	0	0	1	0	0	1	0	1
r_7	0	1	0	0	0	0	0	0	0	0	0	1	0	1
r_5	1	0	0	0	1	0	0	0	0	1	0	0	0	1
r_1	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 4. Boolean matrix $\boldsymbol{B}^{\pi}$ induced by the matrix $\boldsymbol{B}$ of Fig.2

	$\bar{x}_1$ c_1	x_1 c_2	$\bar{x}_2$ c_3	x_2 c_4	$\bar{x}_3$ c_5	x_3 c_6	$\bar{x}_4$ c_7	x_4 c_8	$\bar{x}_5$ c_9	x_5 c_{10}	$\bar{x}_6$ c_{11}	x_6 c_{12}	$\bar{x}_7$ c_{13}	x_7 c_{14}
r_6	0	0	0	**1**	0	0	0	**1**	0	0	**1**	0	0	0
r_4	0	0	0	1	0	0	**1**	1	0	0	0	0	0	0
r_3	0	0	0	**1**	0	0	0	**1**	0	0	0	**1**	**1**	0
r_8	0	0	**1**	0	**1**	0	0	0	0	0	0	1	**1**	0
r_2	0	0	0	0	1	0	0	0	**1**	0	0	**1**	0	**1**
r_7	0	**1**	0	0	1	0	0	0	0	0	0	**1**	0	**1**
r_5	**1**	0	0	0	**1**	0	0	0	0	**1**	0	0	0	**1**
r_1	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0

Figure 5. Interval matrix $\boldsymbol{L}^{\pi}$ corresponding to $\boldsymbol{B}^{\pi}$ in Fig. 4

As column folding is formulated as a row permutation problem, during the column folding phase the rows of an array structure are permuted to ensure some columns could be share the same vertical lines of the folded structure using less lines comparing the initial number of columns. To derive the conditions row permutation should satisfy to lead to a maximal size column folding, let consider the result of multiple folding of regular

structure (Figure 1). It corresponds to some row permutation π that defines the row sequence and is one-to-one function of assigning rows of initial matrix $\boldsymbol{B}$ (Figure 2) to horizontal lines: $\{\pi(r_1), \pi(r_2)..., \pi(r_m)\} \rightarrow \{r_6, r_4, r_3, r_8, r_2, r_7, r_5, r_1\}$. So the Boolean matrix $\boldsymbol{B}$ will be transformed into the Boolean matrix $\boldsymbol{B}^{\pi}$ (Figure 4) with permuted rows that is functionally equivalent to $\boldsymbol{B}$ but correspond the other structure realization. Then, the columns of $\boldsymbol{B}^{\pi}$ will be rearranged in top-to-bottom manner and a column partition is resulted describing a folded structure in symbolic form. The matrix $\boldsymbol{B}^{\pi}$ defines a graph $G = (V, E)$, where V is the set of columns and two vertices v_i and v_j are connected by an edge $e_{ij} \in E$ if the appropriate columns c_i^{π} and c_j^{π} intersect (have a 1 in the same row): $c_i \cap c_j \neq \varnothing$.

An ordering of rows induced by permutation π is termed as optimal if it induces the maximal folding of columns or, as the same, the minimum number of physical vertical lines of the folded structure. It follows from the folded structure that there exists column partition $C^{\pi} = \{C_1, C_2,..., C_k\}$ concerning to the row permutation π, such that all the columns of each component set C_k are mutually compatible and so form the folding list, and C^{π} as a folding set is realizable. For our example, for the Boolean matrix $\boldsymbol{B}^{\pi}$ we could obtain $C^{\pi} = \{\{c_{11}, c_7, c_{12}, c_1\}, \{c_4, c_5, c_6\}, \{c_{13}, c_{14}\}, \{c_8, c_3, c_9, c_2, c_{10}\}\}$. The cardinality k of the partition C^{π} defines the width of the row-permuted folded structure or the number of packed columns in it. The goal is to find out a row permutation that permits a column partition C^{π} of minimal cardinality.

Now let us introduce some notions to formulate the procedure of evaluation of a row permutation π. For a given permutation π the regular structure folding is realized by attaching to each vertical line column parts sharing it. Such parts correspond to intervals [introduced by Wong, 1988] that are established in the columns of the matrix $\boldsymbol{B}^{\pi}$ from the topmost to the lowermost 1's: $\boldsymbol{I}_k = [\min_i c_i^k, \max_i c_i^k,]$, where c_i^k is i-th component of k-th column having value 1. Such a way the matrix $\boldsymbol{B}^{\pi}$ induces interval matrix $\boldsymbol{L}^{\pi}$ for a given row permutation π. Just the intervals will be assigned in the proper way to vertical lines of the folded array structure also need the minimum number of vertical lines for the folded structure. When proceeding from $\boldsymbol{B}^{\pi}$ to $\boldsymbol{L}^{\pi}$ the densities d^{π}_i of the rows (the number of columns intersecting the row d^{π}_i) are increased in the general case. Reducing the row densities promotes placing more columns on a single physical line, and that can be done by proper permutation of rows.

For our example, we have the interval matrix $\boldsymbol{L}^{\pi}$ shown in Figure 5 (where the initially given 1's are represented in thick print in contrast to augmented 1's that are of regular type) and following the collection of intervals:

$$L^{\pi} = \{[7,8], [6,6], \{[4,4], \{[1,3], \{[4,7], \{[8,8], \{[2,2], \{[1,3], \{[5,5], \{[7,7], \{[1,1], \{[3,6], \{[3,4], \{[5,7]\}.$$

To find out the size of folding associated with the permutation π let consider the compatibility relation between intervals: a pair of intervals $\boldsymbol{I}_i$, $\boldsymbol{I}_j$ are compatible if they do not intersect (the appropriate columns of the matrix $\boldsymbol{L}_{\pi}$ do not have a 1 in the same row): $\boldsymbol{I}_i \cap \boldsymbol{I}_j = \varnothing$. Compatible intervals may be assigned the same vertical line in a proper layout. The inverse relation is the incompatibility relation that can be represented by the intersection graph

$G = (V, E)$, where the vertices of V correspond to intervals and $(v_i, v_j) \in E$ if and only if $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$ intersect, ($\boldsymbol{l}_i \cap \boldsymbol{l}_j \neq \varnothing$). The graph is an interval graph. Interval graphs are a kind of the chordal graphs [Hsu, 1999] having some remarkable properties reducing NP-complete problem of searching of realizable folding sets to a solvable in polynomial-time.

It have been stated that the size of folding (associated with row permutation π) equals to the chromatic number of an interval graph $G = (V, E)$ and the color classes correspond to collections of intervals each of which can be feasibly assigned to a line. Then the chromatic number of the interval graph is equal to the maximal degree of vertices of the graph G. In other words the size of folding associated with a row permutation π can be found directly from the corresponding interval matrix $\boldsymbol{L}^{\pi}$ – it is the maximal density of the matrix rows (denoted later as d^{π}) [Wong, 1988]. And further a minimum size folding induced a row permutation π can be found solving 1) the task of graph coloring: the color classes of an interval graph may be obtained by applying a simple linear-time algorithm to the vertices of an interval graph $G = (V, E)$ [Golumbic, 1977] , or 2) the task of interval ordering by applying to the collection L^{π} of intervals a special simple "Left Edge Algorithm" [Hashimoto, 71].

The interval graph $G = (V, E)$ for our example is depicted in Figure 6. The graph chromatic number equals 4. The appropriate ordered column folding set is $L^{co} = \{<c_{11}, c_7, c_{12}, c_1>, <c_4, c_5, c_6>, <c_{14}, c_{13}>, <c_8, c_3, x_9, c_2, c_{10}>$, it is implementable and the corresponding folded regular structure is given as a part of Figure 1, b.

Thus from above, it can draw a conclusion that any row permutation is evaluated in a simple way: the value of d^{π} is the numerical value of the row permutation quality. And after choosing the best row permutation, the partition of the columns set on a collection of column folding lists is fulfilled by linear-time algorithm.

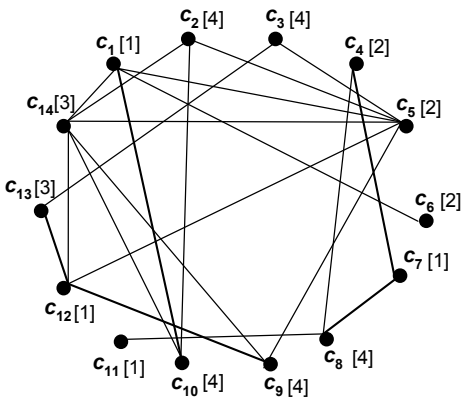


Figure 6. Interval graph $G = (V, E)$

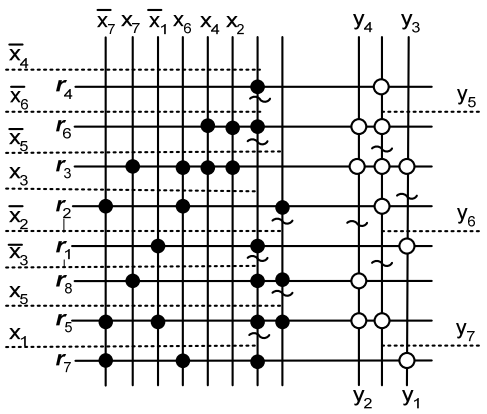


Figure 8. PLA multiple folded form subject to signal routing lines

Multiple folding via the method of simulated annealing

Here we consider how to solve the problem of multiple column folding using the method of simulated annealing. During the column-folding phase, the rows of an array structure are permuted to minimize the area due to folding some columns. In that case array structure will have fixed height and decreased width. The aim of the column-folding phase is to minimize the width of the structure. Indirectly, to minimize the width, it would be necessary to minimize the number of the columns. The goal is to obtain an ordering of rows that will allow maximal packing the columns.

Figure 7 shows how the realized simulated annealing folding algorithm works. The current row order π in the matrices $\boldsymbol{B}^{\pi}$ and $\boldsymbol{L}^{\pi}$ is a state in the process of simulated annealing. And the starting state is the initial row order $\{r_1, r_2,\ldots, r_m\}$. A move is a permutation of a pair of columns and it changes the current state for some neighbor one. The temperature schedule is represented by geometric series where starting value $t_0 = \lambda_t n$ and each successive temperature value is obtained by multiplying the previous one by some constant value that is 0.85 that is less than 1 and so the temperature values approach zero in the limit. Here n is the number of columns, the value $\lambda_t < 1$ is taken as 0.8 (for it experimental results indicate good results obtained) [Wong, 1988]. The number of iterations (moves) at each temperature is accepted as $r_0 = 2\lambda_r C_m^2$, $C_m^2 = m(m–1)/2$ is the number of 2-combinations (the number of possible neighbor states) – the number of all possible moves where the value $\lambda_r < 1$ is taken as 0.5 [Van Laarhoven, 1987], m is the number of rows.

Each trial state is evaluated with a cost function Q^{π}. The goal of an annealing folding algorithm is to find out the interval matrix $\boldsymbol{L}^{\pi}$ with minimum of maximal density: $\max \text{of } d^{\pi} \rightarrow \min$. Taking into account that the row permutation and the corresponding to it maximal density are random quantities, the cost function Q^{π} is expressed as quadratic one:

$$Q^{\pi} = (d^{\pi})^2 + \frac{\lambda_{\pi}}{m}\sum_i (d_i^{\pi})^2 , \qquad (2)$$

where λ is accepted to be 0.5 reducing the influence of extra item $\frac{1}{m}\sum_i (d_i^{\pi})^2$ by a half. The trial is successful if the cost value of the current state is less than that of the best previous state. The simulated annealing process is terminated if the number of successful steps at a current temperature is less than 5% of all moves or the temperature became low enough.

$\boldsymbol{B} \leftarrow \boldsymbol{B}_0$ folding	$\boldsymbol{B}_0$ – initially given Boolean matrix presenting regular structure under
$\pi \leftarrow \pi_0$	π_0 – initial state, i.e. initial order of rows: $\pi_0 = \{c_1, c_2, \ldots, c_m\}$
$\boldsymbol{L}^{\pi} \leftarrow \boldsymbol{L}^{\pi 0}$	$\boldsymbol{L}^{\pi}{}_0$ – interval matrix $\boldsymbol{L}^{\pi}$ corresponding to the row permutation π_0
$t \leftarrow t_0$	t_0 – starting temperature: $t_0 = 0.8\,n$
$r_t \leftarrow r_0$ temperature $r_0 = m(m–1)/4$	r_0 – the number of iterations (moves) at starting

```
Q^π ← Q^π_0                    Q^π_0 – the cost function that is the row permutation quality equaled to
d^π –
                    (maximal number of 1's in a row of the interval matrix L^π)
while ((δ_t ≥ 0.05 r_t) ∨ (t ≠ 0)) do                    not stopping criteria( )
   begin
   i ← 0                       i is the serial number of iteration at a current temperature t
   δ_t ← 0                     δ_t  is the number of successful steps at a current temperature
   t
      while (i < r_t) do                                 for t = t_const
         begin
         π*← generate random neighbor of π
         B* ← B^π*                                       building Boolean matrix B for π*
         L^π* ← L^π(π*)                                  building interval matrix for π*
         Q^π* ← Q^π(π*)                                  evaluating L^π*
         △ ← Q^π* – Q^π
         if ((△ ≤ 0) ∨ (random(0,1) < e^(–△/t))) then
            π ← π*; B ← B^π*; Q^π ← Q^π*;
            δ_t ← increase the number of successful steps(δ_t)
      end while
   t ← reduce temperature(t);                            new temperature will be 0.85 of current value
end while                                                the best solution is obtained
```

Figure 7. Simulated annealing folding procedure

Constrained multiple folding

The discussed procedure of multiple folding will be modified when some constraints on the form of folded structure are imposed. Such folding constraints can be divided on the following basic groups.

1. Structure-defined constraints. This class of constraints is defined by an array structure type in use. For example, use of PLA forces to take into consideration that its layout consists of AND-plane and OR-plane.

2. Input/Output constraints. This class of constraints depends on the environment in which the folded structure should be placed. For example, these can be the ordering of input/output directions, relative positions of input/output lines, grouping of input/output lines.

3. Electrical constraints. This class of constraints forces to limit the folding type. For example, it could be only simple or bipartite folding.

4. Physical Constraints. This class of constraints concerns to input lines carrying a signal and its inversion. For example, the appropriate array structure columns (or rows) may not share the line, so they cannot be folded, and sometimes they should be placed to each other as close as possible.

Further we show how we take proper account of some constraints imposed on folding demonstrating on the example of simple folding and PLA folding.

PLA folding via the method of simulated annealing

A PLA realizes a collection of Boolean functions in disjunctive normal form. It is a two-dimensional array consisting of an AND-plane and OR-plane. Its vertical lines are assigned with the input variables (and their complements) and output variables. Inputs run vertically through the AND-plane that generates signals on its rows, which are used as inputs to the OR-plane. The PLA folding allows some its columns to share a vertical line (or rows to share a horizontal line). An example of the PLA structure and its free folded form is shown in Figure 1. Here, a dot means placing a transistor on a cross point of vertical and horizontal lines.

PLA layout has structure-defined constraint that disallows a column of the AND-plane to share the same vertical line with a column of the OR-plane. Then, any physical realization of a folded PLA must ensure inputs/outputs be connected to the PLA outside. In [Wong, 1988] two possible architectures of implementation of routing lines carrying input/output signals to outside (with the help of horizontal auxiliary connection lines) and the modification of the simulated annealing process are shown.

So, PLA multiple folding is the constrained multiple folding of an array structure. It comes easily to take into account the first constraint when formulating the simulated annealing procedure. In this case the interval matrix $\boldsymbol{L}^{\pi}$ consist of two column submatrixes corresponding to the AND- and OR-planes, and we should distinguish between input $d^{\pi in}{}_i$ and output $d^{\pi out}{}_i$ densities of the matrix $\boldsymbol{L}^{\pi}$ rows. Then we construct two interval graphs $G^{in} = (V^{in}, E^{in})$ and $G^{out} = (V^{out}, E^{out})$ for the AND- and OR-planes. And the folding size associated with a row permutation π equals the sum of chromatic numbers of these graphs or it equals the maximal density d^{π} that is calculated as the sum of maximal input $d^{\pi in}$ and output $d^{\pi out}$ densities. Thus the simulated annealing procedure for solving the task of such constrained multiple folding distinguishes from the above formulated one only with the calculating maximal density d^{π} and accordingly the cost function Q^{π}.

In exactly the same way we can take into account the second constraint (concerning signal input and output), it is made by means of augmenting intervals in a proper way the structure have place to feed signals to its inputs and from its outputs. Thus we take a proper account of increasing interval lengths on the step of transforming matrix $\boldsymbol{B}^{\pi}$ into the interval matrix $\boldsymbol{L}^{\pi}$. The folded form of PLA from Figure 1,a that takes into account the routing lines is depicted in Figure 8.

Simple folding via the method of simulated annealing

Now we consider how to take into account more complex constraints – electrical constraints resulted to limit the folding type, for example, it could be only simple folding. Remember that folding is called simple if utmost two columns (rows) are allowed to share a single physical vertical (and/or horizontal) line.

Simple column folding has an evident advantage over multiple columns folding because external signals could connect to the folded structure either from the top, or the bottom of a folded structure because there is at most one break in any column. This simplifies routing signals, in addition, despite multiple folding might result in larger area reduction, simple folding allows reduce occupied routing area.

To take into account the peculiarity of simple folding in [Wong, 1988] it is proposed to use the same annealing schedule as for multiple folding but with other cost function (2). The function value depends on the matching number of undirected graph of pairwise compatibility (foldability) of PLA structure columns. Thus, at each simulated annealing iteration, we have to make the laborious procedure of compatibility graph construction and its matching number computation.

We propose to reduce the laboriousness of simple folding via simulated annealing method thanks to single implementation of the mentioned procedure of compatibility graph construction and its matching number computation. That cannot allow to obtain maximum folding in some cases but taking into account that simulated annealing method is a heuristic method that does not guarantee the optimum, the proposed idea is good enough.

The main point is based on the heuristics that we can get a good simple folding on basis of a good multiple folding. Thus, first we find out the best row permutation of array structure under folding using the proposed above procedure of multiple folding via the method of simulated annealing. Then we don't find array structure multiple folding itself but we will search for its best simple folding.

So, let we have now row permutation π and the corresponding interval matrix $\boldsymbol{L}^{\pi}$ (for the considered example it is shown in Figure 5). Let remember two intervals $\boldsymbol{l}_i$, $\boldsymbol{l}_j$ are compatible if they do not intersect, a pair c_i, c_j of corresponding columns are foldable and may be assigned the same vertical line in a proper layout. Let $H = (V, E)$ be an undirected graph, where the vertices of V correspond to intervals and $(v_i, v_j) \in E$ if and only if $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$ don't intersect, ($\boldsymbol{l}_i \cap \boldsymbol{l}_j = \varnothing$). The size of simple folding associated with row permutation π equals to the matching number of the graph $H = (V, E)$ and the matching corresponds to collections of folding pairs.

A matching M in a graph H is a set of pairwise non-adjacent edges; that is, no two edges share a common vertex. Each edge in M defines a folding pair, and a set M specifies a folding set. Correspondingly, a maximum matching (that contains the largest possible number of edges) gives a folding set of maximum size. The number of columns of folded such a manner structure is equal the matching number plus the number of unmatched vertices of the graph H.

For our example the graph $H = (V, E)$ (corresponding to interval matrix $\boldsymbol{L}^{\pi}$ in Figure 5) is shown in Figure 9. Here all graph edges, but matching, are shown as thin lines and edges of the matching edges are shown as heavy lines. The folding variant of PLA shown in Figure 1,a is given in Figure 10.

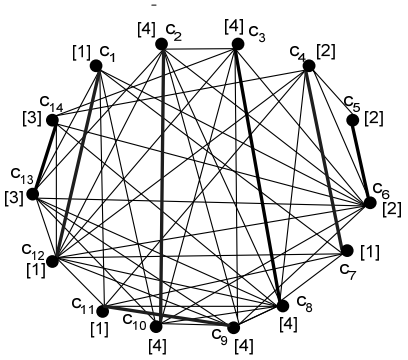


Figure 9. Compatibility graph $H = (V, E)$

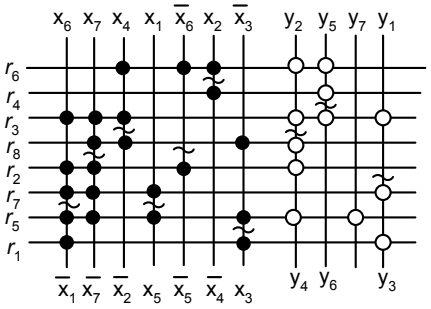


Figure 10. PLA simple folded form

Experimental results

The simulated annealing algorithms for multiple and simple folding (have been described above) were formulated for one of the types of regular structures, PLA structure, and were realized using Visual C++. Then they were compared on the stream of benchmarks [Berkeley, 2006].The objectives of performed computer experiments are to state:

1) the degree of PLA area reduction that could be achieved by both simple and multiple folding;

2) the dependence of degree of PLA area reduction on PLA AND- and OR-planes sparseness;

3) the comparative evaluations of two types PLA folding: simple and multiple ones.

As the basic parameter of PLA structures governed the degree of PLA compaction the density of PLA AND- and OR-planes was accepted. The density of PLA planes is the percent of active transistors in them (in relation to number of all transistors), or the PLA density is the ratio between the number of unit elements of Boolean matrix $\boldsymbol{B}$ (Figure 2) to whole number of its components.

The results of the testing are given in Figure 11; at the bottom of the Figure, in X-direction, bench PLA names are placed accompanied their densities. In Y-direction there are values of PLA area compaction expressed in percentage, that is $\frac{k}{n}\%$, where n is the number of PLA columns under folding and k is the number of the folded PLA columns.

Experimental results allow to draw a conclusion concerning the domains of applicability of simple and multiple folding. The results indicate that:

1) multiple folding gives better results for sparse PLA cases, or numerically in cases when PLA density is less than 20%;

2) simple folding gives better results for dense PLA cases, or numerically in cases when PLA density is more than 25%;

3) in the range of PLA densities 20 – 25% multiple and simple PLA foldings are competed with each other.

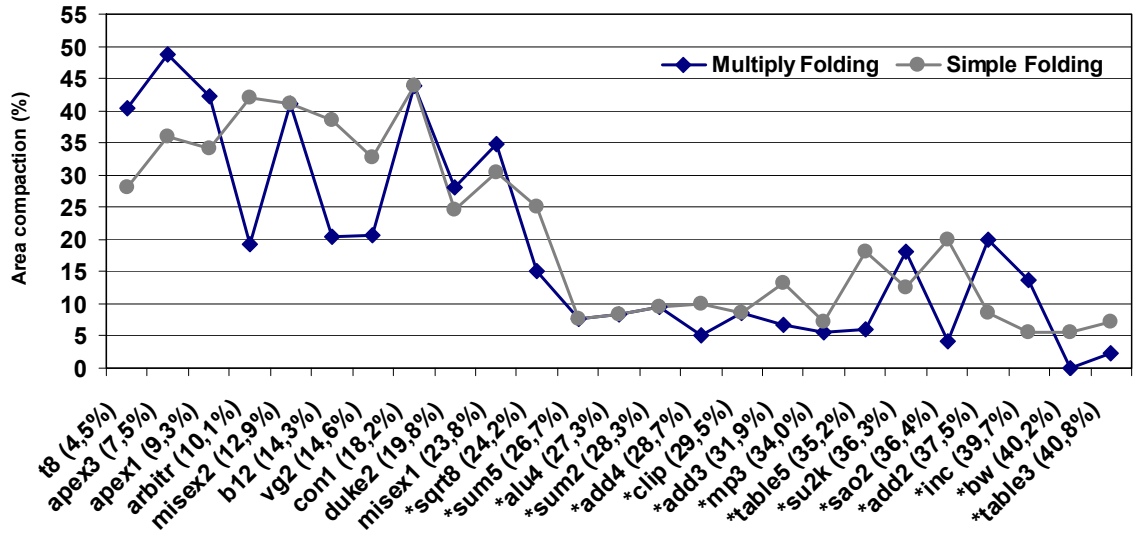


Figure 11. The results of experimental investigations of multiple and simple folding using simulated annealing algorithm

Conclusion

The problem of reducing the area of the layout of two dimensional array structures is investigated. We consider probabilistic heuristic algorithms based on simulated annealing procedure, they are well suited for compacting sparse structures having not great percentage of active elements (transistors).

Simulated annealing folding algorithms are investigated for the case of PLA multiple and simple column folding. The results of investigation show that 1) simulated annealing can give enough good results in the sense of area reduction, and 2) simple folding behaves not worse comparing with multiple folding on the stream of PLAs have been considered allowing for finding often solutions with the same cost function.

Bibliography

[Ullman, 1984] D. Jeffrey Ullman Computational aspects of VLSI. Rockville, Md.: Computer Science Press, 1984, 495 p.

[Hachtel, 1982] G.D. Hachtel, A.R. Newton and A.L. Sangiovanni-Vincentelli. An Algorithm for optimal PLA Folding. In: IEEE Trans. Computer-Aided Design of Integrated Circuit Syst., 1982, vol. CAD–1, No 2, pp. 63–77.

[DeMicheli, 1983] G. DeMicheli and A. Sangiovanni-Vincentelli. A. Multiple Constrained Folding of Programmable Logic Arrays: Theory and Applications. In: IEEE Trans. Computer-Aided Design, 1983, Vol. CAD-2, No 3, pp. 151–167.

[Wong, 1988] D.F. Wong, H.W. Leong, C.L. Liu. Simulated Annealing for VLSI Design. Kluwer Academic Publ., Boston, 1988, 220 p.

[Lee, 1995] Fu-H. A. Lee. Parallel simulated annealing on a message-passing multi-computer, Doctor of philosophy thesis, UTAH STATE UNIVERSITY, Logan, Utah, 1995.

[Kirkpatrick, 1983] S. Kirkpatrick, Jr. C. D. Gelatt and M. P. Vecchi. Optimization by simulated annealing. In: Science, 1983, 220 (4598), pp. 671–680.

[Van Laarhoven, 1987] P.J.M. Van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel, Dordrecht, 1987.

[Hsu, 1999] W.-L. Hsu and T.-H. Ma. Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs. *SIAM J. Comput.*, 1999, Vol. 28, No 3, pp.1004–1020.

[Golumbic, 1977] M.C. Golumbic. The complexity of comparability graph recognition and coloring. Computing, 1977, Vol. 18, pp. 199-208.

[Greening, 1995] D. R. Greening. Simulated Annealing with Errors, Doctor of philosophy thesis, UNIVERSITY OF CA, Los Angeles, 1995.

[Hashimoto, 71] A. Hashimoto, J. Stevens. Wire Routing by Optimizing Channel Assignment Within Larger Apertures. In: Proc. of 8th Design Automation Workshop, 1971.

[Berkeley, 2006] Berkeley PLA test set [Electronic resource]. – Mode of access: http:// www1.cs.columbia.edu/ ~cs4861/sis/ espresso-examples/. – Date of access: 03.05.2006.

The research was partially supported by the Fond of Fundamental Researches of Belarus (Project Ф10P–035).

Authors' Information

Liudmila Cheremisinova *– Principal Researcher, The United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganov str., 6, Minsk, 220012, Belarus, e-mail:* cld@newman.bas-net.by

Major Fields of Scientific Research: Logic Design, CAD systems, optimization

Irina Loginova *– Senior Researcher, The United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganov str., 6, Minsk, 220012, Belarus, e-mail:* cld@newman.bas-net.by

Major Fields of Scientific Research: Topological Design, CAD systems

CONSTRUCTION OF A REALISTIC MOVEMENT ON THE 3D HUMAN MODEL FOR STUDYING AND LEARNING SIGN LANGUAGE

Iurii Kryvonos, Iurii Krak, Anton Ternov

***Abstract**: Complex information technology for communications gesture and mimicry modeling and visualization is created.*

***Keywords**: modeling, sign language, computer system, visual-speech synthesis and analysis.*

***ACM Classification Keywords**: I.2.8 Problem Solving, And Search H.1.1 Systems and Information*

Introduction and problem statement

Among three ways that people communicate (in text, voice, gesture), the important and under-researched is the sign language, which is the basis of communication for deaf people and people with hearing disabilities [Belikov]. Because the information in sign language is transmitted through hands movements, facial expressions and articulations, the main visual tools for learning such a language are a photographic images and video of gestures. On the one hand the use of text books with a verbal description of signs leads to a different interpretation of movements and gestures by each person through his own interpretation of information, which further complicates the understanding and expression the meaning of sign communication. On the other hand, exclusive usage of photos and video of a gesture for the development of modern educational and communicational computer systems is quite difficult and not always possible, since a photographic image does not show the desired dynamics of a gesture, video does not allow to see a gesture from the other side. In addition, simple concatenation of videos does not result in a smooth signing.

Mentioned difficulties in usage in classical approaches to the learning of gestures lead to the need to develop more efficient technologies that would allow to create new computer systems for modeling and learning a sign language. One of the basic approaches is to develop a comprehensive information technology which includes the functionality of the synthesis of movements of a sign language on the spatial model [Krivonos].

It is required to implement an animation capabilities of the communication process with the help of sign language using a virtual human model, to establish appropriate information and mathematical models. These researches aim to build a model for fixing morphemes (the minimal meaningful units) of a sign language; to create technology and software to receive, save and play gestures within the framework of the model; to suggest algorithmic

solutions a) to calculate the typical human hand trajectories, b) to model body movements in transitions from one gesture to another, and c) to animate facial expressions and articulations of a human face.

The spatial model of the person to fix the units of sign language

Animation of a gesture can be considered as an animation with appropriate frequency of different states of simplified skeletal model, which recreates a skeleton of a real person. It can be formalized as a hierarchical structure consisting of units connected by a kinematics pairs of the corresponding class [Kirichenko]. For example, modern three-dimensional modeling packages (Poser, 3D Studio Max) allow to generate an animation using information about changes in the respective angles of bones rotation.

A gesture can be formally described by a set H which is reflected a simplified human skeleton, Euler angles M and an order of their application. It can be used to build the corresponding bones of the skeleton with a change in time (discrete, with appropriate frequency, e.g.,1 / 30 sec):

$$H = \{H_i : H_i = \{k, d_i, M_i \in M\}\}, \quad (1)$$

where H_i – i-th bone in a skeleton ($i = \overline{0, N-1}$, N – number of bones in a skeleton); k – index of a parent bone; $d_i = [x_i, y_i, z_i]^T$ – coordinates of the point of the end of the bones in the coordinate system associated with the beginning of this bone;

$$M = \{M_i : M_i = (order_i, \theta_i)\}, \quad (2)$$

where M_i – values of Euler's angles and the order of rotations over time for the i-th bone; $order_i \in \{1, \cdots 6\}$ – an order of rotations implementation around the respective axes for the i-th bone (1-XYZ, 2-XZY, 3-YXZ, 4-YZX, 5-ZXY, 6-ZYX); $\theta_i = (\theta_i^j)$, $\theta_i^j \in \{(\varphi_{i\,X}^j, \varphi_{i\,Y}^j, \varphi_{i\,Z}^j)\}$ – set of Euler's angles for the i-th bone ($j = \overline{0, K-1}$, K – number of frames to play the gesture at a given frequency).

To obtain a set of angles (2) that characterize the change in the position of bones from the initial state of the skeleton motion capture technology is used [Lander]. Based on this approach corresponding software has been created. With the help of it capture, digitization and fixation of gestures can be done.

Modeling of facial expression and articulation of a human face during pronunciation

To display the facial expression and articulation of a person on a spatial model it's necessary to construct a mathematical model in which it is possible to identify differences in the position of human lips contour for the construction of a system to the proper articulation. The mathematical model must include the implementation of

the possibilities of creating a visual alphabet for a language, visems analysis and applying these results to arbitrary models of the human head.

For the synthesis of the mathematical model let's move from the space of photographic images of the human face in the process of pronunciation to the vector space of characteristic parameters in order to display a human face on the spatial model. Such a transition is implemented in two phases:

1. Detection the inner contour of lips on the image

$$\mathrm{Im}\, L \to D, \tag{3}$$

where $\mathrm{Im}\, L = \{I_k : I_k \in FSV\}$ – ordered set of key frames of the video stream FSV (*Face Speech Video*), formed when shooting facial expressions on the human face, namely the position of lips in pronunciation of the words ($k = \overline{1,K}$ – ordinal index of the frame in the chosen sequence, where K – number of key frames); $I_k = \{col_{ij}^k\}$, $i = \overline{1,m}$, $j = \overline{1,n}$ – image of size $m \times n$ of a face with a mimics state of lips during articulation, m and n – length and width of image I_k respectively; $col_{ij}^k = I_k(i,j)$ – color of a pixel in system RGB with coordinates (i,j) on a image I_k; $D = \{D_k : D_k = \{d_{top}^k, d_{bot}^k\}\}$ – a set of lips contour, D_k – pair of point curves – lips contours (top d_{top}^k and bottom d_{bot}^k) for k-th frame.

2. Approximation with irregular basis splines (NURBS) [Piegl] of the inner contour of lips – obtaining the vector of characteristic features:

$$D \to P, \tag{4}$$

where: $P = \{p_k : p_k^i \in F, i = \overline{1,s}\}$ – space of characteristic features, , F – the characteristic features of the researched object, p_k – characteristic vector, p_k^i – coordinates of this vector, s – dimension P.

Using the transformation (3), (4) an information technology [Krak] has been created which helps in modeling facial expressions and articulations of the human face.

Modeling of gestures animation and facial expressions

For the synthesis of animation of gestures and facial expressions on human spatial model the following formal description of the corresponding set of parameters and algorithms are proposed. Three-dimensional human model, which will be implemented by the process of animation gestures and facial expressions, has the following components: $V = \{v_i : v_i = (x,y,z)\}$ – set of vertices of triangles for the triangulation surface of three-dimensional human model; $N = \{n_i : n_i = (x,y,z)\}$ – set of normales to the vertices; $T = \{t_i : t_i = (t,r)\}$ – set of texture coordinates to the vertices; $V^{ind} = \{V_i^{ind} : V_i^{ind} = (k_1,k_2,k_3)\}$ – set of indices indicating the order

of construction of the triangles from the set of vertices; $I = \{I_i : I_i = \{img\}\}$ – set of photographic images of elements of the model – textures; $i = \overline{1,S}$, S – a certain number.

To modulate a skeleton animation it is necessary to be able to calculate new values of the vertices V of triangles. For this purpose, a mechanism of skinning is used. It can be defined as an algorithm for binding set of vertices V to bones' rotation angles. Then the model of skeletal animation can be formalized as follows:

$$MH = \{MH_i : MH_i = \{k, \{l_1, \ldots, l_m\}, d_i, Glb_i, Order_i\}\}, \quad (5)$$

where MH – description of simplified skeleton of a person (hierarchy of bones) for realization of a skeleton animation. Here MH_i – i-th bone of a skeleton ($i = \overline{0, N-1}$, N – number of bones in a skeleton); k – index of parent bone; $\{l_1, \ldots, l_m\}$ - set of indexes of child bone, $d_i = [x_i, y_i, z_i]^T$ – coordinates of the end of i-th bone in the coordinate system associated with the beginning of this bone; Glb_i – vector to determine the coordinates of the bones in the global coordinate system; $Order_i$ – order of rotation; $Skin = \{Skin_i : Skin_i = \{(IndexVertex_1, Weight_1), \ldots\}\}$ – set of vertices that affect the current vertex when the angles are changed.

Skinning is calculated for each vertex v as follows:

$$v' = \sum_{i=0}^{n} \{(v * IM_{H_i} * TM_{H_i}) * w_{H_i}\}, \quad (6)$$

where n – number of bones belonging to the vertex v; IM_{H_i} – inversion matrix for bone H_i; TM_{H_i} – transfer matrix relatively to the bone H_i; w_{H_i} – weight coefficient of influence of point of bone H_i on vertex v.

Pronunciation of animation and emotional coloring are modeled using morphing technique (a smooth transition from one state of object to another). Only base states are used during morphing in which the intermediate states are calculated and animation are modeled. The method of segment morphing makes it possible to form several different facial expressions based on a small number of morphs and change the state of a face on a pronunciation animation. The advantage of this morphing is an opportunity to animate jaw independently from the lips and eyes, as well as from the emotional expressions on the face. The expression for the morphing from the R morphs in the formalism of the model for fixed frame is

$$V' = V + \sum_{m=1}^{R} w_m \cdot RM_m, \quad (7)$$

where w_m – weights coefficients, V – set of vertices of the base model, RM_m – input morph for blending. The result of the operation - a linear combination of the vertex set of a model and facial expressions.

Morphing using basic visemes is made taking into account weights coefficients in next steps:

1) The number of frames of gesture animation is calculated;
2) The set of emotion that are present in the word is defined;
3) The set of visemes to visualize the process of articulation is calculated;
4) Durations of animations stages, points of visemes emergence and weights for each base morph in frame are calculated.

To synchronize the animation of gesture and mimics it is necessary that the beginning and the end of animation of articulation coincide with the beginning and the end of the animation of a gesture.

Example of the synthesis of the human face conditions by blending morphs of visemes and emotions is shown on Pic. 1.

a) b) c)

Pic. 1. Different states of human face: a) - viseme "y", b) - interrogative expression, c) the blending result during gesture animation

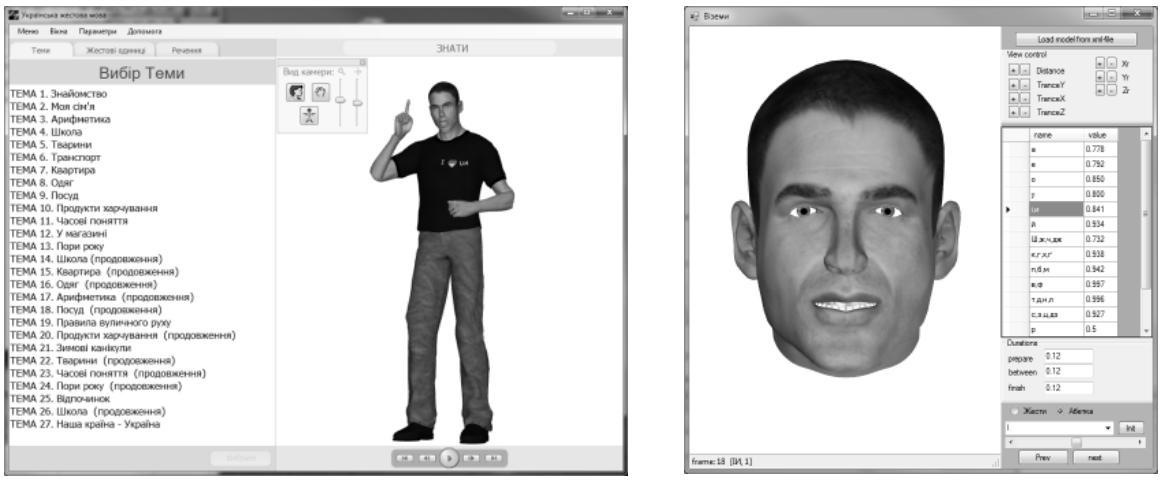
Information technology to play the animation process

To render the animation process of gestures and facial expressions of human three-dimensional model, software that implements the skeletal animation to peform a gesture (Pic. 2a) and morpheme animation to play facial emotions and pronunciation (Pic. 2b) is created.

The corresponding software functionality is implemented, which, using three-dimensional API OpenGL, plays 3D human model by the specified attributes. The model of gesture animation using skinning algorithms (Block of skeletal animation) and morphing (Block of mimics-articulated animation) is shown on Pic.3.

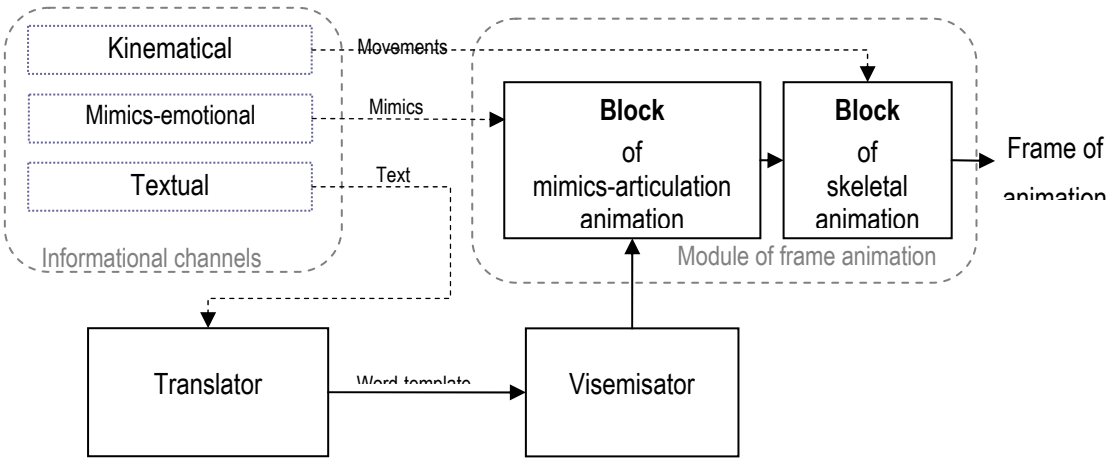
Based on the established information technology Ukrainian Sign Language is implemented. Methodology of teaching sign language in specialized schools for deaf children is implemented. Formally, the technology consists of three informational blocks (topics, words and sentences) and a block to play gesture on virtual model, which

has a specific purpose - with the help of it an opportunity to demonstrate in the learning process dynamics of the gesture is emerged. Since gestures are digitized by sign language native speaker, they, in fact, become the standards of playback of a gesture. This functionality for the frame demonstration of a gesture is a means by which it will be possible to study a gesture without a particular teacher.



a) b)

Pic. 2. a) Spatial model of a person, b) morphemic model of the human face



Pic. 3. Model of gesture animation process

Conclusions

Using the proposed model for the fixation of movements, couple dozens of gestures of Ukrainian Sign Language have been obtained. Thanks to developed technology of playing gestures from this set (using the spatial model) one can model the motion obtained from the video image of a specific person - sign language native speaker.

Developed technology and software allow to unify the study of sign language and in this sense can become the basis for creating a standard set of correct signing for a sign language.

Further studies are aimed at improving the proposed technology:

- filling the database of signs;
- development of tools for semantic binding of sentences on ordinary language with sentences on sign language (text-to-sign translation);
- improvement of the spatial human model taking into account some specifics of implementation of a sign language

Bibliography

[Belikov] Беликов В.И. Жестовые системы коммуникация // Семиотика и информатика. Вып. 20. – М.:Языки русской культуры, Русские словари,1973. – С.127-148.

[Krivonos] Кривонос Ю.Г., Крак Ю.В., Бармак О.В. [та ін.] Інформаційна технологія невербального спілкування людей з вадами слуху // Штучний інтелект. – 2008. №3. – С. 325-331.

[Kirichenko] Кириченко М.Ф., Крак Ю.В., Сорока Р.О. Оптимізація маніпуляційних роботів. – К.: Либідь, 1990. – 144 с.

[Lander] Lander J. Working with Motion Capture File Formats 3D. – LLC, Jan., 1998. – P. 30–37.

[Piegl] Piegl L., Tiller W. The NURBS Book, 2nd Edition. – Berlin, Germany: Springer-Verlag, 1996. – 645 p.

[Krak] Крак Ю.В., Бармак О.В., Тернов А.С. Информационная технология для автоматического чтения по губам украинской речи // Компъютерная математика. – 2009. – №1. – С. 86-95.

Authors' Information

Iurii Kryvonos *- V.M.Glushkov Cybernetics Institute of NASU, Director Deputy,*

Iurii Krak *V.M.Glushkov Cybernetics Institute of NASU, senior scientist*
krak@unicyb.kiev.ua,

Anton Ternov *– V.M.Glushkov Cybernetics Institute of NASU, junior researcher*
anton.ternov@gmail.com

address: 40 Glushkov ave., Kiev, Ukraine, 03680

EFFECTIVE CONSTRUCTIVE MODELS OF IMPLICIT SELECTION IN BUSINESS PROCESSES

Nataliya Golyan, Vera Golyan, Olga Kalynychenko

***Abstract**: Wide introduction of the process-oriented systems for today is connected with the fact that they allow to formalize knowledge about business processes at the enterprises companies, organizations. At the same time, construction of business processes' formal models require a lot of time and material costs, and it is the subject to influence of the human factor because it is often carry out with participation of experts and executors the purposes of which can be not coincided with the purposes of modeled business processes and enterprise overall aims. The given contradiction can lead to a divergence between real business process and its developed model. It can lead to construction of inadequate models of business processes.*

One of the important approaches to the decision of the given problem is realized on the basis of methodology of business processes intelligent analysis. The intelligent analysis is directed on reception of really carried out business processes' models on the basis of research of events registration magazines of such processes.

***Keywords**: business processes, Petri net, implicit selection, logic net, predicate.*

***ACM Classification Keywords**: C.4 Performance of systems - Modeling techniques, J.1 Administrative data processing - Business*

Introduction

Nowadays we can see a transformation to the controlling process from traditional functional, this requires formalization of existing business processes for development of their hierarchic structure using typical fragments of a processes. That is why structure and characteristics refining of business processes is a necessary condition for investigation and development of intellectual data analysis methods [Agrawal, 1998].

The purpose of data analysis is to extract information from the logs of the business process in order to formalize the actual behaviour of BP. Such an analysis is particularly important when logged occurring sequence of events, but the process is partially or completely formalized – i.e. executors can make a decision on how to continue the process on the basis of existing local information and data of the business process rules. It is also important to note that the measures are taken on the basis of local information and personal knowledge of the decision may lead to deviations from the normal operation of the business process [Aalst, 2003].

Currently, a number of efficient algorithms for events analysis and logs basing on business process models were build [Medeiros, 2003], [Weijters, 2004]. But at the same time present studies can't allow to solve a serious problem in the area of intelligent analysis of the processes - the problem of implicit selection identifying in structure of business processes that require, first of all, the construction of formal models of such structures.

1 Set of a problem

In a real operating business processes we can distinguish two types of selection procedures, sequences that represent different types of dependencies: explicit and implicit.

Explicit dependences [Desel, 1995] represent the direct causal connection between the procedures. In the event log of the business process such procedures usually reflected in pairs.

Example: To start the procedure P2 is necessary to complete the procedure P1.

Implicit dependences represent the indirect cause-effect relationship between the procedures. For these relationships the relationship between P1 and P2 directly from the event log is not visible.

For example, procedure P1 and P2 can be linked through a chain of procedures <P3, P5>.

This determines the importance of formalizing the implicit selection of typical designs.

Therefore, the challenge is to get a formal algebraic structures ща implicit logic model selection based on analysis of the main features of standard procedures, sequences with an implicit choice in business processes, as well as allocation failures such fragments existing algorithms.

2 Constructs models of an implicit choice

Formally, we define the explicit and implicit dependencies between procedures of business process by means of the algebra of finite predicates.

We introduce the variables x1, x2, ..., xn, indicating the status of procedures P1, P2, ..., Pn. These variables are defined on some finite set of possible values of state procedures. For example, $x1 \in \{a, b, c\}$, where x1=a means "procedure P1 is not done", x1 = b – "procedure P1 is executing", x1 = c – "P1 procedure done".

We also introduce predicates L1, L2,…, Lk, denoting pairwise links (if they exist) between the procedures P1, P2, ..., Pn. For the above example the relationship between the procedures P1 and P2 will be described by predicate L1 (x1, x2).

Constructions, which implement implicit choice, are characterized by the contradiction between the choice of several alternatives and the need to synchronize selected action with already running in a business process. In other words, the situation is characterized by a combination of an implicit choice of designs and choice of timing, as shown in Fig. 2.1.

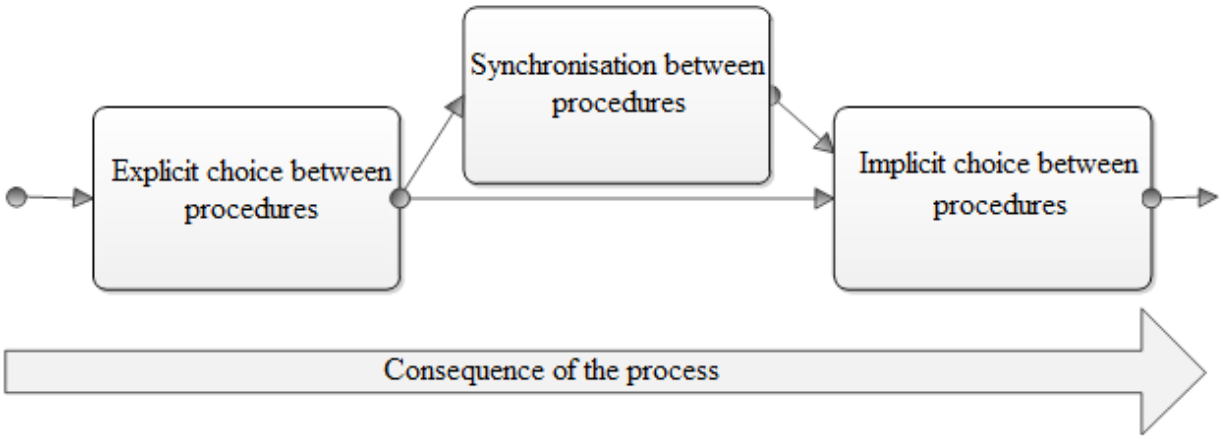


Fig . 2.1 The situation of an implicit choice between the procedures in business process

As can be seen from Fig. 2.1, the implicit choice depends on the choice of various procedures at the earlier stages of the process and is carried out after the results of the previous synchronization with the current selection procedures. This synchronization is necessary in order to have all the input conditions for the final procedures, between which there is an implicit choice in the course of a business process.

The problem of identifying structures of implicit choice in the tasks of business processes' intelligent analysis defined by the fact that the existing algorithms, the mathematical basis of which are Petri's nets, particularly their extension WF-nets [Li, 2003], usually can't process such constructions.

Consider a typical situation in more detail and implicitly selection problems that arise in identifying following situations.

In Fig. 2.2 presents the situation, according to which the final result of the current fragment of the business process depends on an implicit choice between procedures P4 and P5.

The implementation of this fragment begins in the case, the initial condition C1 is done. Then an explicit choice is executing between the procedures P1 and P2. The results of the procedures P1 or P2 must be synchronized with the result of the procedure P3, after which the choice between procedures P4 and P5 is done. Since the choice of one of the procedures determines by the choice of P1 or P2 at an earlier stage of implementation, we find that the choice between P4 and P5 is implicit.

The problem of recognition of this fragment is as follows. Existing algorithms usually do not recognize the connection between the procedures P1 and P4, P2 and P5. Therefore, the resulting model does not appear the situation is an implicit choice between P4 and P5, determined by the choice between P1 and P2. In other words, without taking into account the links between the procedures P1 and P4, P2 and P5 existing approaches to the analysis display generated model, a set of four sequences of the procedures of the form.

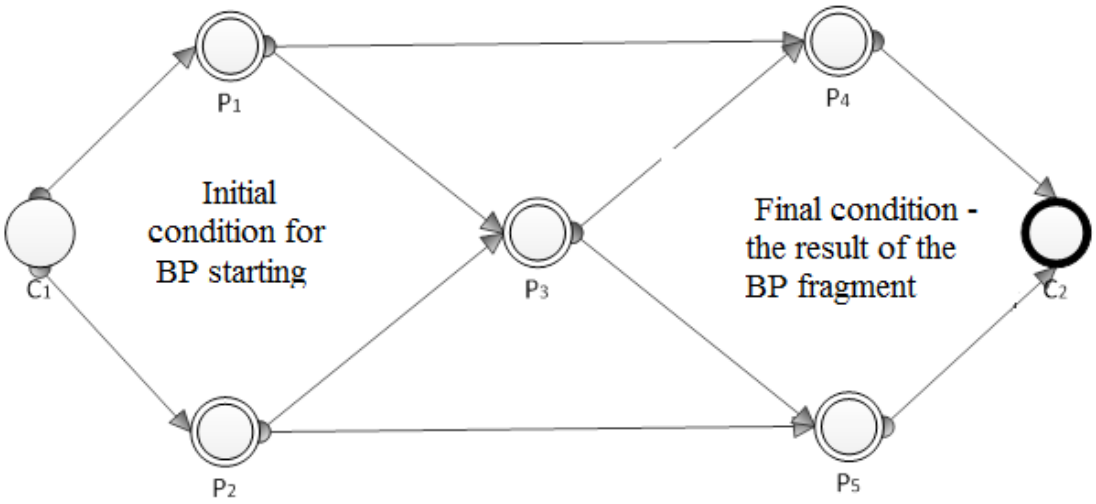


Fig. 2.2 – The situation of an implicit choice between P4 and P5

The problem of recognition of this fragment is as follows:

{ <P1, P3, P4, >, <P2, P3, P5>, <P1, P3, P5, >, <P2, P3, P4>}. (2.1)

At the same time in the real process, the links between P1 and P4, P2 and P5 valid only sequences

{ <P1, P3, P4, >, <P2, P3, P5>}. (2.2)

Way of solving this problem is as follows. The developed method for detecting design implicit choice must fix each previously made a free choice in the future track of the relationship between each pair of treatments.

Formally, this can be done with the mathematical apparatus of the algebra of finite predicates [Bondarenko, 2004] as a logical network [Shabanov-Kushnarenko, 2004], [Leschinski, 2004], ie solution to the problem of determining the optimal sequence of the procedures is reduced to solving a system of predicate equations L1, L2,…, Lk, set above.

The work of this logical network would be in a cyclic sets test of all of the modeled business process links between the procedures, which at each step of the calculations will be expressed in the generation of a set of procedures that must be done to continue the business process.

We introduce the variables x1, x2,…, x5, indicating the status of procedures P1,…, P5. These variables can take values from {0, 1}, respectively, which means performance or failure of procedures. To construct a logical network finds a system of binary predicates L1, L2,…, Lk, which describes the execution logic of the piece of business processes, shown in Fig. 2.2. For this purpose, we introduce an intermediate variable t, which contains

information about explicit and implicit election procedures. The variable t can take values from {0, 1, 2, 3} denoting respectively the performance of procedures P1, P2, P1 and P3, P2 and P3.

The system predicate that describes the logical network model of the first situation:

$$\begin{cases} L_1(x_1, x_3), \\ L_2(x_1, t), \\ L_3(x_2, x_3), \\ L_4(x_2, t), \\ L_5(x_3, t), \\ L_6(t, x_4), \\ L_7(t, x_5). \end{cases} \quad (2.3)$$

The corresponding logical network is represented in fig. 2.3:

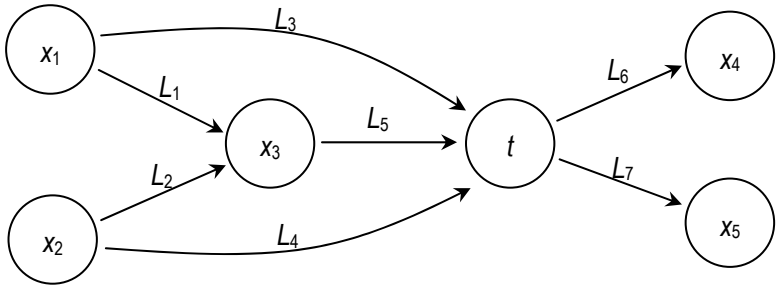


Fig. 2.3 – The logical network of the 1st typical situation

The initial logical network nodes are the variables x1, x2, final - variables x4, x5. At the beginning of the logical network of all its variables, the modeling procedure of a business process, are set to 0 - no procedure is performed. In the course of a logical network, i.e. computation corresponding to this network of binary predicates, some or all of the variables x1, x2,…, x5 logical network takes the values of 1 - all procedures are performed. The variable t takes a value corresponding to the realized in the business process execution sequence of procedures.

Consider the example of the logical network built on Measure:

0 cycle (beginning of the network): x1=0,…, x5=0, t is not defined.

1 cycle: x1=1, t=2, other nodes unchanged.

2 cycle, 1 semi cycle (network status): x3=1.

3 bar (end of the network): x4=1

Obviously, if the initial action in the network will be carrying out the procedure P2, then at the end of the network will result x5 = 1.

The second typical situation is an implicit choice of the sequence of procedures is presented in Fig. 2.4.

In this situation there are two implicit choice - between the procedures P4 and P5, as well as for P3 and P5.

Sequence of this fragment is as follows. The initial condition C1 is an input for a single procedure P1. As a result, its performance can be parallel (in any order), follow the procedures P2 and P3 respectively.

Further, there are several implementation options.

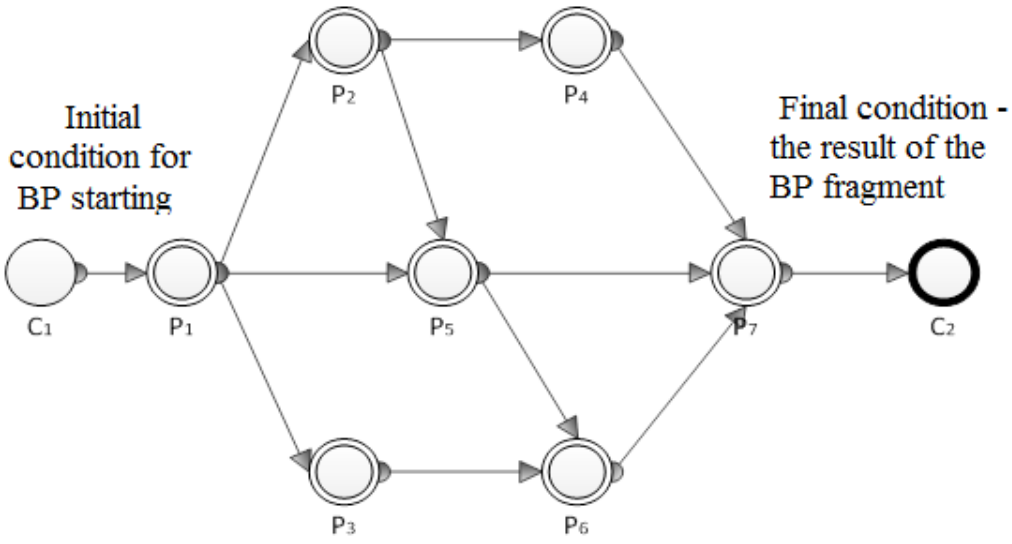


Fig. 2.4 – The situation of an implicit choice between the procedures P4 and P5 , P3 and P5

Option 1. In that case, the procedure P2 is executed before the procedure P3 then the procedure P3, P4 and P5 can be performed in any order, independently of each other. Then you can choose from two parallel branches - a procedure performed P6, or both of the procedures for P6 and P7.

Option 2. If the procedure P3 is executed before the procedure P2, then it becomes a further order of the hard-coded: P2 should be executed after the P4 and P5, P is performed after the P6, and then - P7.

Thus, the possible sets of sequences of the procedures can be written as the following tuples

{ <P1, P2, P5, P6, P7 >, <P1, P2, P3, P4, P6, P7 >, <P1, P3, P2, P6, P4, P7 >,

<P1, P2, P4, P3, P6, P7 >, <P1, P3, P6, P2, P4, P7 >} (2.3)

The problem of identifying the fragment is as follows. Existing algorithms for data analysis does not recognize the connection between the procedures P1 and P5, P5 and P7, which can lead to delays and deadlocks in process models, in particular, if a pair of parallel processes P4 and P5 will be performed only one procedure P5.

The direction of solving this problem is associated with the identification of the cause - effect relationship between the procedures P1 and P5, P5 and P7 on the basis of the analysis of event log.

We construct algebraic-logical model of the second typical situation of an implicit choice of the sequence of procedures similar to the first type situation. We introduce the variables x1, x2,..., x7, indicating the status of procedures P1,..., P7. These variables can take values from {0, 1}, respectively, which means performance or failure of procedures.

To construct a logical network determine a system of binary predicates L1, L2,..., Lk, which describes the execution logic of the piece of business processes, shown in Fig. 2.4. For this purpose, we introduce an intermediate variable t, which contains information about the sequence of explicit and implicit election procedures. The variable t can take values from {0, 1} by the rule: t = 0, if the procedure P2 is executed before the procedure P3; t = 1, on other way the procedure P2 is executed after the procedure P3.

System of binary predicate describes the logical network of the second type of implicit choice situation, consists of 12 predicates L1, L2,..., L12. The corresponding logical network shown in Fig. 2.5

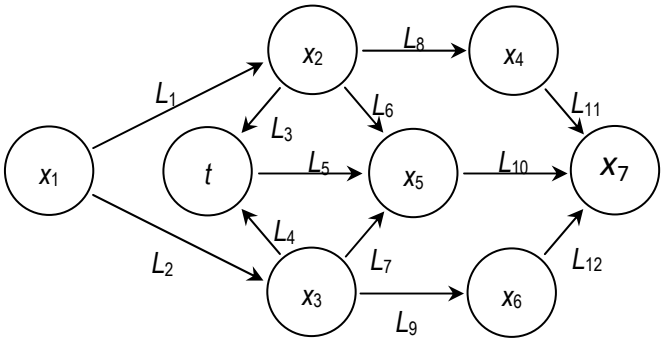


Fig. 2.5 – The logical network of the 2nd typical situation

Consider the example of the logical network built on following cycles:

0 cycle (beginning of the network): x1 = 1, all other nodes unchanged.

1 cycle: t=1, x3=1, all other nodes unchanged.

2 cycles: x2=1, all other nodes unchanged.

3 cycles: x4=1, x5=1, all other nodes unchanged.

4 cycles: x6=1, all other nodes unchanged.

5 cycles (end of the network): x7=1.

In this example, the logic network includes the issue of the procedures appropriate to the 2nd version of the procedure P2 executed after the procedure P3. Cycle 1 is calculated similarly.

Business process model in the form of a logical network, as shown above, generates at any given time a number of procedures that must be done to continue the business process. When performing any procedure, the situation changes once that immediately displays the change in the state of the logical network and thus change the set of procedures that must be done to continue the business process.

The third typical situation an implicit choice of the sequence of procedures is presented in Fig. 2.6.

In this situation, there is an implicit choice between the procedures P3 and P5. Sequence of this fragment is as follows. After the procedure P1 independently can be performed procedures P2 and P3. The final procedure for this fragment, P5 can be executed after the cycle P3 → P4 and the procedure P2.

As we can see, the possible sets of sequences of the procedures can be represented by the following tuples:

{ < P1, P2, P5 >, <P1, P2, P3, P4, P5>,

<P1, P3, P2, P4, P5>, <P1, P3, P4, …, P3, P4, P2, P5>}. (2.5)

The solution of the problem considered in this case involves finding the relationship between the procedures P1 and P5.

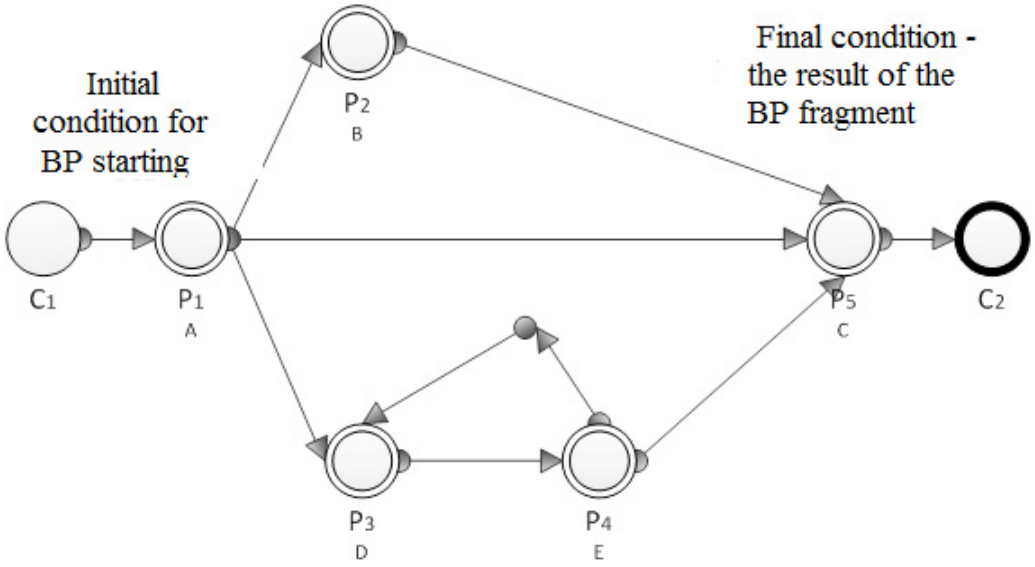


Fig. 2.6 – The situation of an implicit choice between the procedures P3 and P5

Lets construct a logical network of third typical situation an implicit choice of the sequence of procedures similar to the situation considered above. We introduce the variables x1, x2,…, x5, indicating the status of procedures P1,…, P5. We find the system binary predicate L1, L2,…, Lk, which describes the execution logic of the movie business processes depicted in Fig. 2.6. For this purpose, we introduce an intermediate variable t, which contains information about implicit choice between procedures P3 and P5. The variable t can take values from {0, 1} by the

rule: t = 0, if the procedure is executed before the procedure P3 P5; t = 1, if the procedure is executed after the procedure P3 P5.

System of binary predicate that describes the logical network of the second type of implicit choice situation, consists of seven predicates L1, L2, ..., L7. The corresponding logical network shown in Fig. 2.7:

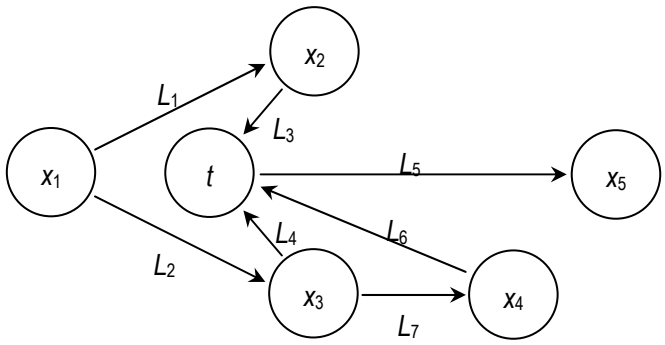


Fig. 2.7 – Logical network of third typical situation

Consider the example of the logical network built on following cycles:

0 cycle (beginning of the network): x1=1, all other nodes unchanged

1 cycle: x2=1, all other nodes unchanged.

2 cycles: x3=1, t=0, the remaining nodes unchanged.

3 cycles: x4=1, all other nodes unchanged.

4 cycles (end of the network): x5=1.

Conclusions

Algebraic-logical models of implicit choice are logical nets have been shown as a predicates system, which depicts correlation between procedures of indirect construction. Such models allow depict implicit connections between existing procedures, which are not directly shown in an event of business process

Using the examples of built logical nets we can see, that in major amount of cycles counts only one predicate, that means that such logical nets work in sequence mode, and they have a prior to other models of same processes of vary predicates. However in shown examples were studied only small "key" fragments of existing business processes, and on practice models usually include many procedures, which can be executed at the same time, in that case usage of model with binary equations system is obvious.

The second feature of built logical net is in similar introduction of additional variable, which characterizes all deviations in work logics of a business process and models are showing the sequence of procedures execution.

Practical aspect of received results is in follows. Realization of a logical net, which realize construction of implicit choice, allows receive a log of event registration which depicts implicit relations between procedures, which gives a condition for development of detecting methods for constructions with implicit choice

Bibliography

[Agrawal, 1998] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In Sixth International Conference on Extending Database Technology, pages 469-483, 1998.

[Aalst, 2003] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering, 47(2):237-267, 2003.

[Medeiros, 2003] A.K.A. de Medeiros, W.M.P. van der Aalst, and A.J.M.M. Weijters. Workflow Mining: Current Status and Future Directions. In R. Meersman, Z. Tari, and D.C. Schmidt, editors, On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, volume 2888 of Lecture Notes in Computer Science, pages 389 -406. Springer-Verlag, Berlin, 2003.

[Weijters, 2004] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128-1142, 2004.

[Desel, 1995] J. Desel and J. Esparza. Free Choice Petri Nets, volume 40 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, UK, 1995.-256 p.

[Li, 2003] M. Z. J.Q. Li, Y.S. Fan. Timing constraint workflow nets for workflow analysis. IEEE Transactions on Systems, Man, and Cybernetics, part A: Systems and Humans, 33(2):179–193, March 2003.

[Bondarenko, 2004] M.F. Bondarenko, Yu. P. Shabanov-Kushnarenko About algebra of predicates // Problems of bionics, 2004. - Vol. 1. – p. 176-185.

[Shabanov-Kushnarenko, 2004] S.Yu. Shabanov-Kushnarenko, M.F. Bondarenko, V.A. Chikina, Yu. P. Shabanov-Kushnarenko Models of Languages // Problems of bionics, 2004. - Vol. 1. – p. 128-139.

[Leschinski, 2004] V.A. Leschinski, S.Yu. Shabanov-Kushnarenko, M.F. Bondarenko, Z.V. Dudar, I.A. Yefimova About Brainlike Machines // Radioeletronics and Informatics. 2004. № 2. С. 12-20.

Authors' Information

Olga Kalynychenko - PhD, associate professor of Software department, Kharkov National University of Radio Electronics Ukraine; Lenin av., 14, 61166 Kharkov, Ukraine; e-mail: okalinichenko@mail.ru

Vera Golian - PhD, Software department, Kharkov National University of Radio Electronics Ukraine; Lenin av., 14, 61166 Kharkov, Ukraine; e-mail: veragolyan@yandex.ru

Nataliya Golian – PhD student of Software department, Kharkov National University of Radio Electronics Ukraine; Lenin av., 14, 61166 Kharkov, Ukraine; e-mail: veragolyan@yandex.ru

EXTRACTING BUSINESS RULES – HYPE OR HOPE FOR SOFTWARE MODERNIZATION

Neli Maneva, Krassimir Manev

***Abstract**: The purpose of this work is to investigate how the Business Rules approach (very promising) can be used for the purposes of feasible and efficient software modernization (very needed). The processes of the forward and backward reengineering of legacy software systems are presented through identification of the main players. Their responsibilities, assigned tasks and expected deliverables are described. Some reasons for need of instrumental tools to support the process are given.*

***Keywords:** software development, software modernization, business rules, extracting business rules.*

***ACM Classification Keywords:** D.2.1 Requirements/Specifications, D.2.7 Distribution, Maintenance, and Enhancement.*

Introduction

According to the statistical data, since 2009 more than 80% of software applications are "remake" of already existing, but older or obsolete applications [Jones, 2010]. Such *legacy software systems* [Dictionary, 2012] still have business importance, but are expensive to manage and difficult to modify, with poor competitiveness and compatibility with modern equivalents. This explains the current intensive research and development work on software modernization. Experts, involved in software modernization, have borrowed a number of general and validated Software Engineering (SE) methods and tools, but in order to solve specific problems, they create their own approaches [Comella-Dorda, 2000]. For example, to make some unavoidable changes and to meet new requirements, the achievements of Requirements Engineering theory and practice can be used as a base, but should be applied within a modified process, because of insufficient and/or lost expert's knowledge. The missing information (including algorithms and business rules) should be extracted from selected legacy system artifacts and together with the identified new requirements, should be built into the modernized application.

The paper is devoted to some problems of introducing a Business Rule (BR) approach to a specific software engineering activity, namely software modernization [Hay&Healy, 2000]. Section 2 describes the main players and their roles within a BR–based software development, which can be considered as a proactive approach, facilitating further maintenance and modernization of the developed software system. In Section 3 the conception

of the BR-based software modernization as reverse engineering is presented. Some perspectives for computer-aided support to a number of the considered activities are briefly commented in Section 4. In Conclusion the pros and cons of the proposed approach are summarized and a few ideas for future research and development work are mentioned.

BR-based software development

Development of software with usage of business rules is very close to the traditional one – just much more attention is dedicated to the rigorous analysis of the enterprise policy. As a result of this analysis a set of business rules is constructed. Those of business rules, influencing in any way some aspects of the software system under development, should be identified, separated in a group and further processed as special (BR-generated) system's requirements. For better understanding, the process of elicitation and analysis of such requirements can be presented through the work flow and data exchange among the involved participants. Using the classification of WebSphere ILOG BRMS of IBM [Stineman, 2009], we rank the distinguished 5 main roles of people, taking part in the BR-based development of a software system, as follows:

- Business Analyst – responsible for domain modeling and preparing the business rules vocabulary;
- Policy Manager – translates the business policy into detailed BR. For our purposes the Policy Manager is representative of the business people who (following the *Zachman framework)* are involved in the process of creating the business application [Zachman International, 2008];
- Software Architect – responsible for software system architectural design;
- Software Developers – create and test the application, iteratively adding new functionality, as required;
- System Administrator – observes and controls the running application in order to achieve the stated goals.

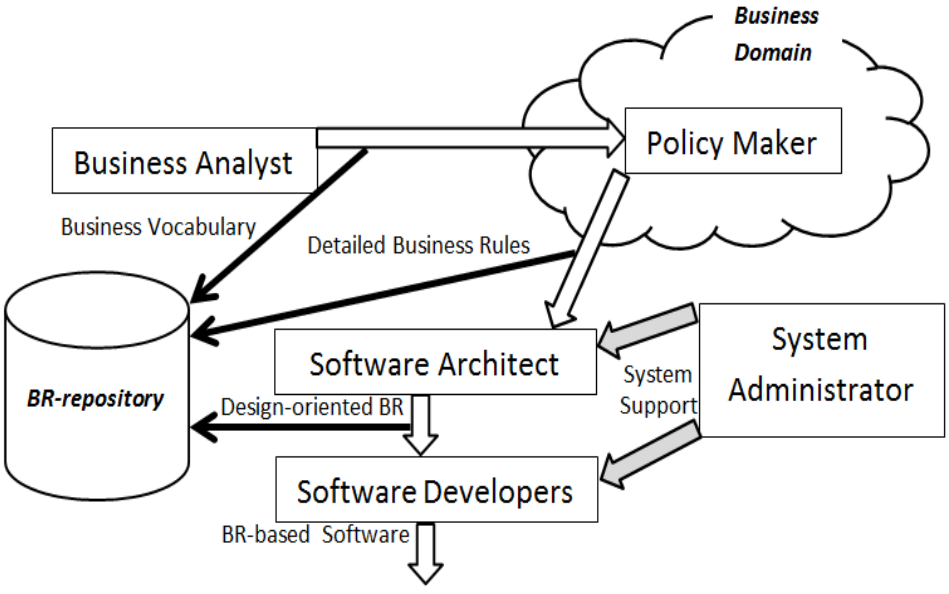


Fig. 1. BR-based software development

In Fig.1 the double arrows demonstrate the main work stream of the forward process of software development. The single arrows show which deliverables, obtained as results of a participant's activity during the different steps of process, will be saved in the BR-repository.

Next follows a brief description of responsibilities for two roles, specific for BR processing.

Business analyst is a person with an appropriate background, qualification and experience, who supplies the vocabulary of the business domain. We assume that in general (s)he should be an expert, hired as an external consultant for the project under consideration. The independence of the business analyst has to guarantee that the vocabulary is as complete as possible and acceptable for each business subject in the domain.

Policy Maker of the enterprise formulates the management policy of a given enterprise and expresses it in a set of business rules with the desired level of details. The Vocabulary and the set of business rules, represented by appropriate formalism (see for example [Hay&Healy, 2000]), are saved in the repository and can be used during the detailed design of the enterprise's software system.

After this initial phase of requirements definition (including BR-generated ones) the further development process can be quite similar to the traditional one, slightly modified in order to reflect some peculiarities, imposed by the business rules [Andreescu, 2009].

BR-based software modernization – BR extraction from the software system

Modernization of a legacy software system with usage of business rules could be split in three major stages – Extracting the built in the legacy software business rules, Modernization of the extracted rules and Re-design and implementation of the system on the base of modernized business rules. Obviously, the most challenging of these three stages is the first one. That is why we will concentrate on it. The Fig. 2 schematically shows one possible way of process organization through description of the roles of the participating people and the created deliverables.

It is clear that the role of the ***Business Analyst*** will be changed a bit during the process of BR extraction from the legacy software system. Preparing of adequate and acceptable business vocabulary will not be enough for successful extraction of business rules. As the classification of BR in [Hay&Healy, 2000] suggests, instead of vocabulary, an *ontology of the business domain* has to be applied. Here we will use the term *ontology* as it is understood in computer science and information technologies, i.e. an ontology is consisting of "the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them" [Genesereth&Nilsson, 1987]). In such a way mapping of candidate business rules, extracted from the legacy system, to sentences in the language of the business people has to be much more easy.

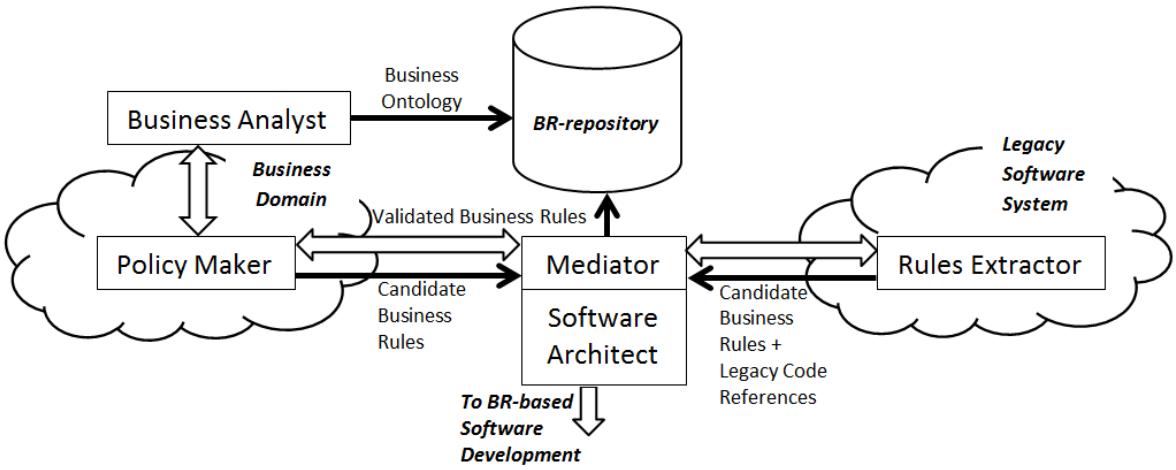


Fig. 2. Extracting BR from Legacy Software System

Nowadays there are well developed and validated ontologies of many domains that are actively used in *knowledge-based* (or *expert*) *systems*. Usage of ontologies became even more popular with introducing the *semantic Web* concept some years ago and so there is very high level expertize in this area. That is why it will be relatively easy to find or elaborate the necessary ontology for the BR-based modernization of software systems in an arbitrary area. By the way, usage of ontology instead of a simple vocabulary of the domain will be much more helpful in the design and development of BR-based software system when starting their development from scratch.

If the owners of the business would like to have a modern BR-driven software system, they have sooner or later to describe their main business policy in a form of detailed business rules. Such task is intrinsic for the ***Policy Maker*** who in our concept is a representative of the owners. His role in the process of BR extraction from a legacy software system is double. From one side, the role will be practically the same as in the process of BR-development from scratch. The Policy Maker is translating the business policy of owners into business rules again, using formal description and the ontology of the domain. But from the other side, the Policy Maker has to try to define as more as possible candidate business rules, which, it is supposed, are implemented in the legacy software system. For making our concept clearer, it will be better in this case to rename the role of Policy Maker to ***Policy Translator***.

An important new role in the process of BR extraction from a legacy software system will be the ***Rules Extractor***. His task is to mine available software system resources (source code, executables, documentation, data bases, etc.) and to try to formulate some candidates for business rules, incorporated in the software system. These rules will be expressed in the terms of the software system (variable and function names, attributes of data base tables,

caption of elements in screen forms, etc.) but with the same formalism, used by the Policy Translator. It is very important that to each candidate rule, extracted from the system, there will be an assigned respective reference to the software system. In such a way, if the extracted rule has to be changed in the modernized version of the system, Software Developers will be able to localize the component of the system to be rewritten.

Even more important new role in the process of BR extraction from a legacy system is the ***Mediator***. Let *A* be the set of candidate rules specified by the Policy Translator and *B* be the set of candidate rules, extracted from the system by the Rules Extractor. The task of the Mediator will be to find a candidate rule *p* from *A* and a candidate rule *q* from *B,* which seems to be adequate. Let r_q be the reference assigned to *q* by the Rules Extractor. We will call the couple (p, r_q) *extracted rule.* As many rules are extracted from the legacy system, as easy will be the work of Mediator during the next stage of modernization, does not matter how the process will be performed – with rewriting parts of the system or from scratch.

The work of the Mediator could not be successfully done without the help of people who know in detail the legacy code and platform. In the Fig. 2 this role is denoted by Software Architect with the clear understanding that there is a team of people that stay behind it. In practice this will be the IT team of the enterprise that exploits and maintains the system and we will call it ***the team of Software Architect***. It is quite possible some people from the team of Policy Maker to be involved in activities from this stage of the extraction process, too.

When the process of BR extraction from the legacy system is completed, the extracted rules have to be checked for compliance with the new requirements to the enterprise software. Some of the extracted rules will **contradict the new requirements**. The elimination of these rules will be crucial for the design of the modernized system. If it is possible to eliminate the part of the system that implements these rules then the modernization could be made by introducing some changes. Else, the system probably has to be rewritten from scratch.

If the system will not be rewritten from scratch then it is worth to outline the extracted rules that **deviate from the new requirements.** This will give the team of Software Architect the possibility to identify the parts of the legacy system which will be affected by modernization. Finally, a set of these BR is composed that meets to a satisfying degree the new requirements to the enterprise system. Then the process could continue as in the case of the normal BR-based software development.

We would like to stress a major difference between the two processes. The BR-based design and development is almost a linear process and the outlined steps of this process are executed consecutively. The BR-based modernization of a legacy software system, as demonstrated above, is much more sophisticated from managerial point of view. In some cases it will be very difficult to organize the process properly - especially its part, dedicated to BR extracting from the system through a clearly defined tasks sequence. The opposite, our expectations are that different activities of the extraction will be performed in parallel, mutually challenging and influencing each other.

How to automate the BR extraction from a legacy system

As it was shown above, the process of BR-based modernization of legacy software system comprises a number of nontrivial tasks. It is difficult to imagine that these tasks could be solved manually. The main goal of this paper is to investigate the process of legacy system modernization so as to outline the tasks or subtasks, for which some special-purpose software tools will be necessary. As the resources that the legacy software system provide in order to help the process of modernization will be the inputs of such software tools, let us first identify the most important of them (in decreasing order of their importance):

Source code of the legacy system – this is the **most valuable resource** of the legacy system for the BR-based modernization process, even if it is written in an exotic programming language and for an exotic platform. Being a totally formal object, the source code is the most appropriate for automated mining. Something more, many software tools have been developed for years with different purposes – calculating software metrics, transparent-box software testing, etc. Some well developed methods, as **static** and **dynamic analysis**, could be tuned for the purposes of the BR extraction from code. Manual mining in the source code of the legacy system could be necessary in some very specific cases, e.g. when a formal grammar or any other formal description of the used language is missing, as well as there no available working compiler for checking some hypothesis about unknown elements of the syntax.

Executable modules of the system could be the most valuable resource, when the system's **source code is not available,** but if and only if the ***platform,*** necessary to run executables is available. Executables could be used in two ways. First, to use them without modifications, i.e. as they are. This is very close to the usage of executables in nontransparent (or black box) software testing. Such approach is also well developed and could be adapted for BR extraction. The second possibility to use executables is to reassembly them and to try to use the obtained result as a source code. No doubt, extraction of rules from code, written in assembly language, will be more difficult and algorithmically different from rules extraction from code, written in high level programming language. In this case it is worth to consider translation from assembly language to a very close to assembly high level programming language (kernel C, for example). Even better results have to be expected when the extracting process is **performed in parallel** on the source code and on the executables.

Data base of the legacy system could be valuable source for extracting rules. Even not working data base could be helpful with its data base scheme and the formal description of the tables, especially when the attributes of the records have some mnemonic names. Rules extraction from the structure of the base is relatively easy and could be done manually or with a simple program. More interesting and more difficult task will be extracting rules from SQL scripts. Unfortunately, the approaches for rules extraction from high level programming languages will be probably unusable for SQL scripts.

As it was shown, the three above mentioned resources are suitable for different by hardness and used algorithms automation. The corresponding programs will form the toolkit of the agent, called Rules Extractor. From the other

side of the process – those of Police Maker – automation is almost impossible. The candidate rules that Police Maker will propose will be extracted manually from two sources. The first of them is the ***knowledge of the staff*** about the legacy software that has to be collected and formulated as candidate rules by interviews with all level staff, but especially with the people that exploit and maintain the software.

The other resource that has to be explored by the team of Policy Maker is the ***system maintenance documentation*** and ***user manuals*** of the legacy system (if they exist and are with good quality). Some automation here is also possible – there are examples of intelligent text processors that could be tuned to search in documentation in digital form. We think that this work requires intelligence and special competences and will be done rather manually.

The final stage of BR extraction covers the tasks of the Mediator's team. Suppose that the two set of candidate rules are expressed by the same formalism and the difference is that the first set uses the words of the ontology and the other – the mnemonic names of the implementation. Then a mapping of mnemonic names into the terms of the ontology has to be done in order to obtain a set of business rules and each rule is assigned to the place(s) in the software where it is implemented. It seems that this activity could be automated, using the knowledge and experience of the researchers in the domain of ontologies generation and maintenance.

Conclusion

The paper presents an initial study how the BR approach can be integrated into the process of software modernization. The main contribution is that the BR-related activities are described in the context of working environment, in which the participants with predefined roles are responsible for a set of dynamically assigned tasks, accomplished through participatory teamwork.

A number of hard problems, hampering the real life implementation of the proposed approach, can be seen even in this initial stage of research. For example, it will be difficult to select the most appropriate presentation and adequate transformation of the BR, shared among participants with quite different background and decision-making status. There are some very hard pure technological problems, e.g. how to organize the BR repository so as to assure both easy and fast access to the saved items, etc.

We intend to continue the research, trying to combine the BR approach with the described in [Maneva&al., 2011] model-based software modernization.

Acknowledgments

This work is supported by the National Scientific Research Fund of Bulgaria under the Contract ДТК 02-69/2009.

Bibliography

[Andreescu, 2009] A. Andreescu. A General Software Development Process Suitable for Explicit Manipulation of Business Rules. In: International Conference on Computer Systems and Technologies – CompSysTech'09, 2009.

[Comella-Dorda&al., 2000] S. Comella-Dorda, K. Wallnau, R. Seacord and J. Robert. A Survey of Legacy System Modernization Approaches. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2000.

[Dictionary, 2012] Legacy System, http://dictionary.reference.com/browse/legacy+system, visited 15.02.2012.

[Genesereth&Nilsson, 1987], M. R. Genesereth, N. Nilsson. Logical Foundations of Artificial Intelligence. Morgan Kaufmann Publishers: San Mateo, CA, 1987.

[Hay&Healy, 2000] D. Hay and K. A. Healy (eds.). Defining Business Rules ~ What Are They Really? GUIDE Business Rules Project Final Report, rev. 1.3., July, 2000.

[Jones, 2010] C. Jones. Software Engineering Best Practices. Mc Grow Hill, 2010.

[Maneva&al., 2011] N. Maneva, Kr. Kraychev and Kr. Manev. Mathematical Models-Based Software Modernization. In: Mathematika Balkanika J., New Series, vol.25, No 1-2(2011), pp. 131-144.

[Stineman, 2009] B. Stineman, IBM WebSphere ILOG Business Rule Management Systems: The Case for Architects and Developers. IBM Software Group, November 2009.

http://www-01.ibm.com/software/websphere/products/business-rule-management/, visited 15.02.2012.

[Zachman International, 2008] The Zachman Framework: The Official Concise Definition. Zachman International, 2008.

http://zachman.com/about-the-zachman-framework, visited 15.02.2012.

Authors' Information

Neli Maneva *– Associated Professor, Ph.D., Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 8, Sofia 1113, Bulgaria; e-mail: neman@math.bas.bg*

Major Fields of Scientific Research: Software Engineering, Software Quality Assurance, Model-driven software development.

Krassimir Manev *– Associated Professor, Ph.D., Department of Mathematics and Informatics, Sofia University, 5 J. Bourchier blvd., Sofia-1164, Bulgaria; e-mail: manev@fmi.uni-sofia.bg*

Major Fields of Scientific Research: Discrete mathematics and Algorithms, Formal Methods in Software Engineering, Source Code Analysis, Software Testing.

TABLE OF CONTENTS OF IJ ITA, VOL.18, 2011

Contents of IJ ITA, Vol.18, No.:1, 2011

Contents of IJ ITA, Vol.18, No.:2, 2011

Contents of IJ ITA, Vol.18, No.:3, 2011

TABLE OF CONTENTS OF IJ ITA, VOL.18, NO.:4, 2011