ITHEA

International Journal
INFORMATION THEORIES & APPLICATIONS

2008 Volume 15 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
Volume 15 / 2008, Number 3

IJ ITA is official publisher of the scientific papers of the members of the ITHEA International Scientific Society, the Association of Developers and Users of Intellectualized Systems (ADUIS) and the Association for Development of the Information Society (ADIS)

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*.
The camera-ready copy of the paper should be received by e-mail: *info@foibg.com*.
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.15, Number 3, 2008

Printed in Bulgaria

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

Publisher: **Institute of Information Theories and Applications FOI 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)**

HYPER-RANDOM PHENOMENA: DEFINITION AND DESCRIPTION

Igor Gorban

Abstract: *The paper is dedicated to the theory which describes physical phenomena in non-constant statistical conditions. The theory is a new direction in probability theory and mathematical statistics that gives new possibilities for presentation of physical world by hyper-random models. These models take into consideration the changing of object's properties, as well as uncertainty of statistical conditions.*

Keywords: *uncertainty, random, hyper-random, phenomenon, probability, statistics.*

Introduction

The most of physical phenomena (electrical, electromagnetic, mechanical, acoustics, nuclear, and others) are an indeterminate type. Usually, different stochastic methods are used to describe them. However, possibilities of such methods are limited. There are serious problems, when the observation conditions are changed in space or time and it is impossible to determine the statistical regularity, even by a large experimental sample size.

The changed conditions are met everywhere. It is impossible to image any real event, value, process or field in absolutely fixed conditions. All mass measures are led in the variable conditions, controlled only partly.

The fixed statistical condition and the probability measure are linked together. When it is said the fixed (constant) condition about it is meant that there is the probability measure for every samples of the researched set.

When a physical phenomenon is observed in more or less invariable condition, there is possibility to achieve the statistical regular results. However, if the condition is changed in wide bounds, the statistical estimates are not stable and it is impossible to obtain probable estimates.

To image a depth of the problem, let us apply to well known classic task with tossing a coin. The stability of head or tail (A or B) essentially depends from the style of tossing [1]. In a fixed statistical condition there are stable event frequencies $p_N(A)$, $p_N(B)$, which tend to any probabilities $P(A)$, $P(B)$, when the number of experiments N is tend to infinity. In case of variable condition, the frequencies $p_N(A)$ and $p_N(B)$ are continuously changed. They oscillate in any intervals and not tend to any fixed probabilities.

The condition stability plays the important role in the probability theory that marked by a number of scientists, beginning from Jakob Bernoulli [2]. R. von Mises proposed even to define [3] the probability conception on the base of the event frequency in fixed condition.

It is not simple to determine correctly a probability measure for real physical phenomenon. This fact was marked in many works, for instance, in the article [4].

Difficulties and often impossibility to use the probability theory stimulate the developing of new theories, such as fuzzy logic [5], neural network [6], chaotic dynamical systems [7], and others. The new theory of hyper-random phenomena, the bases of which are presented below, may be included to this list.

The aim of the paper is to review the original author's researches published in articles [8 – 15] and generalized in the monograph [16].

The theory is oriented to description of different type uncertainty, as a contingency, when the probability measure exists, as another one, when the probability measure does not exist.

In modern mathematics, the random phenomena are defined by the probability field that assigned by the triad $(\Omega, \Im, P)$, where Ω represents the set of the simple events $\omega \in \Omega$, $\Im$ – the Borel field, P – the probability measure of the subsets.

The hyper-random phenomena may be defined by the tetrad $(\Omega, \Im, G, P_g)$ [9], where Ω and $\Im$ are the set of simple events and the Borel field (as in the case of the probability field), G is the set of the conditions $g \in G$, and P_g – the probability distributions for the condition g .

Any hyper-random phenomena (events, variables, functions) may be regarded as a set (family) of random subsets. In this construction, every subset is associated with any fixed observation condition. The probability measures are determined for the elements of each subset; however, the measures are not determined for the subsets of the set.

Hyper-random Events and Variables

The hyper-random event A from the Borel field $\Im$ cannot be described by any probability. However, the event A/g under the condition $g \in G$ may be presented by the probability $P_g(A) = P(A/g)$. This probability oscillates when condition is changed. The range of the oscillation may be described by the supremum $P_S(A)$ and the infimum $P_I(A)$ of the event probability defined as

$$P_S(A) = \sup_{g \in G} P(A/g), \quad P_I(A) = \inf_{g \in G} P(A/g).$$

In the constant condition ($g = \text{const}$) these bounds are congruent and the hyper-random event degenerates to the random one with the probability $P(A) = P_S(A) = P_I(A)$.

The bounds $P_S(A)$, $P_I(A)$ are half-measures. There has been obtained the expressions that are similar to the formulas, describing the product and the addition rules, the Bayes' and other theorems of the probability theory.

To describe the scalar hyper-random variable X a number of the characteristics have been proposed. They are similar to the probability characteristics of a random variable. The main of them are the supremum $F_S(x)$ and the infimum $F_I(x)$ bounds of the distribution function and also the probability density functions $f_S(x)$ and $f_I(x)$ of these bounds. They are determined by the following expressions:

$$F_S(x) = \sup_{g \in G} P\{X \le x/g\}, \quad F_I(x) = \inf_{g \in G} P\{X \le x/g\},$$

$$f_S(x) = \frac{dF_S(x)}{dx}, \qquad f_I(x) = \frac{dF_I(x)}{dx},$$

where $P\{X \le x/g\}$ is the probability of the inequality $X \le x$ for the condition g.

It has been found that the bounds of the distribution function and the probability density functions of the bounds for hyper-random variable have the same particularities as according characteristics for a random variable and in addition $F_S(x) \ge F_I(x)$.

Among the bounds of the distribution function there is a zone of the ambiguity (fig. 1). For a random variable X its width $\Delta F(x) = F_S(x) - F_I(x)$ equals to zero for all x. If the supremum $F_S(x)$ of the distribution function is tend for all x to unit and the infimum $F_I(x)$ – to zero, zone of the ambiguity is tend to maximum. In this case, the hyper-random variable approaches to a chaos one.

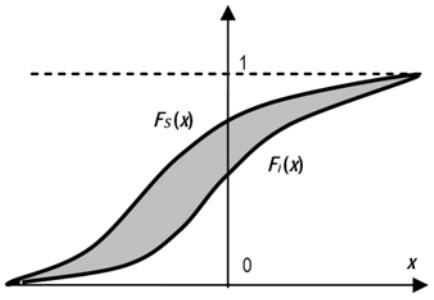


Fig. 1. The bounds of the distribution function and the zone of the ambiguity (the black-out part)

To describe the hyper-random variable, the characteristics similar to random variable ones, may be used. They are the bound's crude and the central moments determined for the hyper-random variable X on the base of the bound's expectation $\mathrm{M}_S[\varphi(X)]$, $\mathrm{M}_I[\varphi(X)]$ of the function $\varphi(X)$:

$$\mathrm{M}_S[\varphi(X)] = \int_{-\infty}^{\infty} \varphi(x) f_S(x) \mathrm{d}x, \quad \mathrm{M}_I[\varphi(X)] = \int_{-\infty}^{\infty} \varphi(x) f_I(x) \mathrm{d}x.$$

In particular, the bound's means m_{Sx}, m_{Ix} are $m_{Sx} = \mathrm{M}_S\left[X\right]$, $m_{Ix} = \mathrm{M}_I\left[X\right]$. For the real hyper-random variable X the bound's variance D_{Sx}, D_{Ix} are $D_{Sx} = \mathrm{M}_S\left[(X - m_{Sx})^2\right]$, $D_{Ix} = \mathrm{M}_I\left[(X - m_{Ix})^2\right]$. The bound's crude moments are determined by the expressions $m_{Sxv}=\mathrm{M}_S\left[X^{\nu}\right]$, $m_{Ixv}=\mathrm{M}_I\left[X^{\nu}\right]$ and the bound's central moments – by the ones $\mu_{Sxv}=\mathrm{M}_S\left[(X - m_{Sx})^{\nu}\right]$, $\mu_{Ixv}=\mathrm{M}_I\left[(X - m_{Ix})^{\nu}\right]$, where ν is the order of the moment.

To describe the hyper-random variables other type characteristics may be used too. They are supremum and infimum of the crude moments and the same bounds of the central moments. These characteristics are determined on the base of the expectation of the function $\varphi(X/g)$:

$$\mathrm{M}_s[\varphi(X/g)] = \sup_{g\in G}\int_{-\infty}^{\infty}\varphi(x/g)f(x/g)\mathrm{d}x, \quad \mathrm{M}_i[\varphi(X/g)] = \inf_{g\in G}\int_{-\infty}^{\infty}\varphi(x/g)f(x/g)\mathrm{d}x,$$

where $f(x/g)$ is the probability density function in condition g.

In particular, for the hyper-random variable X the supremum m_{sx} and the infimum m_{ix} of the mean are $m_{sx}=\mathrm{M}_s[X]$, $m_{ix}=\mathrm{M}_i[X]$, and the supremum D_{sx} and the infimum D_{ix} of the variance are $D_{sx}=\mathrm{M}_s[(X - m_{x/g})^2]$, $D_{ix}=\mathrm{M}_i[(X - m_{x/g})^2]$, where $m_{x/g}$ represents the mean of the random variable X/g. The crude moment's bounds m_{sxv} and m_{ixv} are described by the expressions $m_{sxv}=\mathrm{M}_s\left[X^{\nu}\right]$, $m_{ixv}=\mathrm{M}_i\left[X^{\nu}\right]$, and the central moment's bounds μ_{sxv} and μ_{ixv} – by the following ones $\mu_{sxv}=\mathrm{M}_s[(X - m_{x/g})^{\nu}]$, $\mu_{ixv}=\mathrm{M}_i[(X - m_{x/g})^{\nu}]$.

In general, the operators $\mathrm{M}_S\left[\cdot\right]$, $\mathrm{M}_I\left[\cdot\right]$ differ from the operators $\mathrm{M}_s\left[\cdot\right]$, $\mathrm{M}_i\left[\cdot\right]$ and the bound's moments differ from the moment's bounds, although in some particular cases they may be expressed by each other, for instance, when the distribution functions $F(x/g)$ for different conditions g have not interception points. Then, if the variance $D_{x/g}$ is raised with raising the mean $m_{x/g}$ ("a" type distribution) there are the following equalities: $m_{Sx} = m_{ix}$, $m_{Ix} = m_{sx}$, $D_{Sx} = D_{ix}$, $D_{Ix} = D_{sx}$; if the variance $D_{x/g}$ is reduced with raising the mean $m_{x/g}$ ("b" type distribution) there are the equalities: $m_{Sx} = m_{ix}$, $m_{Ix} = m_{sx}$, $D_{Sx} = D_{sx}$, $D_{Ix} = D_{ix}$.

The results were generalized to complex $\dot{X}$ and vector $\vec{X}$ hyper-random variables, to real $X(t)$, complex $\dot{X}(t)$, and vector $\vec{X}(t)$ functions.

Hyper-random Functions

The scalar hyper-random process $X(t)$ has been presented as a family of the random processes $X(t)/g$ determined for a set conditions $g \in G$. The process described by the supremum $F_S(\vec{x};\vec{t})$ and the infimum $F_I(\vec{x};\vec{t})$ of the distribution function, probability density functions $f_S(\vec{x};\vec{t})$, $f_I(\vec{x};\vec{t})$ of these bounds, the bound's moments $m_{Sx\vec{\nu}}(\vec{t})$, $m_{Ix\vec{\nu}}(\vec{t})$, $\mu_{Sx\vec{\nu}}(\vec{t})$, $\mu_{Ix\vec{\nu}}(\vec{t})$ and the moment's bounds $m_{sx\vec{\nu}}(\vec{t})$, $m_{ix\vec{\nu}}(\vec{t})$, $\mu_{sx\vec{\nu}}(\vec{t})$, $\mu_{ix\vec{\nu}}(\vec{t})$, where $\vec{\nu}=(\nu_1,\ldots,\nu_L)$ is the order vector of the moment, L is the measure of the distribution. These characteristics are described by expressions that are similar to ones for hyper-random variable:

$$F_S(\vec{x};\vec{t}) = \sup_{g\in G} P\{X(t_1)\le x_1,\ldots,X(t_L)\le x_L/g\}, \quad F_I(\vec{x};\vec{t}) = \inf_{g\in G} P\{X(t_1)\le x_1,\ldots,X(t_L)\le x_L/g\},$$

$$f_S(\vec{x};\vec{t}) = \frac{\partial^L F_S(\vec{x};\vec{t})}{\partial x_1 ... \partial x_L}, \quad f_I(\vec{x};\vec{t}) = \frac{\partial^L F_I(\vec{x};\vec{t})}{\partial x_1 ... \partial x_L},$$

$$m_{Sx}(t) = \mathrm{M}_S[X(t)] = \int_{-\infty}^{\infty} x f_S(x;t)dx, \quad m_{Ix}(t) = \mathrm{M}_I[X(t)] = \int_{-\infty}^{\infty} x f_I(x;t)dx,$$

$$m_{Sx\nu_1...\nu_L}(t_1,\ldots,t_L) = \mathrm{M}_S[X^{\nu_1}(t_1)\ldots X^{\nu_L}(t_L)] = \int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} x_1^{\nu_1}\ldots x_L^{\nu_L} f_S(x_1,\ldots,x_L;t_1,\ldots,t_L)dx_1\ldots dx_L,$$

$$m_{Ix\nu_1...\nu_L}(t_1,\ldots,t_L) = \mathrm{M}_I[X^{\nu_1}(t_1)\ldots X^{\nu_L}(t_L)] = \int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} x_1^{\nu_1}\ldots x_L^{\nu_L} f_I(x_1,\ldots,x_L;t_1,\ldots,t_L)dx_1\ldots dx_L,$$

$$\mu_{Sx\nu_1...\nu_L}(t_1,...,t_L) = \mathrm{M}_S[(X(t_1)-m_{Sx}(t_1))^{\nu_1}...(X(t_L)-m_{Sx}(t_L))^{\nu_L}],$$

$$\mu_{Ix\nu_1...\nu_L}(t_1,...,t_L) = \mathrm{M}_I[(X(t_1)-m_{Ix}(t_1))^{\nu_1}...(X(t_L)-m_{Ix}(t_L))^{\nu_L}],$$

$$m_{sx}(t) = \mathrm{M}_s[X(t)],\ m_{ix}(t) = \mathrm{M}_i[X(t)],$$

$$m_{sx\nu_1...\nu_L}(t_1,...,t_L) = \mathrm{M}_s[X^{\nu_1}(t_1)...X^{\nu_L}(t_L)], \quad m_{ix\nu_1...\nu_L}(t_1,...,t_L) = \mathrm{M}_i[X^{\nu_1}(t_1)...X^{\nu_L}(t_L)],$$

$$\mu_{sx\nu_1...\nu_L}(t_1,...,t_L) = \mathrm{M}_s[(X(t_1)-m_{x/g}(t_1))^{\nu_1}...(X(t_L)-m_{x/g}(t_L))^{\nu_L}],$$

$$\mu_{ix\nu_1...\nu_L}(t_1,...,t_L) = \mathrm{M}_i[(X(t_1)-m_{x/g}(t_1))^{\nu_1}...(X(t_L)-m_{x/g}(t_L))^{\nu_L}].$$

The bound's correlation functions and the bound's covariance functions are

$$K_{Sx}(t_1,t_2) = \mathrm{M}_S[X(t_1)X(t_2)], \quad K_{Ix}(t_1,t_2) = \mathrm{M}_I[X(t_1)X(t_2)],$$

$$R_{Sx}(t_1,t_2) = \mathrm{M}_S[(X(t_1)-m_{Sx}(t_1))(X(t_2)-m_{Sx}(t_2))],$$

$$R_{Ix}(t_1,t_2) = \mathrm{M}_I[(X(t_1)-m_{Ix}(t_1))(X(t_2)-m_{Ix}(t_2))]$$

and the correlation function's bounds and covariance function's bounds are

$$K_{sx}(t_1,t_2) = \mathrm{M}_s[X(t_1)X(t_2)], \quad K_{ix}(t_1,t_2) = \mathrm{M}_i[X(t_1)X(t_2)],$$

$$R_{sx}(t_1,t_2) = \mathrm{M}_s[(X(t_1)-m_{x/g}(t_1))(X(t_2)-m_{x/g}(t_2))],$$

$$R_{ix}(t_1,t_2) = \mathrm{M}_i[(X(t_1)-m_{x/g}(t_1))(X(t_2)-m_{x/g}(t_2))].$$

Stationary and Ergodic Hyper-random Functions

It has been found that some hyper-random functions have special stationary and ergodic properties. A function $X(t)$ has been called a stationary hyper-random one if the bound's mean do not depend from time and bound's correlation functions depend only from time interval $\tau = t_2 - t_1$: $K_{Sx}(t_1,t_2) = K_{Sx}(\tau)$, $K_{Ix}(t_1,t_2) = K_{Ix}(\tau)$.

A function $X(t)$ has been called stationary hyper-random one for all conditions if the mean $m_{x/g}(t) = \int_{-\infty}^{\infty} x f(x;t/g)\mathrm{d}x$ does not depend from time t ($m_{x/g}(t) = m_{x/g}$) and the correlation function

$$K_{x/g}(t_1,t_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 f(x_1,x_2;t_1,t_2/g)\mathrm{d}x_1\mathrm{d}x_2$$

depends only from the interval τ and the condition g: $K_{x/g}(t_1,t_2) = K_{x/g}(\tau)$.

The bound's correlation functions $K_{Sx}(\tau)$, $K_{Ix}(\tau)$ are determined by bound's spectral density $S_{Sxx}(f)$, $S_{Ixx}(f)$ that linked each other by the following expressions:

$$S_{Sxx}(f) = \int_{-\infty}^{\infty} K_{Sx}(\tau)\exp(-\mathrm{j}2\pi f\tau)\mathrm{d}\tau,\ S_{Ixx}(f) = \int_{-\infty}^{\infty} K_{Ix}(\tau)\exp(-\mathrm{j}2\pi f\tau)\mathrm{d}\tau,$$

$$K_{Sx}(\tau)=\int_{-\infty}^{\infty} S_{Sxx}(f)\exp(\mathrm{j}2\pi f\tau)\mathrm{d}f,\ K_{Ix}(\tau)=\int_{-\infty}^{\infty} S_{Ixx}(f)\exp(\mathrm{j}2\pi f\tau)\mathrm{d}f,$$

where f is a frequency.

The spectral density's bounds are determined by expressions $S_{sxx}(f)=\sup_{g\in G} S_{xx/g}(f)$, $S_{ixx}(f)=\inf_{g\in G} S_{xx/g}(f)$, where $S_{xx/g}(f)$ is the spectral density for condition g:

$$S_{xx/g}(f)=\int_{-\infty}^{\infty} K_{x/g}(\tau)\exp(-\mathrm{j}2\pi f\tau)\mathrm{d}\tau,\ K_{x/g}(\tau)=\int_{-\infty}^{\infty} S_{xx/g}(f)\exp(\mathrm{j}2\pi f\tau)\mathrm{d}f.$$

For two hyper-random functions $X(t)$, $Y(t)$ stationary linked each other the bound's correlation functions are determined by the following expressions:

$$K_{Sxy}(\tau)=\int_{-\infty}^{\infty} \dot{S}_{Sxy}(f)\exp(\mathrm{j}2\pi f\tau)\mathrm{d}f,\ K_{Ixy}(\tau)=\int_{-\infty}^{\infty} \dot{S}_{Ixy}(f)\exp(\mathrm{j}2\pi f\tau)\mathrm{d}f,$$

where $\dot{S}_{Sxy}(f)$, $\dot{S}_{Ixy}(f)$ are the bound's spectral density: $\dot{S}_{Sxy}(f)=\int_{-\infty}^{\infty} K_{Sxy}(\tau)\exp(-\mathrm{j}2\pi f\tau)\mathrm{d}\tau$, $\dot{S}_{Ixy}(f)=\int_{-\infty}^{\infty} K_{Ixy}(\tau)\exp(-\mathrm{j}2\pi f\tau)\mathrm{d}\tau$.

The spectral density's bounds are $\dot{S}_{sxy}(f)=\sup_{g\in G}\dot{S}_{xy/g}(f)$, $\dot{S}_{ixy}(f)=\inf_{g\in G}\dot{S}_{xy/g}(f)$, where $\dot{S}_{xy/g}(f)$ is the spectral density for condition g: $\dot{S}_{xy/g}(f)=\int_{-\infty}^{\infty} K_{xy/g}(\tau)\exp(-\mathrm{j}2\pi f\tau)\mathrm{d}\tau$, $K_{xy/g}(\tau)=\int_{-\infty}^{\infty}\dot{S}_{xy/g}(f)\exp(\mathrm{j}2\pi f\tau)\mathrm{d}f$.

It has been determined the particularities of these characteristics and introduced a number of new conceptions, in particular hyper-random white noise.

Some hyper-random function $X(t)$ may be presented as a set of the random functions determined on the disjoint intervals $T_g=[Tg,T(g+1))$ with longitude T on that the conditions are not changed ($g=0,\pm1,\pm2,\ldots$). Let $X_g(t)$ is the part of the function $X(t)$ according to interval T_g and reduced to interval $[-T/2,T/2)$:

$$X_g(t-T(g+0,5))=\begin{cases}X(t), & \text{if} \quad t\in T_g,\\ 0, & \text{if} \quad t\notin T_g.\end{cases}$$

The function $X_g(t)$ in a fixed condition $g=0,\pm1,\pm2,\ldots$ is the random function determined on the interval $t\in[-T/2,T/2)$. The set of these functions in uncertainty conditions is a hyper-random function $Y(t)=\{X_g(t),\quad g=0,\pm1,\ldots\}$. A hyper-random function is any function $\varphi(Y(t_1),\ldots,Y(t_L))$ too, where $t_1,\ldots,t_L\in[-T/2,T/2)$.

A hyper-random function $X(t)$, that is stationary for all conditions and $\lim_{T\to\infty}\overline{m}_{\varphi}(T)=m_{\varphi}$, has been called an ergodic one. Here $\overline{m}_{\varphi}(T)$ is the sample mean:

$$\overline{m}_{\varphi}(T)=\overline{\mathrm{M}}_T[\varphi(Y(t_1),\ldots,Y(t_L))]=\frac{1}{T}\int_{-T/2}^{T/2}\varphi(Y(t_1+t),\ldots,Y(t_L+t))\mathrm{d}t$$

and $m_{\varphi} = \mathrm{M}[\varphi(Y(t_1),...,Y(t_L))]$ is the mean of the function $\varphi(Y(t_1),...,Y(t_L))$.

A hyper-random ergodic function $X(t)$ may be presented by the following serious: $X(t) = \lim_{T\to\infty} \sum_{g} X_g(t - T(g + 0,5))$.

When $T \to \infty$ the mean's bounds and the correlation and the covariance function's bounds are described by the following expressions:

$$\overline{m}_{sx_T} = \sup_{g\in G} \frac{1}{T} \int_{T_g} x_g(t)\, \mathrm{d}t\,,\ \overline{m}_{ix_T} = \inf_{g\in G} \frac{1}{T} \int_{T_g} x_g(t)\, \mathrm{d}t,$$

$$\overline{K}_{sx_T}(\tau) = \sup_{g\in G} \frac{1}{T} \int_{T_g} x_g(t+\tau)\, x_g(t) \mathrm{d}t\,,\ \overline{K}_{ix_T}(\tau) = \inf_{g\in G} \frac{1}{T} \int_{T_g} x_g(t+\tau)\, x_g(t) \mathrm{d}t\,,$$

$$\overline{R}_{sx_T}(\tau) = \sup_{g} \frac{1}{T} \int_{T_g} [x_g(t+\tau) - \overline{m}_{x_T/g}][x_g(t) - \overline{m}_{x_T/g}] \mathrm{d}t,$$

$$\overline{R}_{ix_T}(\tau) = \inf_{g} \frac{1}{T} \int_{T_g} [x_g(t+\tau) - \overline{m}_{x_T/g}][x_g(t) - \overline{m}_{x_T/g}] \mathrm{d}t,$$

where $\overline{m}_{x_T/g} = \frac{1}{T} \int_{T_g} x_g(t)\, \mathrm{d}t$.

Hyper-random Models

Developed approaches give possibilities to model different types of real physical objects and their estimates under uncertainty changing of object's properties and statistical observation conditions. It has been proposed different measure models: determine – hyper-random, random – hyper-random, and hyper-random – hyper-random ones, in that the objects are presented by determine, random, and hyper-random models and their estimates – by hyper-random models.

In case of determine – hyper-random measure model, in the fixed condition g the accuracy of vector estimation $\vec{\Theta}^*$ of parameter $\vec{\theta}$ may be described by the expectation of error's square $\Delta_g^2 = \mathrm{M}[\left|\vec{\Theta}^* - \vec{\theta}\right|^2 / \vec{\theta}, g]$, where M is expectation operator. For the indefinite condition the accuracy is characterized by the interval where the value Δ_g^2 may be situated. The bounds of this interval are $\Delta^2_{\min} = \min[\Delta_S^2, \Delta_I^2]$, $\Delta^2_{\max} = \max[\Delta_S^2, \Delta_I^2]$ where $\Delta_S^2 = \mathrm{M}_S[\left|\vec{\Theta}^* - \vec{\theta}\right|^2 / \vec{\theta}]$, $\Delta_I^2 = \mathrm{M}_I[\left|\vec{\Theta}^* - \vec{\theta}\right|^2 / \vec{\theta}]$ are the bound's quadratic estimate.

The accuracy of point estimation may be characterized by bounds of quadratic estimate:

$$\Delta_s^2 = \sup_{g\in G} \mathrm{M}[\left|\vec{\Theta}^* - \vec{\theta}\right|^2 / \vec{\theta}, g], \quad \Delta_i^2 = \inf_{g\in G} \mathrm{M}[\left|\vec{\Theta}^* - \vec{\theta}\right|^2 / \vec{\theta}, g].$$

In scalar case the volumes Δ_S^2, Δ_I^2 and Δ_s^2, Δ_i^2 may be presented as $\Delta_S^2 = \sigma_S^2 + \varepsilon_{S0}^2$, $\Delta_I^2 = \sigma_I^2 + \varepsilon_{I0}^2$ and $\Delta_s^2 = \sup_{g\in G}[\sigma_g^2 + \varepsilon_{0/g}^2]$, $\Delta_i^2 = \inf_{g\in G}[\sigma_g^2 + \varepsilon_{0/g}^2]$ where $\sigma_S^2 = \mathrm{M}_S\left[\left(\Theta^* - m_S\right)^2 / \theta\right]$, $\sigma_I^2 = \mathrm{M}_I\left[\left(\Theta^* - m_I\right)^2 / \theta\right]$ are the variances of error bounds, $\sigma_g^2 = \mathrm{M}\left[\left(\Theta^* - m_{\theta^*/g}\right)^2 / \theta, g\right]$ is the error variances for condition g, $\varepsilon_{S0} = (m_S - \theta)$, $\varepsilon_{I0} = (m_I - \theta)$ are the systematic errors for estimation distribution bounds, and $\varepsilon_{0/g} = (m_{\theta^*/g} - \theta)$ is the systematic error for condition g (fig. 2).

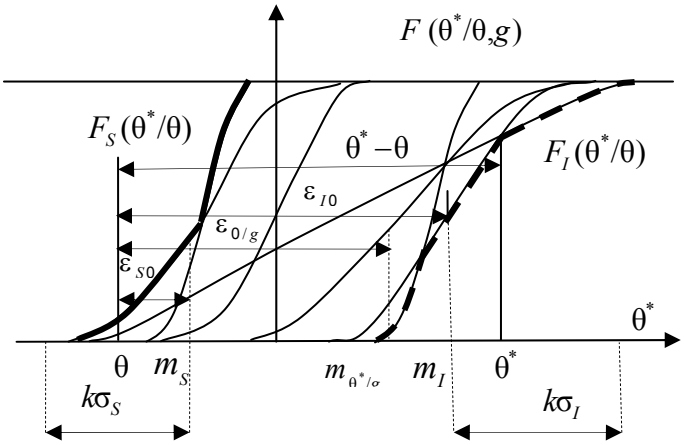


Fig 2. The fan of distribution functions $F(\theta^*/\theta,g)$ (thin curves) for different conditions g and supremum $F_S(\theta^*/\theta)$ and the infimum $F_I(\theta^*/\theta)$ bounds of the distribution function (bold curves)

To characterize the error $\Delta=\Theta^*-\theta$ of scalar parameter θ and it's estimate Θ^* the intervals $[\varepsilon_{S0}-k\sigma_S,\ \varepsilon_{I0}+k\sigma_I]$ and $[m_S-k\sigma_S,\ m_I+k\sigma_I]$ may be used correspondently, where σ_S, σ_I are the bound's error standard deviation, and k is the constant (fig. 2). If conditional distributions of random values $\Theta^*/\theta,g$ are not penetrates and the variance $D_{x/g}$ is raised or reduced with rising the mean $m_{x/g}$, the last interval is determined by error mean's bounds m_s, m_i and error standard deviation's bounds σ_s, σ_i. For "a" type distribution it may be presented as $[m_i-k\sigma_i,\ \ m_s+k\sigma_s]$ and for "b" type distribution – as $[m_i-k\sigma_s,\ \ m_s+k\sigma_i]$.

A hyper-random estimate $\vec{\Theta}^*$ of fixed parameter $\vec{\theta}$ was called consistent one if it converged in probability to this parameter under all conditions $g\in G$: $\lim\limits_{N\to\infty}P\{\left|\vec{\Theta}^*-\vec{\theta}\right|>\varepsilon/\vec{\theta},g\}=0 \quad \forall g\in G$, where N is a sample size for every condition g and $\varepsilon>0$.

The necessary condition, that the hyper-random estimate is a consistent type, is that it degenerates to random estimate when $N\to\infty$. So, estimates are not consistent if they stay hyper-random type when $N\to\infty$.

It was made a hypothesis (hyper-random hypothesis) that all real physical phenomena are existed in continuously changed statistic conditions and therefore all physical phenomena, usually considered as a random type, really are the hyper-random type. This particularity exists not only in case of finite but infinite interval observation. It is followed from this that all real estimates are not consistent and it is impossible to achieve infinitely large accuracy in any conditions.

The bounds of error's square expectation Δ_s^2, Δ_i^2 formed on the base of sample $\vec{X}$ size N and bounds $\mathrm{D}_s\lfloor\Theta^*/\theta\rfloor$, $\mathrm{D}_i\lfloor\Theta^*/\theta\rfloor$ of estimate's variance $\mathrm{D}\lfloor\Theta^*/\theta,g\rfloor$ are described by the inequalities

$$\Delta_s^2\ge\mathrm{D}_s\lfloor\Theta^*/\theta\rfloor\ge\sup_{g\in G}\left[\left(1+\frac{\partial\varepsilon_{0/g}}{\partial\theta}\right)^2J_{N/g}^{-1}\right],\qquad \Delta_i^2\ge\mathrm{D}_i\lfloor\Theta^*/\theta\rfloor\ge\inf_{g\in G}\left[\left(1+\frac{\partial\varepsilon_{0/g}}{\partial\theta}\right)^2J_{N/g}^{-1}\right],$$

where $J_{N/g}$ – Fisher intrinsic accuracy for random value $\Theta^*/\theta,g$:

$$J_{N/g}=\mathrm{M}\left[\left(\frac{\partial\ln f_N(\vec{X}/\theta,g)}{\partial\theta}\right)^2\right]=-\mathrm{M}\left[\frac{\partial^2\ln f_N(\vec{X}/\theta,g)}{\partial\theta^2}\right],$$

$f_N(\vec{x}/\theta, g)$ – probability density function of sample $\vec{X}/\theta, g$.

The bound's quadratic estimate Δ_S^2, Δ_I^2 and the bound's variance $\mathrm{D}_S\left\lfloor \Theta^*/\theta \right\rfloor$, $\mathrm{D}_I\left\lfloor \Theta^*/\theta \right\rfloor$ are defined by inequalities

$$\Delta_S^2 \geq \mathrm{D}_S\left\lfloor \Theta^*/\theta \right\rfloor \geq \frac{\left(1+\frac{\partial \varepsilon_{S0}}{\partial \theta}\right)^2}{\mathrm{M}_S\left[\left(\frac{\partial \ln f_{SN}(\vec{X}/\theta)}{\partial \theta}\right)^2\right]}, \qquad \Delta_I^2 \geq \mathrm{D}_I\left\lfloor \Theta^*/\theta \right\rfloor \geq \frac{\left(1+\frac{\partial \varepsilon_{I0}}{\partial \theta}\right)^2}{\mathrm{M}_I\left[\left(\frac{\partial \ln f_{IN}(\vec{X}/\theta)}{\partial \theta}\right)^2\right]},$$

were $f_{SN}(\vec{x}/\theta)$, $f_{IN}(\vec{x}/\theta)$ are bound's probability density function of sample $\vec{X}/\theta$.

Analogues results were obtained for hyper-random – hyper-random measure model too.

On the base of hyper-random hypothesis was shown that in any case accuracy of any real physical measurements is limited, all real estimates are not consistent ones, and therefore all real physical phenomena are hyper-random type.

Processing of Hyper-random Signals

Developed body of mathematics may be effectively used for signal processing. The example illustrating such possibilities presents below.

Let us look the measure process of the level noise in the production area when there is a lot of production equipment which time to time switch on and switch off and therefore the noise condition is changed in widely boundaries. The measurement is done on the basis of the data obtained for a long time.

This task may be concretized by different manner. If the noise in fixed condition and the rule of changing condition may be regarded as random processes, the task becomes a classic one that consists of estimation of a random variable or some random variables. To solve this task it is requested to know the distribution functions type or at least have information that such distributions exist.

If it is impossible to propose, the changing conditions may be described by any distribution, the task is a hyper-random type. In this case, the recorded data is a sample from a general population of the hyper-random function $X(t)$.

By the processing of this data it is possible to obtain estimates of different characteristics. The image of the recorded data and estimates of some characteristics give the fig. 3 – 4. It has been proposed that the process is an ergodic type.

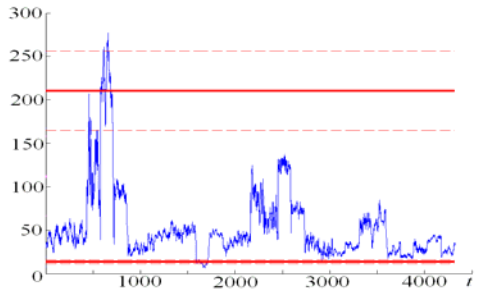


Fig. 3. Current noise level in the production area (solid line), estimates of the bounds of the expected values m_{sx}^*, m_{ix}^* (straight solid bold lines), and the bounds of the standard deviations $m_{sx}^* \pm \sigma_{sx}^*$, $m_{ix}^* \pm \sigma_{ix}^*$ (dashed lines).

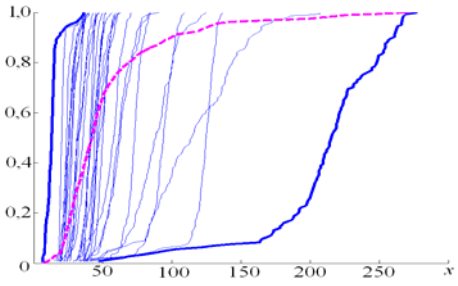


Fig. 4. The estimates of the distribution functions $F^*(x/g)$ (solid lines), the estimates of the bounds of the distribution function $F_S^*(x)$, $F_I^*(x)$ (solid bold lines), and the estimate of the distribution function $F^*(x)$ calculated in the hypothesis that the data are random type (bold dashed line).

It is followed from the figures that presented parameters and functions give a lot of useful information that is essentially more informative than characteristics usually used for describing of random processes.

Conclusion

1. Any hyper-random phenomena (events, variables, functions) may be regarded as a set (family) of random subsets. In this construction, every subset is associated with any fixed observation condition. The probability measures are determined for the elements of each subset; however, the measures are not determined for the subsets of the set.

2. Hyper-random variables and functions may be described by the supremum and the infimum bounds of the distribution function. Among the bounds of the distribution function there is a zone of the ambiguity. Random and chaotic phenomena are the degenerate hyper-random phenomena.

3. In addition to the bounds of the distribution function, the main characteristics describe hyper-random variables and functions are bound's crude and the central moments and also crude and the central moment's bounds. They are, in particular, bound's mean, bound's variance and also mean's bounds variance's bounds and so on.

4. Estimations of hyper-random variables and functions were researched. It was paid attention to all real statistical conditions were continuously changed. Therefore all real physical phenomena usually regard as random tapes, in really, are hyper-random tapes. This particularity occurs not only in case of finite but also in case of infinite interval observations. It is follows from this that all estimations of real variables and functions are not consistent and so it is impossible to achieve infinite physical measurement accuracy in any real conditions.

Bibliography

[1] J.B. Keller, "The probability of heads," *Am. Math. Monthly*, vol. 93. P. 191, 1986.

[2] Jakob Bernoulli, "The art of conjecturing," 1713.

[3] R. von Mises, "Mathematical theory of probability and statistics," Edited and complemented by H. Geiringer. N.Y. and London, Acad. Press, 1964.

[4] Yu I Alimov, Yu A Kravtsov, "Is probability a "normal" physical quantity?" *Physics-Uspekhi,* vol. 35, no 7, pp. 606-622, 1992.

[5] L.A. Zadeh and J. Kacprzyk (Eds.), "Fuzzy logic for the management of uncertainty," John Wiley & Sons, New York, 1992.

[6] M.T. Hagan, H.B. Demuth, and M.H. Beale, "Neural network design," Boston, MA: PWS Publishing, 1996.

[7] R.M. Crownover, "Introduction to fractals and chaos," Jones and Bartlett Pub., Inc., Boston – London, 1995.

[8] I.I. Gorban "Random, hyper-random, chaos, and uncertainty," *Стандартизація, сертифікація, якість,*. no. 3, pp. 41 – 48, 2005.

[9] I.I. Gorban "Hyper-random phenomena and their description," *Акустичний вісник,* vol. 8, no 1 – 2, pp. 16 – 27, 2005.

[10] I.I. Gorban "Hyper-random functions and their description," Радиоэлектроника (Radioelectronics and Communications Systems), no 1, pp. 3 – 15, 2006.

[11] I.I. Gorban "Description of hyper-random variables and functions," Акустичний вісник. vol. 8, no 3, pp. 24 – 33, 2005.

[12] I.I. Gorban "Stationary and ergodic hyper-random functions," Радиоэлектроника (Radioelectronics and Communications Systems), no 2, pp. 54 – 70, 2006.

[13] I.I. Gorban "Estimation of hyper-random variables," Математические машины и системы, no 1, pp. 40 – 48, 2006.

[14] I.I. Gorban "Estimation methods for hyper-random variables," Математические машины и системы, no 2, pp. 3 – 14, 2006.

[15] I.I. Gorban "Imaging physical phenomena by hyper-random models," Математические машины и системы, no 1, pp. 34 – 41, 2007.

[16] I.I. Gorban "Theory of hyper-random phenomena," Kiev: National Academy of Sciences, 2007.

Author's Information

***Igor Gorban**, Deputy Director General in Science, The State Enterprise "Ukrainian Scientific-Research and Training Center", Ph.D.,Dr.Sc., Prof., Kyiv, Ukraine, e-mail:* gorban@ukrndnc.org.ua

TWO FUNDAMENTAL PROBLEMS CONNECTED WITH AI [1]

Dimiter Dobrev

***Abstract**: This paper is about two fundamental problems in the field of computer science. Solving these two problems is important because it has to do with the creation of Artificial Intelligence. In fact, these two problems are not very famous because they have not many applications outside the field of Artificial Intelligence.*

In this paper we will give a solution neither of the first nor of the second problem. Our goal will be to formulate these two problems and to give some ideas for their solution.

***Keywords**: AI Definition, Artificial Intelligence.*

***ACM Classification Keywords**: I.2.0 Artificial Intelligence - Philosophical foundations*

Introduction

Since year 2000 we have a definition of AI [1,2,3] and since 2005 we have a program which satisfies this definition [4, 5]. Actually, these two facts are not very popular, first because the definition of AI is no accepted from almost no one except its author and second because the program which satisfies the definition of AI is useless from the practical point of view due to the combinatorial explosion.

From theoretical point of view we divide the programs in two types. The first are the non-terminating programs which will work infinitely long and the second are the terminating programs which will stop after a finite number of steps. On the other hand, from practical point of view, we divide the programs in ones that work in real time and ones which cannot work in real time. So, the fact that one program is a terminating one is useless for practical purposes if this program will work practically for infinitely long time.

That is why the program which is described in [4, 5] has no use for practical purposes and no one recognises it as AI because it does not satisfy the major requirement which is to work in real time. Even the program from [4, 5] is represented only as an algorithm. It is not written as a program because it is useless to write a program which will terminate after the end of the universe.

Therefore, if we want to make a program which will be recognized as AI we have to correct the algorithm from [4, 5] and make it work in real time. Here we have to deal with the problem of the combinatorial explosion. Even in this case the term "combinatorial explosion" is not very proper because we use this term for the cases when a programmer writes a program which should work in real time but, actually, is not working. Also, we usually assume that when we have a combinatorial explosion a faster computer can eventually help us solve the problem. In this case the situation is different. We have an algorithm which is not designed to work in real time. There is not any attempt to make the algorithm faster. The main priority has been to make the description short and clear without taking into consideration the efficiency because it is obvious that this algorithm has only theoretical value and that it will never work as a real program.

Example with the perfect compression program

So, our task is to make a real program from one algorithm which is not designed to work in real time. Actually, the algorithm in [4, 5] describes the perfect AI but we need a working AI, which does not need to be perfect.

We have a similar problem with the perfect compression algorithm and real compression programs. Let us define the perfect compression algorithm in order to see how little the connection between it and the real compression programs is.

Here perfect compression algorithm is called the algorithm which enumerates all programs and returns the first one (i.e. the shortest one) which generates the string which has to be compressed.

[1] This publication is partially supported by the Bulgarian Ministry of Education (contract БОЕ 4-02/2004 г.)

There are two things to note here. First, we have to mention that this algorithm is a non-terminating one due to the undecidability of the halting problem. In order to make it a terminating one we have to add a requirement for efficiency of the program which we search for. We can say: "the first one which generates the string for no more than ***N*** steps" but we do not want to include an additional parameter ***N*** in the definition. That is why we will say: "the program which generates the string and which has the minimal sum between its length and the number of steps which it makes while generating the string". With such correction we will obtain a compression algorithm which is a terminating one from the theoretical point of view. (Anyway, this algorithm is non-terminating in practice and therefore it is useless.)

On second place, this algorithm generates the perfect self-extracting compression file but if we assume that we have a decompression program then a shorter data file may exist, which will return our string if we input this data in the decompression program. This means that here we are talking only of self-extracting compressions.

So, we have the perfect compression algorithm. We do not say the perfect compression program because no one wrote this algorithm as a program because this is useless work. The description of this algorithm can be obtained directly from the definition of Kolmogov's complexity [10]. This means that we can say that Kolmogov is the author of the first compression algorithm but maybe this is not correct because this algorithm cannot work in real time. Today we have many programs which make compression (including self-extractable compression). These programs are not perfect but they can work in real time. Actually, these programs are much more complicated than the perfect compression algorithm and you cannot construct them directly from the perfect compressor because they are based on totally different principles.

The situation is similar with the perfect AI and the real AI. We have the perfect AI but we cannot extract a real AI which will be able to work in real time directly from it. This comes to show how difficult our task to make a real AI is.

Dividing the problem in two parts

In order to construct a real AI we will divide its work in two parts. The first part is to find a good model of the world and the second part is to choose the best action on the basis of the selected model.

Actually, in the perfect AI these two parts are not separated. We will remind that the perfect AI from [4, 5] works by trying all possible strategies in all possible models and chooses the best strategy with the biggest average result (the average result is calculated on the basis of all possible models). So, the perfect AI solves these two tasks jointly, without separating them. Nevertheless, the separation of this two problems is natural and we will make it.

If we have real time solution of both these problems then we will have a real AI. Unfortunately, both these problems lead us to a combinatorial explosion. These two problems are not very famous because they do not have many applications outside the field of the Artificial Intelligence.

We will start with the second problem which is more famous and better studied.

Finding of the correct action on the basis of a given model

We have an algorithm for solving of this problem. The name of this algorithm is **Min-Max** and we use it with great success in Chess playing programs. Nevertheless, this algorithm is not proper in all cases because sometimes it gives a combinatorial explosion. Actually, it gives combinatorial explosion even with chess but in this game we can go around the combinatorial explosion by limiting the depth in which we examine the tree of the game. This is possible with the game of chess because we can make good evaluation of the position on the basis of things like the number of pieces on the board and on the "territory" which these pieces cover. Therefore, in some cases this problem is solvable in real time but not in all cases.

A famous example is the problem how to make a program which can play the Go game well enough to beat a professional player. A price of one million dollars was offered for working out this problem [11]. Unfortunately, the prize was not taken because the problem is too complex. The Go game looks like the chess but in it you cannot apply the **Min-Max** algorithm directly because you do not have a good evaluation function for the positions. The problem is that we have too many possible moves and mostly because in the Go game after many moves nothing essential happens (nothing which can be easily detected by a simple evaluation function).

As we said at the beginning, we will not give a solution to this problem. This is not because the One Million Dollar Prize has already expired but because we do not know how to solve this problem. Anyway, we will give some ideas. The main idea is to define intermediate goals and large steps. Actually, intermediate goal is used in the chess playing programs where this goal is to increase the value of the position. Unfortunately, this intermediate goal is given by the programmer but for AI this goal should be generated automatically because AI cannot depend on a programmer to say what is right to be done in each case.

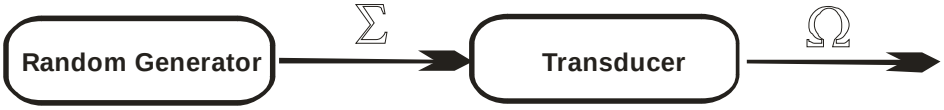
What is to think in large steps? This means to plan a chain of intermediate goals which leads to the main goal. Here we will say "goal" for events which we evaluate as good ones. One event can be evaluated as a good one by a priory or because it is part of a chain which leads to an event which has already been evaluated as a good one. So, thinking by large steps will be planning chains of events. For such planning we can use the **Min-Max** algorithm but here the problem is how to define events automatically and how to automatically find the way for transition from one event to another. For example, with the game of chess you have events "taking of enemy piece" and "winning the game". There is a connection between these two events and this connection is built in the chess playing programs by their creators. So, the chess playing program tries to take enemy pieces in order to win the game. The problem is how to make a program which defines events automatically and automatically evaluates these events as good or bad. Also, AI has to be able to automatically find connections between these events in order to plan a chain of events.

Actually, all these thoughts lead us to the fact that in order to solve the second problem we need a solution of the first one because in order to think in large steps we need an automatic detection of events and this is part of the problem of finding a good model of the world. From this point on we will talk only about the solution of the first problem.

Formalization of the first problem

Here when we say a good model we mean an adequate one. So, this means a model which will give correct predictions for the future.

The first step in solving a problem is to formalize it. Let us examine the following formal problem. Let us have a two finite alphabets Σ and Ω. Let us have a random generator which generates letters from the alphabet Σ and a transducer which inputs a letter from Σ and outputs a letter from Ω. The goal is to built a model of this transducer which will give us the possibility to guess its next output if we know the input letter and if we know the entire history (i.e. if we know the row $a_0, b_0, a_1, b_1, ..., a_{n-1}, b_{n-1}, a_n$ where a_i are the letters from the random device which are inputted in the transducer and b_i are the letters which are outputted.) So, the question is what will be the output on the step n if we know all data from step **0** to step $n-1$ and the input on the step n. In other words, what will be b_n if we know $a_0, b_0, a_1, b_1, ..., a_{n-1}, b_{n-1}, a_n$.



What is the connection between this formalization and the definition of AI [1,2,3]? Here we have a random generator and transducer, which interact. In the definition of AI the transducer corresponds to the concept of World. Here we try to make a model of the transducer but in [1,2,3] AI tries to understand the World. This means that here the random generator corresponds to the AI from [1,2,3]. Where is the difference? AI reads the output of the World (of the transducer) but the random generator does not have any input. AI is able to carry out some experiments in order to understand the World but the random generator does not make any intentional experiments. Anyway, if the observer waits long enough, the random generator will make all experiments (accidentally).

Note: In [1,2,3] the alphabets Σ and Ω are Ω and Σ and the letters a and b are d and v. This can cause confusion in understanding the connection between [1,2,3] and this paper.

How to find a good model of the world (transducer)

So, we have a formalization and now our problem is formal. As we said this problem is not famous because it has not many applications outside the field of AI. It is even difficult to find an example for a practical problem which

leads to this theoretic formalization. The only such example which we have in mind is the following. Let us have a program protected against illegal use by a hardware key device. If we want to break this protection we have to understand how this hardware key works and try to recreate it. Really, the practical problem allows us to open the key and see how it is designed but here we assume that we have no such possibility and that we have to observe the key as a black box and to study only its input and output.

As you see, there are not many applications of this problem. Maybe this is the reason that nobody offers a price for its solution but nevertheless, here we will discuss this problem.

So, is this problem solvable? In the general case the answer is no because if we do not make any suggestions about the transducer then we will not be able to say anything about its next output. For example, if it outputs one and the same letter one hundred times in a row regardless of the input then we can predict that on the next step it will work out the same letter. This prediction looks natural but it lies on the conjecture that the simpler explanation is more probable than the complicated one. Without this conjecture we cannot make any prediction because it is possible that in this case we have a transducer which outputs one hundred times one and the same letter and on the next step it outputs another letter.

We said that we will look for the simplest model of the transducer. Also, we have to bear in mind that we need a solution which works in real time.

Another question is whether our transducer is deterministic device or not. It will be much easier if we assume that the transducer is a deterministic device but if we restrict our search only in the set of deterministic models then the chance to find a proper model in a concrete situation is very small.

Next question: How many internal states our transducer has? It is reasonable to suggest that the number of internal states is finite (i.e. that it is finite automata). Anyway, the more general case is to suggest countably many internal states. It is no use suggesting an uncountable number for the internal states because only a countable subset of them will be obtainable in the deterministic case. In the non-deterministic case there is some use in suggesting an uncountable number of internal states but if we restrict our observation to the set of calculable functions then again there is no use suggesting an uncountable number for the internal states of the transducer.

The last question: Is our transducer a calculable function or not. Definitely yes. We are looking for a practical solution so it has to be a calculable function and even it has to be an easy calculable function (i.e. calculable for small number of steps without problems like combinatorial explosion). Besides that, every non-calculable function can be approximated with a calculable one (of course, until the concrete moment n but not until the infinity).

One theoretical solution

Here we will give the next useless theoretical solution which cannot work in real time. The reason that we give this solution is to show that such one exists. This is important because we cannot give a solution which can work in real time. Instead of that at the end of this paper we will give some ideas about the creation of real time solution.

Here is our theoretical solution. First for the deterministic case:

It will enumerate all programs and will return the first one (i.e. the shortest one) which generates the row $b_0, ..., b_{n-1}$ if the input is $a_0, ..., a_{n-1}$. Here we have a problem with undecidability of the halting problem again. So, we will take not the shortest one but this which has minimal sum between its length and the maximum number of program steps which it needs to generate any of the outputs (i.e. any of $b_0, ..., b_{n-1}$). So, this algorithm will give us a short and quick program which makes a very good prediction of b_n. The only problem is that we will have to wait this algorithm to finish almost forever.

For the non-deterministic case we have to complicate our algorithm a little bit.

First we will complicate our programs (which we use as models) by adding one subroutine **random( )** which will return zero or one with possibility **1/2**. With this subroutine we cannot generate even the possibility **1/3** but by using subroutine **random( )** we can approximate any possibility (nevertheless is it rational or irrational number).

Now, when we deal with non-deterministic models we cannot say simply yes or no to the question does this model generate our sequence or not. Instead of that, we can calculate the possibility for our sequence to be generated. Of course, here we will have the problem with the non-terminating models again and in order to keep

things calculable we will add one constant ***Max*** and we will calculate the possibility of the model to generate our sequence for no more than ***Max*** steps per output.

What is the prediction of one non-deterministic model for b_n. First we do not know what is the internal state of the model when it inputs a_n because there may be more than one possible way for this model to generate $b_0, ..., b_{n-1}$. Even if we know the internal state we cannot say which letter will be worked out as b_n because our model is non-deterministic. Nevertheless, we can calculate for concrete model the possibility for every letter to be worked out.

Every model will give us some prediction but we have to choose which one to trust and which prediction to accept as the better one. This question will not be discussed in this paper.

Some ideas about the practical solution

First, in order to make real time solution we will restrict the observation to the set of models with a finite number of states (finite automata). Of course, this restriction is essential because some of the worlds (transducers) cannot be described with finite models. Anyway, in many cases the finite models are sufficient or at least they can give a good approximation of the World. (You can find in [7] the idea that we can raise the finite models with first order axioms in order to make models for more complex worlds.)

Second, we have to mention that we will look for a set of good models instead of a single model. The chance to find a single model which describes the world is small. It is more probably to find many different simple models which describe different features of the world. Also, in this way our system will be more consistent because in its life (work time) it will change some of the selected models instead of changing the only model which can make its behavior totally different.

Now, let us start with the case of deterministic models. Such model looks like a deterministic finite automata (with finite number of states, starting state, arcs labeled with the letters from Σ, etc.) but here we will have only one type of states (no final states) and we will have a second label on every arc which will be a letter from Ω.

If we have such a model with a reasonable number of states we can easily find it by a backtracking algorithm similar to the one from [8]. Anyway, the existence of such model is very suspicious because if we have deterministic model then we will be able 100% correctly to predict the future. This will mean that the world is very simple, which is not the interesting case.

Let us look for a non-deterministic model of the world. Actually, as we said, we will look for a set of many non-deterministic models.

We will divide the non-deterministic models in two groups - partially deterministic and totally non-deterministic. Examples of these two types of non-deterministic models are found in [6, 7].

Partially deterministic models will look like the deterministic models but with the difference that they will have a second label on the arcs, which is a set of letters from Ω instead of one letter. Actually, this will be a set of 2-tuples from letter and possibility because every letter from Ω will have its own possibility to be worked out in the case when the model is in the respective state and the input letter from Σ is that which is the first label of the arc.

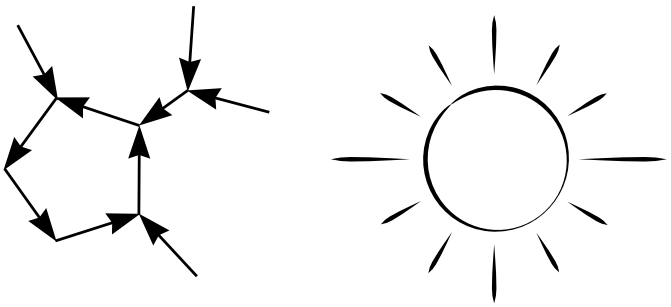
The good side of partially deterministic models is that their current state is determined. From every deterministic automata on alphabet Σ we can make partially deterministic model by defining the possibilities through statistics on the basis of life experience ($a_0, b_0, ..., a_{n-1}, b_{n-1}$). In fact, statistics will not give us the possibility but the number of times certain letter is worked out in certain situation. In this case (1, 1) is different from (30, 30) because in both cases we have 50% possibility but in the second case this is more certain.

So, we have so many partially deterministic models and the question is which of them are better. This is a very difficult question and we will not discuss it here. We will say only that if one model gives in some cases (i.e. arcs) prediction which is useful (for example 100% possibility for certain letter) and reliable (i.e. this link is used many times) then this model is useful.

How to find a good partially deterministic model? First we need a definition which strictly says which model is better. The second problem is that we have to search for this model in huge set of possible candidates.

The idea which we will give in this paper is to observe the behavior of a single letter. We will call this method the "sunshine" method for finding finite automata. This idea is based on the fact that if we observe only the arcs

which have a certain letter as a first label these arcs make one or more figures which we will call "Suns". The Sun is a cycle with paths which flow in it. This figure looks like the picture of the sun which children use to draw.



The idea is that we will be able to relatively easy detect the dependency in the figure Sun and after constructing several suns to construct the finite automate from these suns. In order to catch dependencies for one letter we will need to observe long sequences of this letter. We may wait long until the random generator generates such sequence (especially if the alphabet Σ is big and this is the general case). That is why we will use elimination of letters and construction of compound letters. Elimination of letters is when we assume that some letters do not change the state of the model. Compound letters are sets of letters which we accept as one letter. In [6, 7] we have an example of partially deterministic model where we use letters "left" and "right". There we assume that all other letters do not change the state of the model (i.e. these letters are eliminated). In the next model in [6, 7] we have the letter "victory or loss" which is compound. Actually, this compound letter is not from Σ but from Ω. Really, in the deterministic model there is no sense to include the output of the transducer as information our model depends on but in a non-deterministic case this information is essential and it is reasonable to use it in our model.

Bibliography

[1] Dobrev D. D. AI - What is this. In: PC Magazine - Bulgaria, November'2000, pp.12-13 (in Bulgarian, also in [9] in English).

[2] Dobrev D. D. AI - How does it cope in an arbitrary world. In: PC Magazine - Bulgaria, February'2001, pp.12-13 (in Bulgarian, also in [9] in English).

[3] Dobrev D. D. A Definition of Artificial Intelligence. In: Mathematica Balkanica, New Series, Vol. 19, 2005, Fasc. 1-2, pp.67-74.

[4] Dobrev D. D. Formal Definition of Artificial Intelligence. In: International Journal "Information Theories & Applications", vol.12, Number 3, 2005, pp.277-285.

[5] Dobrev D. D. Formal Definition of AI and an Algorithm which Satisfies this Definition. In: Proceedings of XII-th International Conference KDS 2006, June, 2006 Varna, Bulgaria, pp.230-237.

[6] Dobrev D. D. Testing AI in One Artificial World. In: Proceedings of XI-th International Conference KDS 2005, June, 2005 Varna, Bulgaria, pp.461-464.

[7] Dobrev D. D. AI in Arbitrary World. In: Proceedings of 5th Panhellenic Logic Symposium, July 2005, University of Athens, Athens, Greece, pp. 62-67.

[8] Dobrev D. D. First and oldest application, http://www.dobrev.com/AI/first.html

[9] Dobrev D. D. AI Project, http://www.dobrev.com/AI/

[10] Kolmogorov A. N. and Uspensky V. A. Algorithms and randomness. - SIAM J. Theory of Probability and Its Applications, vol. 32 (1987), pp.389-412.

[11] The Million Dollar Prize http://www.reiss.demon.co.uk/webgo/million.htm

Author's Information

Dimiter Dobrev – *Institute of Mathematics and Informatics, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; P.O.Box: 1274, Sofia-1000, Bulgaria; e-mail: d@dobrev.com*

SIMULATION-BASED APPROACH TO VERIFICATION OF LOGICAL DESCRIPTIONS WITH FUNCTIONAL INDETERMINACY

Liudmila Cheremisinova, Dmitry Novikov

***Abstract**: A verification task of proving the equivalence of two descriptions of the same device is examined for the case, when one of the descriptions is partially defined. In this case, the verification task is reduced to checking out whether logical descriptions are equivalent on the domain of the incompletely defined one. Simulation-based approach to solving this task for different vector forms of description representations is proposed. Fast Boolean computations over Boolean and ternary vectors having big sizes underlie the offered methods.*

***Keywords**: design automation, verification, Boolean computations, simulation.*

***ACM Classification Keywords**: B.6.2 [Logic Design]: Reliability and Testing; G.4 [Mathematical Software]: Verification; I.6.8 [Simulation and Modeling]: Types of Simulation – Parallel*

Introduction

In a typical design project more than half of the efforts go not into designing but into verifying that the design is correct. In this process the design is checked against the specification to ensure that every requirement of the specification is satisfied by its implementation. The objective of formal verification is to prove behavior equivalence of two descriptions representing different design stages of the same device. The task is well studied, if both descriptions are completely specified and are given as structural representations. This case is reduced to checking on whether two combinational circuits are equivalent. Efficient methods for equivalence checking were proposed (see [1 – 4] for example).

In the paper, the verification task is examined for a more general case, when one (initial) of logical descriptions is not completely specified. The case usually occurs on early stages of designing complex devices when assignments to primary inputs of designed device exist which will never arise during normal mode of the device usage. Thus when hardware implementing this device its outputs in response of these inputs may be arbitrary defined. The initial logic description of a verifiable combinational device is considered as a system of partially defined Boolean functions.

We are focusing on a case, when the combinational structure obtained in the process of device decomposition is a multi-block structure. Each block of the structure is described by a system of completely defined Boolean functions. The task of proving equivalence of two logical descriptions is transformed for this case into a task of checking out whether the resulting description which is defined functional completely is an extension of the initial one having functional indeterminacy; i.e. equivalence of logical descriptions is tested on the domain of the initial system of partially defined Boolean functions.

Different vector forms of representation of partially defined Boolean functions are considered, such as tuples and intervals of the Boolean space. Simulation-based approach to solving the verification task for different vector forms of description representations is proposed. The offered methods focusing on the mentioned forms of representation of Boolean functions are based on fast Boolean computations over Boolean and ternary vectors having big sizes.

Representation of a system of partially defined Boolean functions

In the paper we consider the case, when the first of the descriptions is a system $F = \{f_1(X), f_2(X), \ldots, f_m(X)\}$ ($X = \{x_1, x_2, \ldots, x_n\}$) of partially defined Boolean functions. Two cases of representing the system F are considered below:

1) on tuples (or patterns) of values of Boolean variables from X, i.e. complete Boolean assignments to the variables of X;

2) on intervals (or cubes) of values of Boolean variables from X, i.e. incomplete Boolean assignments to the variables of X.

In the first case, a partially defined Boolean function $f_i(X) \in F$ is specified by a pair of sets M_{fi}^1 and M_{fi}^0 that are on and off sets of the function f_i and consist of n-tuples $(b_1, b_2,\ldots, b_n) \in B^n$ ($b_i \in B = \{0,1\}$) of values of Boolean variables from $X = \{x_1, x_2, \ldots, x_n\}$ on which the function takes values 1 and 0 correspondingly. The value of the function is not defined (takes don't care value "–") on all other n-tuples that are neither in M_{fi}^1 nor in M_{fi}^0. In this paper, a case is considered when the system F is weakly defined, i.e. the cardinality of its domain $M = \bigcup_{i=1}^{m}(M_{f_i}^1 \cup M_{f_i}^0)$ is greatly less the cardinality of the whole Boolean space B^n, i.e. $|M| << 2^n$.

Such a system F can be represented by a pair of matrices $\boldsymbol{B}$ and $\boldsymbol{T}$ (Fig. 1,a) having the same number l of rows as the cardinality of the set $M = \bigcup_{i=1}^{m}(M_{fi}^1 \cup M_{fi}^0)$. The Boolean matrix $\boldsymbol{B}$ contains as its rows all n-tuples from the set M and the ternary matrix $\boldsymbol{T}$ specifies the values of the functions on these n-tuples. A value given on the cross of the i-th column and the j-th row of the matrix $\boldsymbol{T}$ is used to point out the value of the function $f_i(X)$ on the j-th tuple of M. The value is "1" or "0", if the function $f_i(X)$ is defined on the corresponding n-tuple, and it is "–" (don't care) otherwise. Matrices $\boldsymbol{B}$ and $\boldsymbol{T}$ have n and m columns respectively.

	x_1	x_2	x_3	x_4	x_5		f_1	f_2	f_3	
	0	1	0	1	0		–	1	0	*1*
	0	0	0	1	1		1	0	–	*2*
	1	1	0	0	0		1	–	0	*3*
B=	1	1	1	1	1	**T=**	–	0	1	*4*
	0	0	1	0	1		0	–	–	*5*
	1	0	0	1	1		0	1	1	*6*
	1	0	1	1	0		–	0	0	*7*
	1	0	0	0	1		0	–	1	*8*

a)

	x_1	x_2	x_3	x_4	x_5		f_1	f_2	f_3	
	0	1	0	–	–		0	–	–	*1*
	0	0	0	1	1		–	1	1	*2*
	1	1	–	0	0		1	–	0	*3*
U=	1	1	1	–	1	**T=**	1	0	–	*4*
	0	–	1	0	1		0	–	0	*5*
	1	0	0	1	1		1	1	–	*6*
	1	–	1	1	0		–	0	0	*7*
	1	0	0	0	–		1	1	–	*8*

b)

Fig. 1. Systems of partially defined Boolean functions represented: a) on tuples; b) on intervals

In the second case, a partially defined Boolean function $f_i(X)$ is specified by a pair of sets U_{fi}^1 and U_{fi}^0 of intervals that represent on and off sets of the function correspondingly. An interval generally specifies more than one tuple. More precisely an interval of the rank k consists of a set of 2^{n-k} n-tuples, taking into account that the rank of an interval is the number of the variables of X, having values 0 or 1. An interval represents a subcube in the Boolean space B^n and fixes the values of only k variables. The system F can be represented in these case by a pair of ternary matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ of the same cardinality (Fig. 1,b): the matrix $\boldsymbol{U}$ contains as its rows intervals of the set $U = \bigcup_{i=1}^{m}(U_{fi}^1 \cup U_{fi}^0)$ and the matrix $\boldsymbol{T}$ specifies the values of the functions of F on the intervals of U as follows. A value given on the cross of the i-th column and the j-th row of the matrix $\boldsymbol{T}$ is used to point out the value of the function $f_i(X)$ on the j-th interval of U. The value is "1" or "0", if the function is defined on all the n-tuples of the j-th interval, and it is don't care otherwise.

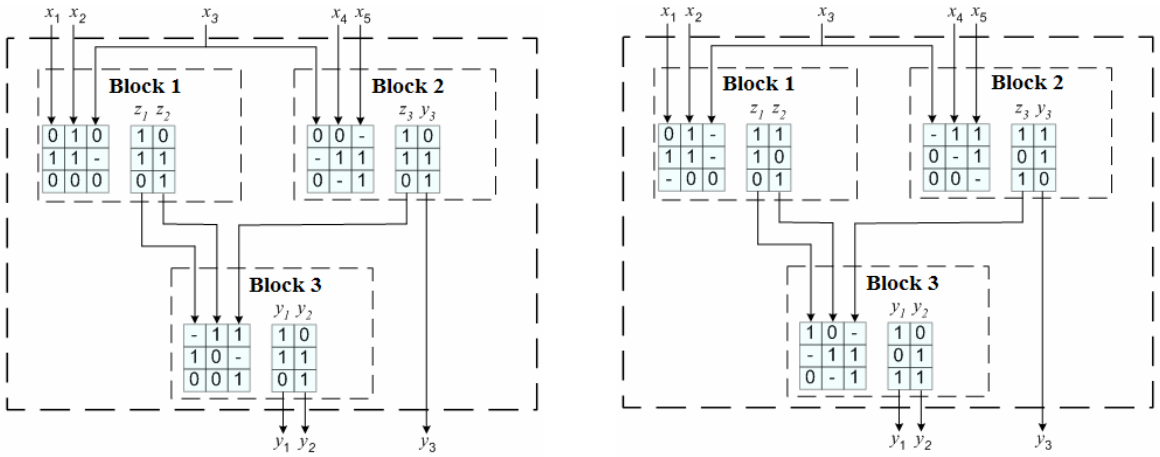
Representation of a function by the interval form has the following distinctive features. Intervals $u_i, u_j \in U$ can intersect each other (in contrast to a representation by the tuple form). The value "–" of an element t_i^j of the matrix $\boldsymbol{T}$ means that either the value of the function f_j can be not specified (is don't care) on the whole interval u_i or the function f_j does not take the same definite value (1 or 0) on the whole interval u_i, i.e. there exist at least two n-tuples belonging to the interval u_i on which the function f_j has different values from the set {1, 0, –}. Thus, the value "–" of the component t_i^j of the matrix $\boldsymbol{T}$ does not always mean that the value of the function f_j is not specified in the whole interval u_i (from such a viewpoint, that the value of f_j can be specified arbitrarily when it is implemented at the design step). So don't care value of the function f_j points only out that it can take different values inside of the interval u_i.

Transforming a representation of a multi-block structure

Each block of a multi-block structure S is a multi-output one (as in Fig. 1, 2) and is specified by a system of disjunctive normal forms (DNFs). The system of DNFs is represented by a pair of matrices: ternary and Boolean. The rows of the first one specify elementary conjunctions. The component on the cross of the i-th row and the j-th column of the Boolean matrix is equal 1, if the j-th DNF includes the i-th conjunction. The set of primary input variables of the structure S is the same as the set X of arguments of the system F. Each of m functions $y_i(X)$ implemented by the structure S must be an extension of the corresponding function $f_i(X)$ from F. A structure S implements a system F of partially defined functions if and only if for each $f_i(X) \in F$ and the corresponding $y_i(X)$ that is the i-th output of the structure S the next two relations hold:

$$M_{fi}^1 \subseteq M_{yi}^1 \text{ and } M_{fi}^0 \subseteq M_{yi}^0, \quad (1)$$

i. e. the values of f_i and y_i must be the same in the domain $M_{f_i}^1 \cup M_{f_i}^0$ of the function $f_i(X)$.



a) b)

Fig. 2. A three-block structures implementing systems of Boolean functions shown: a) in Fig. 1,a; b) in Fig. 1,b

Each block realizing a system of disjunctive normal forms can be considered as a three-level multi-output combinational circuit. Its first level consists of invertors which are used to invert those primary input variables of the block, which are represented in the inverse form at least in one conjunction. The second level is formed by multi-input AND gates implementing conjunctions, and the last (third) level is composed by multi-input OR gates. AND gates are specified in a natural way by the intervals from M. For example the interval 1 –0 1 – 0 corresponds to the conjunction $x_1\ \bar{x}_3\ x_4\ \bar{x}_6$ implemented by 4-input AND gate and two invertors fed upon variables x_3 and x_6.

Let us number gates inside of blocks of the multi-block structure and blocks themselves in a topological order, i.e. in such a way that any connection between gates will connect an output of a gate having a smaller number to an input of a gate having a bigger number. It should be pointed out that the necessary and sufficient condition for making the ordering is absence of feedbacks in the structure. The condition holds for each intrablock structure, and we assume that the structure itself satisfies the condition.

As the result the multi-block structure can be considered as a multi-output combinational circuit C which consists of invertors, AND and OR gates. Let us assign internal variables z_i to outputs of all the elements of C, marking out those of them that are used to implement m Boolean functions $y_j(X)$ (primary outputs). A method proposed below can be extended for nets consisting of any other elements implementing symmetric logic operations $\varphi(z_1, z_2, \ldots, z_k)$.

Solving the verification task for the tuple based representation

An idea of the proposed method for checking out whether a system F of partially defined Boolean functions is implemented by a multi-block structure S is to simulate a combinatorial circuit C corresponding to the structure on the domain of the system F. The system F is implemented by the structure S, if for each n-tuple $\boldsymbol{b}_j \in \boldsymbol{B}$ of values assigned to the variables from X the equality $f_i(\boldsymbol{b}_j) = y_i(\boldsymbol{b}_j)$ holds for all i for which the value $f_i(\boldsymbol{b}_j)$ is defined (i.e. is

equal to 0 or 1). In other words, the vector $\boldsymbol{f}(\boldsymbol{b}_j)$ being the j-th row of the matrix $\boldsymbol{T}$ must cover the vector $\boldsymbol{y}(\boldsymbol{b}_j)$ consisting of values generated by primary outputs of the circuit C under assignment $\boldsymbol{b}_j$ to its primary inputs. A ternary (or binary) vector $\boldsymbol{a}$ is covered by a ternary vector $\boldsymbol{b}$ of the same size, if for all components of $\boldsymbol{b}^i$ which are equal to 1 or 0 the equality $\boldsymbol{b}^i = \boldsymbol{a}^i$ holds.

Thus according to condition (1) we are interested in values of functions y_i on n-tuples $\boldsymbol{b}_j$ from the set M represented by rows of the matrix $\boldsymbol{B}$.

An idea of parallel binary simulation [5] is used. The combinational circuit C is simulated under all possible inputs simultaneously, i.e. all tuples from the set M are examined at the same time. When parallel simulation of the circuit is performed under all l n-tuples from M at the same time, a state of each node of the circuit (corresponding to a primary input or output of a gate) is represented by a Boolean vector. The latter has the size l and specifies the values of the same variable in all l tuples.

Thus, each Boolean vector represents states of a corresponding node for all l considered assignments to primary inputs of the simulated circuit, and the union of the i-th components of all the vectors describes the state of all nodes of the circuit for the i-th assignment to primary inputs.

At the beginning of the simulation, the ordered set of n Boolean vectors having the size l is taken; they correspond to the columns of the matrix $\boldsymbol{B}$. Then gates of the circuit C are simulated in the predefined topological order. Let a gate of the circuit implement a function $\varphi_i(z_{1i}, z_{2i}, \ldots, z_{ki})$. As each argument z_{ji} is related to a Boolean vector $\boldsymbol{z}_{ji}$ having been computed already, the simulation is reduced to computation of the operation φ_i over Boolean vectors $\boldsymbol{z}_{1i}, \boldsymbol{z}_{2i}, \ldots, \boldsymbol{z}_{ki}$ in the bitwise style. The result of the simulation is a new Boolean vector $\boldsymbol{z}_i$ of the same size l.

As soon as the last gate of the circuit has been simulated, value assignments to the primary outputs of the circuit C for all assignments to the primary inputs belonging to the domain M of the system F are found. At that each output function y_i of the circuit has a definite value (0 or 1) for each assignment used to the primary inputs, in particular for those assignments, for which corresponding function $f_i \in F$ has a definite value as well.

Thus, we need to check the orthogonality of the values of the functions y_i and f_i on the domain $M = M_{fi}^1 \cup M_{fi}^0$. This is reduced to checking out whether the following pairs of vectors are orthogonal: the ternary vector $\boldsymbol{t}^i$ corresponding to the i-th column of the matrix $\boldsymbol{T}$ and the Boolean vector $\boldsymbol{z}_{ip}$ corresponding to the primary output y_i of the circuit. The multi-block structure S implements the system F, if all these pairs of the vectors are not orthogonal. Otherwise, the block responsible for violating the implementing condition can be found by back traversal of the combinational circuit C.

To demonstrate the method proposed let us check out that the system of partially defined Boolean functions shown in Fig. 1,a is implemented by the three-block structure depicted in Fig. 2,a. A sequence of Boolean vectors generated by parallel simulation of the circuit C simultaneously for the whole domain of the system F is shown below. Boolean vectors representing values of circuit nodes are accompanied with operations performed over the vectors corresponding to their arguments. Here the variable corresponding to the i-th output of AND gate of j-th block of the structure is denoted by $k_{i,j}$ and Boolean vectors representing states of primary outputs of the structure C are printed in bold.

Inputs.	0 0 1 1 0 1 1 1	x_1
	1 0 1 1 0 0 0 0	x_2
	0 0 0 1 1 0 1 0	x_3
	1 1 0 1 0 1 1 0	x_4
	0 1 0 1 1 1 0 1	x_5
Block 1.	1 1 0 0 1 0 0 0	x_1
	0 1 0 0 1 1 1 1	x_2
	1 1 1 0 0 1 0 1	x_3
	1 0 0 0 0 0 0 0	$k_{1,1} = x_1 \wedge x_2 \wedge \overline{x_3}$
	0 0 1 1 0 0 0 0	$k_{2,1} = x_1 \wedge x_2$
	0 1 0 0 0 0 0 0	$k_{3,1} = x_1 \wedge \overline{x_2} \wedge \overline{x_3}$
	1 0 1 1 0 0 0 0	$z_1 = k_{1,1} \vee k_{2,1}$
	0 1 1 1 0 0 0 0	$z_2 = k_{2,1} \vee k_{3,1}$
Block 2.	0 0 1 0 1 0 0 1	x_4
	0 0 1 0 0 0 0 1	$k_{1,2} = \overline{x_3} \wedge \overline{x_4}$
	0 1 0 1 0 1 0 0	$k_{2,2} = x_4 \wedge x_5$
	0 1 0 0 0 1 0 1	$k_{3,2} = \overline{x_3} \wedge x_5$
	0 1 1 1 0 1 0 1	$z_3 = k_{1,2} \vee k_{2,2}$
	0 1 0 1 0 1 0 1	$\boldsymbol{y_3} = k_{2,1} \vee k_{3,1}$
Block 3.	0 1 0 0 1 1 1 1	z_1
	1 0 0 0 1 1 1 1	z_2
	0 1 1 1 0 0 0 0	$k_{1,3} = z_2 \wedge z_3$
	1 0 0 0 0 0 0 0	$k_{2,3} = z_1 \wedge z_2$
	0 0 0 0 0 1 0 1	$k_{3,3} = \overline{z_1} \wedge z_2 \wedge z_3$
	1 1 1 1 0 0 0 0	$\boldsymbol{y_1} = k_{1,3} \vee k_{2,3}$
	1 0 0 0 0 1 0 1	$\boldsymbol{y_2} = k_{2,3} \vee k_{3,3}$

When comparing pairs, including ternary vector representing values of functions $f_i \in F$ and Boolean vector derived under simulation for the corresponding primary output y_i, we see that for all pairs the second one is covered by the first:

f_1: – 1 1 – 0 0 – 0	f_2: 1 0 – 0 – 1 0 –	f_3: 0 – 0 1 – 1 0 1
y_1: 1 1 1 1 0 0 0 0	y_2: 1 0 0 0 0 1 0 1	y_3: 0 1 0 1 0 1 0 1

The example demonstrates as the considered task is reduced to Boolean computations over vectors (sequences of bits) having the same (but arbitrary) size.

Solving the verification task for the interval based representation

In this case, the verification task can be solved by one of the following ways: 1) by unfolding intervals of the domain of the system into sets of tuples, i.e. by reducing the task to the case considered above (when a system of partially defined Boolean functions is in a tuple based form); 2) by solving the task directly by using the interval based representation. The first way can be used, when the number of intervals of the set M having ranks less than n is not big and these ranks are close to n. In this case, an n-tuple based representation of the domain of a given system of partially defined Boolean functions will be not much bigger than an interval based representation given. The second way can be used, when the number of intervals of the set M having ranks less than n is big and/or these ranks are much less than n. In this case, unfolding intervals could be impossible at practice. Further we discuss this second way.

As well as earlier we will perform parallel simulation of the circuit C, in the discussed case the simulation will be carried out on intervals given as rows of the matrix $\boldsymbol{U}$. Now during circuit simulation, a state of every node (including those corresponding to primary inputs) of the circuit is represented by a ternary vector. Thus, each ternary vector represents states of the corresponding node for all l considered partial assignments to primary inputs (specified by intervals from M) of the simulated circuit, and the union of the i-th components of all the state vectors describes the state of all nodes of the circuit for the i-th partial assignment to primary inputs (defined by the i-th interval of M). In this case, the don't care value of the i-th component of a state vector of the j-th node means only that a function implemented by the node can have different values for different n-tuples of the i-th interval of M.

At the beginning of the simulation, the ordered set of n of ternary vectors having the size l is taken. The vectors represent states of n primary inputs and correspond to the columns of the matrix $\boldsymbol{U}$. Then gates of the circuit C are simulated in the predefined topological order. Let a gate implementing the function $\varphi_i(z_{1i}, z_{2i}, \ldots, z_{ki})$ is simulated. As for each its argument z_{ji} a ternary vector $\boldsymbol{z}_{ji}$ having been computed already corresponds to, the simulation of the gate is reduced to performing the logic operation φ_i over ternary vectors $\boldsymbol{z}_{1i}, \boldsymbol{z}_{2i}, \ldots, \boldsymbol{z}_{ki}$ in the bitwise style. The result of the simulation is a new ternary vector $\boldsymbol{z}_i$ of the same size l. A definition of basic operations over ternary variables and vectors is given bellow (all possible combinations of two ternary values are considered and don't care is interpreted as uncertainty):

$\boldsymbol{a}$:	0 0 0 – – – 1 1 1
$\boldsymbol{b}$:	0 – 1 0 – 1 0 – 1
$\bar{\boldsymbol{a}}$:	1 1 1 – – – 0 0 0
$\boldsymbol{a} \vee \boldsymbol{b}$:	0 – 1 – – 1 1 1 1
$\boldsymbol{a} \wedge \boldsymbol{b}$:	0 0 0 0 – – 0 – 1

After the simulation process is over, it is necessary to check whether the system F of functions is implemented by the multi-block structure S. This checking can be time consuming. As soon as the last gate of the circuit has been simulated the following pairs of vectors are compared: the ternary vector $\boldsymbol{t}^p$ (p = 1, 2,…, m) corresponding to the p-th column of the matrix $\boldsymbol{T}$ (the column of values of the function $f_p \in F$) and the ternary vector $\boldsymbol{z}_{ip}$ corresponding to the primary output y_p of the circuit C. The following three cases are possible:

1. Vectors $\boldsymbol{t}^p$ and $\boldsymbol{z}_{ip}$ are orthogonal in some component. Hence, the circuit C does not implement the function f_p.

2. The vector $\boldsymbol{t}^p$ covers the vector $\boldsymbol{z}_{ip}$, i.e. values of all definite (1 or 0) components of the vector $\boldsymbol{t}^p$ are the same as the values of the corresponding components of $\boldsymbol{z}_{ip}$. In this case, the circuit C implements the function f_p.

3. The value of some j-th component (corresponding to the interval $\boldsymbol{u}_j$ of the matrix $\boldsymbol{U}$) of the vector $\boldsymbol{z}_{ip}$ is don't care, while the value of the corresponding component of the vector $\boldsymbol{t}^p$ is equal 1 or 0. In this case, there exists no unambiguous answer whether the circuit C implements the function f_p or does not.

In the third case, an additional analysis is needed to detect the reason of distinction of the values of the j-th components of the vectors $\boldsymbol{z}_{ip}$ and $\boldsymbol{t}^p$. The simplest way is to simulate the circuit C once more on all tuples of the interval $\boldsymbol{u}_j$ of the matrix $\boldsymbol{U}$. A more refined method is based on-the-fly analysis of the simulation process. If the output z_q of a gate of the circuit C gets don't care value, the interval corresponding to the value assignment to the inputs of the gate is to be decomposed into subintervals, in which the output z_q gets a definite value. This will help us to prevent propagation of indeterminacy during the simulation. For example, if $z_q = x_1 x_3 x_4 x_6$, and computations are to be performed for the interval u_i = 1 0 – 1 – –, resulting in $z_q(u_i)$ = –, then the interval is decomposed into three following ones: u_i^1 = 1 0 1 1 – 1, : u_i^2 = 1 0 0 1 – –, u_i^3 = 1 0 – 1 – 0. For these subintervals, z_q takes definite values: $z_q(u_i^1) = 1$, $z_q(u_i^2) = z_q(u_i^3) = 0$.

To demonstrate the method proposed let us check out that the system shown in Fig. 1,b of partially defined Boolean functions is implemented by a three-block structure depicted in Fig. 2,b. A sequence of ternary vectors generated by parallel simulation of the circuit C for intervals of the matrix $\boldsymbol{U}$ is shown below.

Inputs.	0 0 1 1 0 1 1 1	x_1
	1 0 1 1 – 0 – 0	x_2
	0 0 – 1 1 0 1 0	x_3
	– 1 0 – 0 1 1 0	x_4
	– 1 0 1 1 1 0 –	x_5
Block 1.	1 1 0 0 1 0 0 0	x_1
	0 1 0 0 – 1 – 1	x_2
	1 1 – 0 0 1 0 1	x_3
	1 0 0 0 – 0 0 0	$k_{1,1} = \bar{x}_1 \wedge x_2$
	0 0 1 1 0 0 – 0	$k_{2,1} = x_1 \wedge x_2$
	0 1 0 0 0 1 0 1	$k_{3,1} = \bar{x}_2 \wedge \bar{x}_3$
	1 0 1 1 – 0 – 0	$z_1 = k_{1,1} \vee k_{2,1}$
	1 1 0 0 – 1 0 1	$z_2 = k_{1,1} \vee k_{3,1}$
Block 2.	– 0 1 – 1 0 0 1	x_4
	– 1 0 – 0 1 0 0	$k_{1,2} = x_4 \wedge x_5$
	– 1 0 0 0 1 0 –	$k_{2,2} = \bar{x}_3 \wedge x_5$
	– 0 – 0 0 0 0 1	$k_{3,2} = \bar{x}_3 \wedge \bar{x}_4$
	– 1 – – 0 1 0 1	$z_3 = k_{1,2} \vee k_{3,2}$
	– 1 0 – 0 1 0 –	$\boldsymbol{y_3} = k_{1,2} \vee k_{2,2}$
Block 3.	0 1 0 0 – 1 – 1	z_1
	0 0 1 1 – 0 1 0	z_2
	0 0 1 1 – 0 – 0	$k_{1,3} = z_1 \wedge \bar{z}_2$
	– 1 0 0 0 1 0 1	$k_{2,3} = z_2 \wedge z_3$
	0 1 0 0 0 1 0 1	$k_{3,3} = \bar{z}_1 \wedge z_3$
	0 1 1 1 – 1 – 1	$\boldsymbol{y_1} = k_{1,3} \vee k_{3,3}$
	– 1 0 0 0 1 0 1	$\boldsymbol{y_2} = k_{2,3} \vee k_{3,3}$

When comparing pairs, each of which includes ternary vector representing value of a function $f_i \in F$ and Boolean vector derived under simulation for the corresponding primary output y_i, we see that for all pairs but one the second of the vectors is covered by the first one:

f_1: 0 – 1 1 **0** 1 – 1	f_2: – 1 – 0 – 1 0 1	f_3: – 1 0 – 0 – 0 –
y_1: 0 1 1 1 **–** 1 – 1	y_2: – 1 0 0 0 1 0 1	y_3: – 1 0 – 0 1 0 –

The only exception is in the 5-th component of f_1 for which the case 3 takes place. By splitting the interval 0–101 into two tuples and simulating the structure on them we find out $y_1(00101) = y_2(01101) = 0$, i.e. y_1 implements f_1.

In order to avoid unnecessary decompositions of intervals during simulation, it makes sense to orthogonalize the initial system of partially defined Boolean functions. A system is orthogonalized, if any interval u_i of the domain M has the following property: every function of the system F has the same value (1, 0, –) for all tuples forming u_i . In this case, the value "–" of a function means that the value of this function is undefined for all tuples of the interval and may be arbitrarily redefined during implementing the system by circuit.

Orthogonalization of a system of partially defined Boolean functions

In [6, 7], an orthogonalization task is examined for a system of completely defined Boolean functions $F = \{f_1(X), f_2(X), \ldots, f_m(X)\}$ which is represented by a pair of matrices: ternary and Boolean. Rows of the ternary matrix represent intervals. If an element in the Boolean matrix takes the value 1, then the corresponding function takes the value 1 for the corresponding interval. The task is to find a set of pair-wisely orthogonal completely defined Boolean functions $\varphi_1(X), \varphi_2(X), \ldots, \varphi_r(X)$ such that any function $f_i \in F$ can be represented as disjunction of some pair-wisely orthogonal functions φ_j (j = 1, 2, …, r), and the number r of different functions φ_i must be minimal. Functions φ_j and φ_k are orthogonal, if the condition $\varphi_j \wedge \varphi_k = 0$ holds for any value assignment to variables of X. At that, if functions φ_i are all reciprocally orthogonal, the intervals for which different functions φ_j and φ_k are defined don't intersect each other, however intervals for which the same function is defined may intersect each other.

In a similar manner as for systems of completely defined Boolean functions the orthogonalization task can be formulated for a system of partially defined Boolean functions $F = \{f_1(X), f_2(X), \ldots, f_m(X)\}$, when each $f_i \in F$ is represented by a pair of functions f_i^1 и f_i^0. The on set of the function f_i^1 is the union of intervals for which f_i takes the value 1. The on set of the function f_i^0 is the union of intervals for which f_i takes value 0. In the parts of the Boolean space that are complementary to on and off sets, the functions f_i^1 и f_i^0 take the value 0. In this way, we come to the system F' of completely defined Boolean functions having twice more functions than the initial system F of partially defined Boolean functions. Now, the orthogonalization task can be solved for F' by some method from [6,7], and then a backward transition of F' into the orthogonalized system F can be performed.

A distinguishing feature of an orthogonalized system of partially defined Boolean functions is that any two intervals for which a function of the system takes different values are not intersect. In this case, the don't care value of a function of the system for any interval means that the function is not defined on all tuples of the interval, and a definite value of the function can be arbitrary assigned under a synthesis procedure.

Conclusion

Methods proposed in the paper are focused on systems of weakly defined Boolean functions which have much smaller domains than Boolean spaces in which the systems are specified. The methods can be used for verifying complex logical descriptions.

The task of checking out whether a multi-block structure implements a system of partially defined Boolean functions is reduced to Boolean computations over ternary and Boolean vectors having an arbitrary size. Methods can be implemented effectively by using classes CBV [8] и CTM [9] developed for performing operations over Boolean and ternary vectors of arbitrary sizes in C++. Computational complexity of the methods linearly depends on the total number of nodes of the simulated combinational circuit (more precisely, on the total number of inputs of all gates of the circuit) and on the number of bytes (or 32-bit words) used to represent an l-bit vector.

Bibliography

[1] Drechsler R. (Ed.). Advanced Formal Verification. – Kluwer Academic Publishers, 2004. – 260 p.

[2] Mishchenko A., Chatterjee S., Brayton R., Een N. Improvements to Combinational Equivalence Checking // Proc. ICCAD'06, Nov. 5–9, 2006. – San Jose, CA, 2006.

[3] Ganai M.K., Zhang L., Ashar P., Gupta A., Malik S. Combining strengths of circuit-based and CNF-based algorithms for a high-performance SAT solver // Proc. ACM/IEEE Design Automation Conference, 2002 – P. 747–750.

[4] Goldberg E., Novikov Y. BerkMin: A fast and robust SAT-Solver // Proc. European Design and Test Conference, 2002. – P. 142–149.

[5] Zakrevskii A.D., Pottosin U.V. Cheremisinova L.D. Foundations of logic design. Book 3. Design of logic control devices. – Minsk: National Academy of Sciences of Belarus, United Institute of Informatics Problems of National Academy of Sciences of Belarus, 2006. – 252 p. (in Russian).

[6] Kuznetzov O.P. Orthogonal systems of Boolean functions and their application to analysis and synthesis of logical nets // Automation and remote control, 1970. – N 10, – P. 117–128 (in Russian).

[7] Pottosin U.V., Shestakov E.A. Orthogonalization of systems of completely defined Boolean functions // Logic design. – Minsk: The Institute of Engineering Cybernetics of National Academy of Sciences of Belarus. – 2000. – Vol. 5. – P. 107–115 (in Russian).

[8] Vasilkova I.V., Romanov V.I. Boolean vectors and matrices in C++ // Logic design. – Minsk: The Institute of Engineering Cybernetics of National Academy of Sciences of Belarus. – 1997. – Vol. 2. – P. 150–158 (in Russian).

[9] Cheremisionov D.I., Cheremisinova L.D. Ternary vectors and matrices in C++ // Logic design. – Minsk: The Institute of Engineering Cybernetics of National Academy of Sciences of Belarus. – 1998. – Vol. 3. – P. 146–156 (in Russian).

Authors' Information

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

Dmitry Novikov *– Post-graduate student, The United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganov St., 6, Minsk–220012, Belarus; e-mail: yakov_nov@tut.by*

WORKFLOW MODELLING IN GRID SYSTEM FOR SATELLITE DATA PROCESSING

Andrii Shelestov

***Abstract**: This paper focuses on a problem of Grid system decomposition by developing its object model. Unified Modeling Language (UML) is used as a formalization tool. This approach is motivated by the complexity of the system being analysed and the need for simulation model design.*

***Keywords**: Grid system, satellite data processing, modeling.*

Introduction

In the past decades a great attention has been paid to the development of Grid technologies. Description of the state-of-the-art in the Grid system development can be founded in [1, 2, 3, 4, 5]. Since Grid system is a complex distributed system that enables solution of complex problems in different domains, the development of such systems requires the use of system analysis approach. It is especially important to Grid systems for environment monitoring where heterogeneous resources are used: satellite and in-situ data, modeling data, etc. Such Grid system is not just a computational system, or it is intended for data management either. It represents a set of virtual organizations that jointly solve complex problems using data from geographically distributed archives with given level of reliability and security. The examples of such initiatives include GMES (Global Monitoring for Environment and Security) [6], GEOSS (Global Earth Observation System of Systems) [7] and WAG (Wide Area Grid) [1], where Ukraine is among the participants. Analysis of such complex systems requires approaches that involve system decomposition and structural and functional analysis of system components to provide the further optimization and efficient management [8].

In this paper we carry out Grid system decomposition by developing its object model. For this purpose Unified Modeling Language (UML) is applied.

Properties of Grid systems for satellite data processing

Grid system integrates computational, informational and other kinds of resources that are managed by geographically distributed organizations. Traditionally such systems are used for complex computational problems solving that require the use of high-performance computing, or problems that require processing of high volumes of data. The example include EGEE (http://www.eu-egee.org) project that is aimed at solving problems in high-energy physics [9], gravitational waves [10], astronomy [11], and bio-informatics [11]. Such Grid systems are considered as computational Grids.

Recent years much attention has been paid to the development of Grid system for environmental monitoring with the use of satellite data [2]. In such systems Grid infrastructure can be used not only for providing high-performance computations, but also for efficient data management. The Earth observation (EO) data should be processed, catalogued, and archived [13]. For example, GOME instrument onboard Envisat satellite generates nearly 400 Tb data per year [14]. EUMETCast system for environmental data dissemination that is a part of global GEONETCast system of GEOSS [15] enables acquisition of more than 50 Tb of processed and unprocessed information per year. Moreover, satellite data are processed not by the single application with monolithic code, but by distributed applications. This process can be viewed as complex workflow that is composed of many tasks: geometric and radiometric calibration, filtration, reprojection, composites creation, classification, products generation, post-processing, visualization, etc. [16]. For example, calibration and mosaic composition of 80 images generated by ASAR instrument onboard Envisat satellite takes 3 days on 10 workstations of Earth Science GRID on Demand that is being developed in ESA and ESRIN.

Thus, complexity of Grid systems for environmental monitoring and decision support using satellite data is of particular interest from system analysis point of view. The investigation of specifics and peculiarities of such systems is primary objective of DGREE project [17] initiated within EGEE-II project. The reported results provide the following requirements to the Grid system for satellite data processing:

— Security requirements: satellite data and corresponding software are distributed according to specialized licenses. So, observance of license agreements is one of the main issues of such systems.

— Reliability: such systems should operate in operational mode and should provide required reliability and QoS of the results.

— Standardization: applied problems should be solved according to standardized, verified and validated methods. Interoperability issues are also of a great concern.

These requirements provide flexibility during workflow execution, and necessity to efficient management of resources on physical and logical layers. One of the important issues is structural and functional analysis based on system modeling.

Grid System Modeling

The task of Grid system modeling is very important and motivated by the following goals. Since the development of Grid system requires large amounts of financial resources to be spent, the modeling will allow the developer to design optimal architecture of such distributed system. Another issue corresponds to the modeling of the existing system. The development of system workload model will allow one to reveal system bottleneck, to estimate system capacity and to plan future trends of its extension. And at last, system modeling is a part of task of system performance forecasting and development of scheduling techniques for efficient resource management. By resource management we mean estimation of optimal computational resources parameters to meet the requirements of applied problems being solved, resource discovery, reservation, scheduling and monitoring. To develop optimal methods for resource management one need to run many experiments with established parameters of the system and established external conditions. Such experiments in real systems are almost impractical due to the following reasons:

— computational resources are managed by different organizations that complicates experiment;

— user requirements and properties of system resources are evolving through the time that makes it difficulty to repeat the experiment under the same conditions;

— development of Grid infrastructure to the specific experiment is time and cost-consuming.

All these factors make the modeling the only practical approach to the analysis of Grid infrastructure, its internal properties and analysis of influence of external conditions.

Approaches to Grid System Modeling

Different approaches to Grid system modeling can be used among which we can consider analytical or statistical models [18]. As a rule, analytical expression can be used to describe the properties of the system under specific assumptions such as independence of parameters, linearity, instantaneity of state transitions, etc. If these assumptions apply, the model will be in good correspondence with real object. Otherwise, there will be considerable difference between the model and real object.

That is why, for the modeling of complex distributed Grid systems simulation modeling is applied. Simulation entails the functioning of the Grid system being analyzed by integrating its elements in the single structure and imitating its interaction. Advantages of simulation modeling approach are its generality, the possibility to simulate systems of any complexity, possibility to acquire new data about properties of the system. All these factors enable detailed analysis of the system and its components. In this paper simulation modeling is applied to the Grid system intended for satellite data processing.

Classification of tasks in Grid system for satellite data processing

One of the main issues in Grid system modeling is task description and modeling. Data about tasks represent as a rule inputs to Grid system models, as well as system load for estimation and forecasting of system productivity. That is why adequate description of tasks in Grid systems will enable effective modeling of Grid system and its load. In the framework of task 2.6 tasks in Grid system for satellite data processing were classified on *Data Transfer Task* (DTT) and *Computational Task* (CT). These types of tasks represents “building blocks” for more complex tasks for environmental monitoring and decision support.

Data Transfer Task are characterized by the volume of transferred information which provides the following requirements to the system: bandwidth connection (to Internet or local network), data storages (hard drives or magnetic strips) I/O speed. Data sources, frequency and QoS should also be taken into account.

DTTs are represented by the following set of parameters:

- task identifier;
- frequency (regularly or by user query);
- input and output data volumes;
- data source (e.g. Internet, local network, hard drive, magnetic strips, etc.).

Computational Tasks represent the unit of program that carry out data processing or computations. CT can be run on either single processor or in parallel mode on different processors.

CTs are represented by the following set of parameters:

- task identifier;
- frequency (regularly or by user query);
- complexity of problem (computational complexity, memory requirements, software and hardware requirements, etc.);
- parallelism by code or data;
- number of processes required.

Another type of tasks that should be mentioned is the task of control – database search or other control instructions. But these tasks can be viewed as CT ones.

Formal task description

In order to begin with formal description of tasks with definition of Data type structure. Data can be represented by the following setoff parameters:

$$Data = \{ID, V, DS, Sec\}, \quad (1)$$

where ID — task identifier, V — data volume (in Mb), DS — data source: local discs, data storage, resource in local system or Internet. DS can be one of the following values:

DS = {local disc, Internet, data storage, local network}.

In (1) Seq describes level of required security.

DTT can be described by:

$$DTT = \{ID, Freq, I/O: Data\}, \quad (2)$$

where ID — task identifier, Freq — frequency of task completion:

$$Freq = \{cycle, request\}. \quad (3)$$

If the task is carried out periodically then Freq = cycle, and determines the period of time the task is being completed (e.g. in minutes). If the task is initiated by the query of user or other system then Freq = request, and the query time should be given by some distribution.

CT can be described by the following set:

$$CT = \{ID, Freq, C, Par\}. \quad (4)$$

In Eq. (4): ID — task identifier, Freq — frequency according to (3), C — task complexity described by the following equation

$$C=\{CC, Size, Op\}, \quad (5)$$

where CC — computational complexity, Size — required memory, Op — additional requirements (e.g. requirements to software or hardware).

In Eq. (4) Par defines parallelism of the task: the type of parallelism (by code or data), and number of required processors NumProcess.

Based on the described analysis UML class diagram for tasks of Grid system for satellite data processing was developed (Fig. 1).

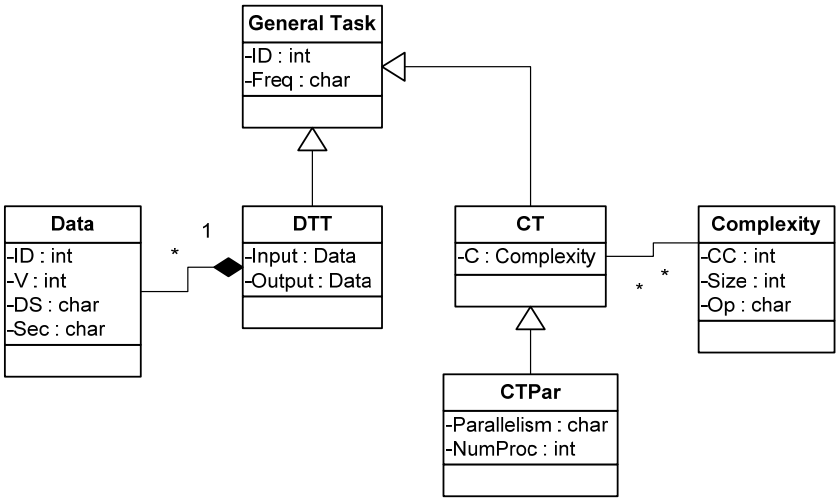


Fig. 1. Class diagram for tasks running in Grid system for satellite data processing

Description of complex tasks in Grid system for satellite data processing

The set of tasks that compose a workflow – a job – can be described by the directed acyclic graph (DAG). The nodes of the DAG represent elementary tasks – CT or DTT. Edges correspond to interdependency between tasks. Thus, a job is described by the following expression:

$$Job = \{S, W, QoS\},$$

where $S = CT \cup DTT$ — the set of basic tasks: CT or DTT, $W \subset SxS$ — set of pairs (s_i, s_j) with edge that directed from s_i to s_j, QoS — quality of service (for example, TTS — time-to-schedule time, maximum number of repeated completions).

Example of object model for the problem of biodiversity estimation using remote sensing data from space

In the framework of innovation project Space Research Institute NASU-NSAU and Centre for Aerospace Research of the Earth NASU have developed information service for estimation biodiversity in Near Black Sea region using remote sensing data from space [19]. Biodiversity refers to the variation of taxonomic life forms within a given ecosystem, biome or for the entire Earth. Species biodiversity is characterized by two criteria: the total number of species and distribution between species [20].

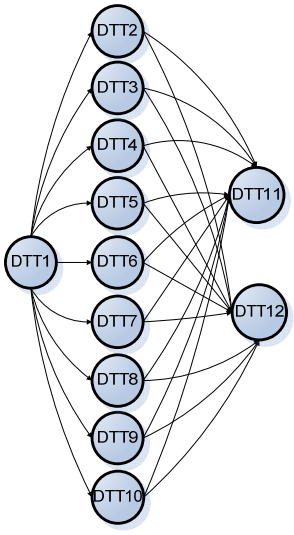


Fig. 2. DAG for data acquisition process

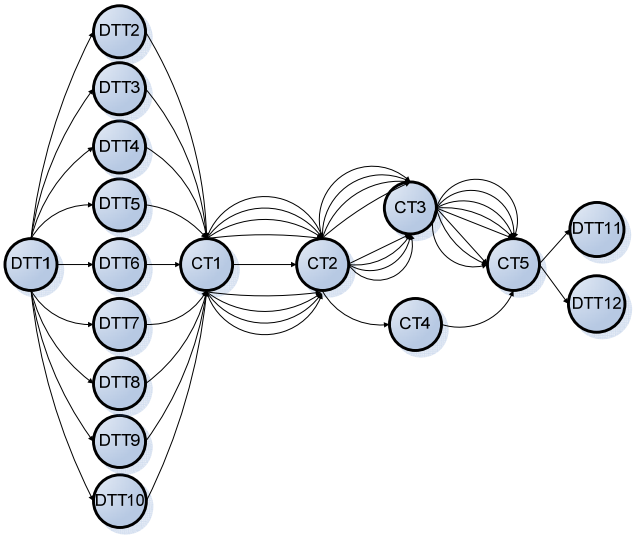


Fig. 3. DAG for data processing step

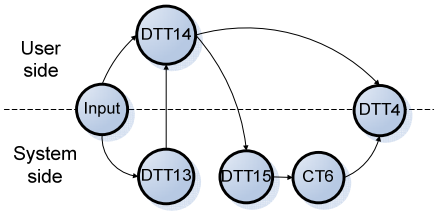


Fig. 4. DAG for the task of visualization of processing results

The workflow for biodiversity estimation using satellite data acquired by MODIS instrument aboard Terra satellite is organized as follows:
— data acquisition;
— data processing;
— visualization of the results.

The detailed description of these steps and corresponding sequence diagrams for this task are given in [21]. In this paper we focus on the presentation of workflow in the form of direct acyclic graph (DAG). Using the mentioned above approach to the decomposition of complex tasks into simple one, we represent the complex workflow for biodiversity estimation as DAGs (Fig. 2-4).

Corresponding notations to the figures 2-4 are given in table 1.

Table 1. Notations to the DAGs for biodiversity estimation task

Title	Description	Tasks structure
DTT1	Data transfer initialization	
DTT2	Data transfer (DT) MOD03A2	{cycle/24 h, I/O: 30 Mb, Internet}
DTT3	DT MOD05L2	{cycle/12 h, I/O: 6 Mb, Internet}
DTT4	DT MOD11A1	{cycle/12 h, I/O: 20 Mb, Internet}
DTT5	DT MOD12Q1	{cycle/0,5 year, I/O: 3 Mb, Internet}
DTT6	DT MOD13Q1	{cycle/16 days, I/O: 30 Mb, Internet}
DTT7	DT MOD15A2	{cycle/16 days, I/O: 5 Mb, Internet}
DTT8	DT MOD17A3	{cycle/8 days, I/O: 7 Mb, Internet}
DTT9	DT AE_Land3	{cycle/1 months, I/O: 30 Mb, Internet}
DTT10	DT SRTM DEM	{request, I/O: 1,5Gb, Internet}
DTT11	Data saving (in storage)	
DTT12	Updating data (in index service)	
CT1	Data reprojection	CT = {request, C, DP} C={O(N*M), 1,61 Gb}, where DP — data parallelism; N, M — image sizes.
CT2	Data scaling	CT = {request, C, DP} C={O(N*M), 1,61 Gb}
CT3	Composite creation	CT = {request, C, DP} C={O(N*M*t), 101*t Mb}, where t — averaging period.
CT4	Solar irradiation estimation using digigtal elevation model	CT = {request, C, PD} C={O(N*M), 1,5 Gb}
CT5	Biodiversity estimation	
DTT13	Obtaining parameters of processed data	{request, I/O: 100Kb, storage}
DTT14	Selecting parameters to be visualized	{request, I/O: 100Кб, Internet}
DTT3	Obtaining data for layers generation	{request, I/O: 1Mb, storage}
CT6	Layer generation by map server	CT = {request, C, DP} C={O(N*M), 1Mb}
DTT4	Transferring data to client browser	{request, I/O: 1Mb, Internet}

The presentation of complex tasks in the form of DAG is more suitable for scheduling and allows one to identify possibilities for parallelization.

Conclusions

In this paper we described requirements to the Grid systems aimed at satellite data processing for applied problems solving. We provided classification and description of different types of tasks executed in such Grid systems. We proposed an object model of complex task that is composed of a set of simple tasks and presented it in the form of directed acyclic graph. Such approach enables automatic creation workflows that need to be executed in Grid system. We applied our approach to task of biodiversity estimation using remote sensing data from space. Future works will be directed to the development of model of system resources and extension of software tools for simulation of Grid systems.

Acknowledgement

This research is supported by the Science and Technology Center in Ukraine (STCU) and the National Academy of Sciences of Ukraine (NASU) within project “GRID technologies for environmental monitoring using satellite data”, no. 3872.; and INTAS-CNES-NSAU project “Data Fusion Grid Infrastructure”, Ref. No 06-1000024-9154.

Bibliography

1. Shelestov A.Ju., Kussul N.N., Skakun S.V. Grid-technologies in monitoring systems based on satellite data // J. of Automation and Control. — 2006. — № 1-2. — P. 259-270. (in Russian)
2. Shelestov A.Yu., Kravchenko A.N., Korbakov Ì.B., Kussul N.N., Skakun S.V., Rudakova A.I., Illin N.I., Tyutyunnik L.I. Grid-technology of Implementation of GEOSS Ukrainian Segment // J. of Communications (special edition). – 2006. - P. 106-125. (in Russian)
3. Zagorodniy A.G., Zinoviev G.M., Martynov E.S., Svistunov S.Y., Shadura V.M. Grid – new information and computation technology for science // Bulletin of NASU. — 2005. — № 6. — P. 17-25. (in Ukrainian)
4. Zgurovsky M.Z. Development of educational and research segment of information society in Ukraine // Системні дослідження та інформаційні технології — 2006. — № 1. — С. 7-17.
5. Shelestov A.Ju., Korbakov M.B., Lobunets O.G. Implementation of Grid-infrastructure for satellite data processing // J. Problems of Programming. – 2006. - №2-3. – P. 94-101. ISSN1727-4907 (in Russian)
6. Building a European information capacity for environment and security. A contribution to the initial period of the GMES Action Plan (2002-2003) // Office for Official Publications of the European Communities (Luxembourg). — 2004. 238 p.
7. Global Earth Observation System of Systems (GEOSS), 10-Year Implementation Plan Reference Document // ESA Publication Division, Netherlands, 2005. — 209 p.
8. Zgurovsky M.Z., Pankratova N.D. System analysis: problems, methodology, applications, Kyiv, Naukova Dumka, 2005, 743 pages.
9. Holtman K. CMS Requirements for the Grid // Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP2001). — 2001.
10. Deelman E., Blackburn K., et al. GriPhyN and LIGO, Building a Virtual Data Grid for Gravitational Wave Scientists // Presented at 11th Intl. Symposium on High Performance Distributed Computing. — 2002.
11. Annis J., Y. Zhao et al. Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey // Technical Report GriPhyN-2002-05, 2002.
12. Peltier, S.T., et al. The Telescience Portal for Advanced Tomography Applications // Journal of Parallel and Distributed Computing: Computational Grid. — 2002. — 63(5). — P. 539-550.
13. Fusco L. Earth Science GRID on Demand // CEOS WGISS-21 GRID Task Team Meeting. — Budapest, May. — 2006.
14. Fusco L., Goncalves P., Linford J., Fulcoli M., Terracina A., D’Acunzo G. Putting Earth-Observation on the Grid // ESA Bulletin. — 2003. — 114. — P. 86-91.
15. GEONETCast, http://www.earthobservations.org/progress/GEONETCast.html.
16. Reis W.G. Remote Sensing Basics, Moscow: Technosphera, 2006, 336 pages. (in Russian)
17. Dissemination and Exploitation of GRids in Earth sciencE. — http://www.eu-degree.eu.

18. Popkov Yu.S. Macro-systems and Grid technologies: modelling dynamical stochastic networks // J. of Automation and Information Science. — 2003.— Num. 3.— P. 10-20. (in Russian)
19. Kussul N., Popov M., Shelestov A., Stankevich S., Korbakov M., Kravchenko O., Kozlova A. Information service for biodiversity assessment in Pre-Black Sea region in the framework of development of Ukrainian segment of GEOSS // J. Science and Innovations. — 2007. — Vol. 3, Num.6. — P. 17-29. (in Ukrainian)
20. Odum Yu. Ecology.- Vol.2.– Moscow: Mir, 1986.– 376 p. (in Russian)
21. Shelestov A.Yu. An object task model in satellite data processing Grid system// Transactions of Donetsk National Technical University. — 2007. — Issue 8(120). — P. 317-330.

Authors' Information

Andrii Yu. Shelestov *– PhD, Senior Researcher, Department of Space Information Technologies and Systems, Space Research Institute of NASU-NSAU, Glushkov Ave 40, Kyiv-187, 03680 Ukraine, e-mail: inform@ikd.kiev.ua.*

LEONTIEF MODEL ANALYSIS WITH FUZZY PARAMETERS BY BASIC MATRIXES METHOD

Vladimir Kudin, Grigoriy Kudin, Alexey Voloshin

Abstract: *The basic matrixes method is suggested for the Leontief model analysis (LM) with some of its components indistinctly given. LM can be construed as a forecast task of product's expenses-output on the basis of the known statistic information at indistinctly given several elements' meanings of technological matrix, restriction vector and variables' limits. Elements of technological matrix, right parts of restriction vector LM can occur as functions of some arguments. In this case the task's dynamic analog occurs. LM essential complication lies in inclusion of variables restriction and criterion function in it.*

Keywords: *Leontief model, quantitative and qualitative analysis, fuzzy set, basic matrix, membership function.*

Introduction

Mathematical apparatus of fuzzy sets is the way of indefinite parameters assigning, the values of which are unknown until the moment of decision-making. One of the mechanisms of vagueness removal in parameters assigning at model construction is the presence in the outline the decision-making person (DMP). DMP is aimed in workmanlike manner to determine the model's structure, to indicate the mechanism of vagueness removal at its formation [Orlovskij, 1981]. LM essential complication (LM) [Leontief, 1972], [Hass, 1961] is the inclusion of restrictions on variables' meanings (values) [Orlovskij, 1981]. One of the LM peculiarities is the inclusion of mathematical problems analysis series of linear systems as systems of linear algebraic equation (SLAE) with the quadratic nondegenerate matrix of restrictions, linear algebraic inequalities (SLAI), with the corresponding matrix of restrictions and also the tasks of linear programming (TLP) [Voloshin, 1993], [Vojnalovich, 1987], [Vojnalovich, 1988], [Kudin, 2002]. Realization of model's qualitative analysis [Orlovskij, 1981] predetermines as well the inclusion of quantitative analysis of its structural elements' consistency [Voloshin, 1993], [Vojnalovich, 1987], [Vojnalovich, 1988], [Kudin, 2002]. We can differentiate the following stages of analysis:

- testing of mathematical and computer-assisted non-degeneracy of restrictions matrix, determination of its rank's value;
- directing correction of restrictions matrix's rank's value with the means of changing its single elements (in case of necessity);
- revelation of LM common features itself and the restrictions on variables – solubility (insolubilities);

- determination of LM restriction's peculiarities for polyhedral set of relative constraints on variables, realization, in case of necessity, directed changes;
- solution finding at solvability by compatibility;
- solutions properties establishment.

Problem formulation

Let's examine LM variants, which one can achieve as the result of canonical model equivalent transformations:

1. (SLAE) of mode

$$Au = C, \tag{1}$$

2. (SLAI) of mode

$$Au \le C, \tag{2}$$

3. SLAI models (2) can be investigated in presence of aspect's criterion function

$$\max_{u \in R^m} \mathrm{Bu}, \tag{3}$$

as the task of linear programming model's analysis (2)-(3), in which $A = \left\{a_{ij}\right\}_{\substack{i=\overline{1,m} \\ j=\overline{1,m}}}$ is the quadratic undegenerate matrix with dimension $(m \times m)$, $a_j = (a_{j1}, a_{j2}, ..., a_{jm}),\ j \in J = I = \{1, 2, .., m\}$ – matrix's lines, A, $u = (u_1, u_2, ..., u_m)^T$ – variables' vector,, $B = (b_1, b_2, ..., b_m),\ C = (c_1, c_2, .., c_n)^T$ – vector of gradient criterion function and model's constraints, $a_j u \le c_j,\ j \in J$ half-space which is determined by hyperplane $a_j u = c_j,\ j \in J$.

The elements of constraints are considered as $a_l u \le c_l,\ l \in J$, normals of which occupies k th can undergo changes in accordance with the correlations $a_l(t)u \le c_l(t),\ l \in J$, where $a_l(t) = (a_{l1}(t), a_{l2}(t), \ldots, a_{lm}(t)),\ l \in J$ (element of model a_{lr} becomes equal $a_{lr}(t),\ r \in I$, and c_l will be $c_l(t)$). All certain functions that depend from the argument $t \in (-\infty, +\infty)$ of the class C^2. Such changes in model's elements can be construed as the price changes impact in the range $t \in (t_H, t_B)$ on the value of model's technological elements (2) by k th resource.

It is suggested that the system contains $P = \{1, 2, \ldots, p\}$ experts. Each expert forms its own membership function $\mu_p(t),\ p \in P$. These functions are piecewise linear, for which the expert determines levels of value $\lambda^{(p)},\ p \in P$. This means that at $1 \ge \mu_p(t) \ge \lambda^{(P)}$ is defined the range of value change $T_p,\ p \in P$, where $T_p = [t^{(-)}_{p(H)}, t^{(+)}_{p(B)}] \subseteq (-\infty, +\infty),\ p \in P$ [Orlovskij, 1981]. The resultant range variable's modification $T = (t_H, t_B)$ coordinated by P experts can be determined, e.g., as $T = \bigcap_{p=1}^{P} T_p$ (or $T = \bigcup_{p=1}^{P} T_p$). Model (1) is investigated in the area of E^m. At the presence in the outline the experts' decision making (DMP) phase of qualitative analysis of models (1), (2)-(3) defines a consequent task, as the task of quantitative analysis – investigations at given levels $\lambda^{(p)},\ p = \{1, 2, \ldots p\}$ indicated by experts of elements changes constraints' influence according to the correlation $a_l(t)u \le c_l(t),\ l \in J$ at $t \in T$ on an earlier chosen optimal decision.

In the work is suggested the development of sequential analysis' methodology [Voloshin, 1987] and the basic matrixes method (BMM) [Kudin, 2002] for realization of quantitative analysis of functional changes' influence in LM on its properties – like nonsingularity of constraints matrix, optimal decisions of the original problem (2)-(3) at model's elements changes (k th constraint) in the form of $a_l(t)u \le c_l(t),\ l \in J,\ t \in T$.

Main principles of Basic Matrixes Method (BMM)

In the suggested BMM are introduced horizontal basic matrixes [Vojnalovich, 1987], [Vojnalovich, 1988], [Kudin, 2002]. During interpretation of task's solution the basic matrixes are consecutively changing by leading in and out of her lines-perpendiculars of constraints. In the common case in the model under consideration the number of restrictions exceeds the number of variables mode (2), and in this case in LM $m = n$.

Definition 1. The matrix, which is made up from m linearly independent perpendiculars of constraints (2), we will consider to be basic, and the solution of corresponding to her system of equations $A_{\delta} u_0^T = C^0$ as well basic. Two basic matrixes that differ in one line are called adjacent.

Let: β_{ij}, $i, j \in I = \{1, 2, ..., m\}$ − elements of basic sub-matrix A_{δ}, e_{ri} − elements of matrix A_{δ}^{-1}, which is inverse to к A_{δ}; $e_k = (A_{\delta}^{-1})_k$.− column of inverse matrix. Solution $u_0 = (u_{01}, u_{02}, ..., u_{0m})$ of equations set $A_{\delta} u^T = c^0$, where, in general case, c^0 − subvector C, the components of which consist of right parts of constraints (2), forming normals basic matrix A_{δ}; $\alpha_r = (\alpha_{r1}, \alpha_{r2}, ..., \alpha_{rm})$ - expansion vector of normals of constraints $a_r u_1 \le c_r$ by the lines of basic matrix A_{δ}, $\alpha_0 = (\alpha_{01}, \alpha_{02}, ..., \alpha_{0m})$ - expansion vector of criterion function's gradient (3) by the lines of basic matrix A_{δ}, $\Delta_r = a_r u_0^T - c_r$ – excess of r[th] constraint (2) in the top u_0; J_{δ}, J_H, $J = J_{\delta} \cup J_H$ - an indexes' set of basic and nonbasic constraints (2). In the work [Vojnalovich, 1987] are given the connections formulas of basic solution, expansion coefficient of constraints normals and criterion function (3), inverse matrix's coefficients, discrepancies of constraints and criterion function's values at transfer to basic matrix $\overline{A}_{\delta}$, which is formed from matrix $\overline{A}_{\delta}$ with the change of its line a_k to a_l, that is not included into the basic matrix A_{δ}. The inserted quantities in the new basic matrix $\overline{A}_{\delta}$ we will call – elements of basic matrixes' method and will be marked with a sign-line above, i.e. $\overline{\beta}_{ij}$, $\overline{\alpha}_r$, $\overline{\Delta}_k$, $\overline{e}_{ri}$, $\overline{\alpha}_0$. Let $a_{i1}, a_{i2}, ..., a_{im}$ - are constraints normals, $a_j u^T \le c_j$, $j \in J_{\delta}$, where $J_{\delta} = \{i_1, i_2, ..., i_m\}$ - are constraints indices, the normals of which form the lines of basic matrix $A_{\delta,}$ a_l - is a normal of constraints $a_l u \le c_l$, $\alpha_l = (\alpha_{l1}, \alpha_{l2}, ..., \alpha_{lm})$ - – are the coefficients of vector's a_l expansion into lines of basic matrix $A_{\delta,}$.

Lemma 1. (linear independence criterion of vectors system). An essential and sufficient condition of matrix row's linear independence models $a_{i_1}, a_{i_2}, ..., a_{i_{k-1}}, a_l, a_{i_{k+1}}, ..., a_{i_m}$, which are formed by substitution of the row a_{i_k}, that occupies k[th] row in basic matrix A_{δ}, for row a $_l$, is the fulfillment of the condition $\alpha_{lk} \neq 0$.

Theorem 1. (About the connection between adjacent basic matrixes) Between the expansion coefficient of constraints normals (2) and criterion function (3) for the rows of basic matrix, elements of inverse matrix, basic solutions, discrepancy of constraints (2) and meanings of criterion function for two adjacent basic matrixes exist the following correlations

$$\overline{\alpha}_{rk} = \frac{\alpha_{rk}}{\alpha_{lk}}, \quad \overline{\alpha}_{ri} = \alpha_{ri} - \frac{\alpha_{rk}}{\alpha_{lk}} \alpha_{li,} \quad r = \overline{0, n}; \quad i = \overline{1, m}; \quad i \neq k; \tag{4}$$

$$\overline{e}_{rk} = \frac{e_{rk}}{\alpha_{lk}}, \quad \overline{e}_{ri} = e_{ri} - \frac{e_{rk}}{\alpha_{lk}} \alpha_{li,} \quad r = \overline{1, m}; \quad i = \overline{1, m}; \quad i \neq k; \tag{5}$$

$$\overline{u}_{0j} = u_{0j} - \frac{e_{jk}}{\alpha_{1k}} \Delta_l, \quad j = \overline{1, m}, \tag{6}$$

$$\overline{\Delta}_k = -\frac{\Delta_l}{\alpha_{lk}}, \quad \overline{\Delta}_r = \Delta_r - \frac{\alpha_{rk}}{\alpha_{lk}} \Delta_l, \quad r = \overline{1, n}; \quad r \neq k; \tag{7}$$

$$B\overline{u}_0 = Bu_0 - \frac{\alpha_{0k}}{\alpha_{lk}} \Delta_l, \tag{8}$$

moreover, the condition that the matrix remains basic at substitution by vector a_l k[th] row of basic matrix A_{δ}, is the fulfillment of condition $\alpha_{lk} \neq 0$, by the term of supporting basic solution permissibility is $\alpha_{lk} < 0$, the growth of criterion function's value $\alpha_{0k} < 0$.

The proof of lemma 1 and theorem 1 is based on theoretical statements that are stated at [Vojnalovich, 1987], [Vojnalovich, 1988], [Kudin, 2002].

Correlation (4)-(8) will be basic for the creation of search algorithm not only optimal solution but carrying out the analysis of LM properties with the means of basic matrixes method.

Definition 2. Feasible basic decision u_0 is optimal if $Bu_0 \geq Bu$ for all u that meet (2).

Theorem 2. For the basic decision is optimal u_0 is needed and not negativity of expansion coefficients of criterion function vector normal is sufficient (3) on the rows of basic matrix A_σ, i.e. $\alpha_{ok} \geq 0$ for all $k = \overline{1,m}$, moreover the task (2),(3) with the square nonsingular matrix of constraints has a unique solution only when $\alpha_{0i} > 0,\ i = \overline{1,m}$, while is needed and is sufficient the condition of not uniqueness of task's solution is $\exists\, i \in I$ of such, that $\alpha_{0i} = 0$, in which the solution set is not limited.
Optimal criterion validity arises from formula (8) theorem 1.

Consequence 1. At the variable's values, the function where $\hat{\alpha}_{lk}(t) = a_l(t)(A_b^{-1})_k \neq 0,\ t \in [t_0, t_k]$ is preserved unchangeable the rank quantity while changing the constraint k LM into constraint $a_l(t)u \leq c_l(t),\ l \in J$ where $t \in T$. The values t under which $\hat{\alpha}_{lk}(t) = a_l(t)(A_b^{-1})_k = 0,\ t \in [t_0, t_k]$, $l \notin J_\sigma$ diminish the system's rank (constraints matrix of model (2) becomes degenerated).
Consequence's validity arises from lemma 1.

Consequence 2. Sufficient condition of optimality preservation by the means of changing k constraint of model in the form of $a_l(t)u \leq c_l(t),\ l \in J_\sigma$ the normal of which occupies the k row in the basic matrix serves as fulfillment of correlations: $a_l(t)(A_\sigma^{-1})_k > 0$, $\frac{\alpha_{0k}}{\alpha_{lk}(t)} \geq 0$, $\frac{\alpha_{0i}}{\alpha_{0k}} \geq \frac{\alpha_{li}(t)}{\alpha_{lk}(t)}$. $t \in T,\ i \neq k,\ i \in I$. The condition of uniqueness of solution (2), (3) are $\frac{\alpha_{0k}}{\alpha_{lk}(t)} > 0,\ t \in T$, $\frac{\alpha_{0i}}{\alpha_{0k}} > \frac{\alpha_{li}(t)}{\alpha_{lk}(t)},\ t \in T,\ i \neq k,\ i \in I$.

Evidence. In accordance with lemma 1 and consequence 1 the condition of matrix's non-degeneracy model (2) is the implementation by all $t \in T$ the correlation $\alpha_{lk}(t) = a_l(t)(A_\sigma^{-1})_k \neq 0$. From formula (4) and theorem 2 follows that by substitution of k row of the basic matrix for the fulfillment of the optimality conditions is necessary to accomplish $\overline{\alpha}_{0k} = \frac{\alpha_{0k}}{\alpha_{lk}(t)} \geq 0$, $\overline{\alpha}_{0i} = \alpha_{0i} - \frac{\alpha_{0k}}{\alpha_{lk}(t)}\alpha_{li}(t) \geq 0,\ i \neq k,\ i \in I$, $t \in T$, as the optimality condition is $\alpha_{0i} \geq 0,\ i \in I$. From this it follows that the fulfillment of the condition $\alpha_{lk}(t) = a_l(t)(A_\sigma^{-1})_k \geq 0$ and $\frac{\alpha_{0i}}{\alpha_{0k}} \geq \frac{\alpha_{li}(t)}{\alpha_{lk}(t)}$, $t \in T$ for preservation of solution's optimality. From here is immediate, taking into account theorem 2, comes the validity of uniqueness (not uniqueness) conditions solutions (2)-(3).

Conclusions

The usage of simplex ideology on the basis of BMM at analysis of LM enables:

- to investigate the properties of solutions of SLAE and SLAI (1),(2) at changes in vectors of constraints;
- to carry out the analysis of LM properties by changing the values of separate elements and its components;
- to use the solution of basic LM at analysis of perturbed model;
- to control or directly to change the value system's rank;
- to discover a solution for the square system of equation by a fixed number of steps;
- to construct first task's solutions on the basis of trivial basic matrixes which exclude laborious pioneering calculations;
- to apply the diagram for tasks analysis which presuppose multistepness or recurrence of calculations on models by changes in model's components.

Bibliography

[Леонтьев, 1972] Леонтьев В.В., Форд Д. Межотраслевой анализ воздействия структуры экономики на окружающую среду// Экономика и математические методы.- 1972.-Т.VII.-Вып.3.-С.370-400.

[Гасс, 1961] Гасс С. Линейное программирование. Физматгиз,-1961.

[Волошин, 1987] Волошин А.Ф. Метод локализации области оптимума в задачах математического программирования // Докл. АН СССР. - 1987. -293, N 3.- С. 549-553.

[Орловский, 1981] Орловский С.А Принятие решения при нечёткой исходной информации.- М:, Наука,- 1981,- 206с.

[Волошин,1993] Волошин А.Ф. Войналович В.М., Кудин В.И. Предоптимизационые и оптимизационные схемы сокращения размерности задачи линейного программирования // Автоматика,N4, 1993.

[Войналович, 1987] Волкович В.Л., Войналович В.М., Кудин В.И. Релаксационная схема строчного симплекс метода // Автоматика.- 1987. -N4.-С. 79-86.

[Войналович, 1988] Волкович В.Л., Войналович В.М., Кудин В.И. Релаксационная схема двойственного строчного симплекс метода // Автоматика.-1988. -N 1,-С.39-46.

[Кудин, 2002] Кудин В.И. Применение метода базисных матриц при исследовании свойств линейной системы // Вестник Киевского университета. Серия физ.-мат. науки. - 2002.-2., С. 56-61.

Authors' information

Kudin V.I. *– National Taras Shevchenko University of Kiev, Cybernetics department, Ph. D. Kiev, Ukraine. E-mail: V_I_Kudin@mail.ru*

Kudin G.I. *– National Taras Shevchenko University of Kiev, Cybernetics department, Ph. D.i Kiev, Ukraine. E-mail: kuding@mail.univ.kiev.ua.*

Voloshin O.F. *– National Taras Shevchenko University of Kiev, Cybernetics department, professor. Kiev, Ukraine. E-mail: ovoloshin@unicyb.kiev.ua,*

APPROXIMATION OF EXPERIMENTAL DATA BY BEZIER CURVES

Vitaliy Vishnevskey, Vladimir Kalmykov, Tatyana Romanenko

Abstract: *Very often the experimental data are the realization of the process, fully determined by some unknown function, being distorted by hindrances. Treatment and experimental data analysis are substantially facilitated, if these data to represent as analytical expression. The experimental data processing algorithm and the example of using this algorithm for spectrographic analysis of oncologic preparations of blood is represented in this article.*

Keywords*: graphics, experimental data, Besie's curves*

ACM Classification Keywords*: I.4 Image processing and computer vision - Approximate methods*

Introduction

The experimental data, as a rule, represent the distorted by hindrances certain process fully determined by some unknown function *y= f(x)*, and distorted by hindrances. In most cases experimental data are represented as the graphical curves. The graphs, i.e. graphical curves is, apparently, the simplest and a long ago in-use means of cognitive presentation of experimental data in the most different scopes of human activity which allow to estimate evidently the qualitative property of the process, in spite of hindrances, measurement errors. Graphs displaying the same process, description of certain object can substantially differ

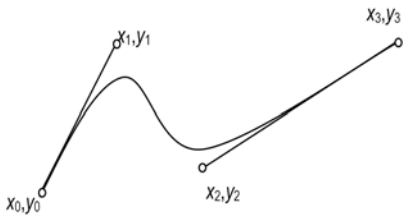


Fig. 1. Bezier curve

from each other by scales, amount of the used observations, level noises, and so on. At the same time the form feature of the graphical curve characterizes the parameters of the displayed object or process.

Automatic (automated) graph (graphical curve) processing supposes comparison of their forms to set, whether the different graphs characterize the same or different processes or objects. The solving of this task, using the neuronet methods or statistical visual patterns recognition methods, is complicated because of every graph can differ from other by scales, noises level, amount of the used measurements. At the same time the method [Kussul, 2004] of neuronet recognition of contours of the images represented with the Bezier curves is known. Using Bezier curves, it is possible to provide the invariance of descriptions of image contours in relation to position of images in eyeshot, scale, turn. So, the experimental data – graphical curves may be replaced with the analytical curves that are congruent enough to the initial graph form, and, at the same time, invariant relatively to the scale change, amount of measurements, noises level. Such curves may be chosen as the polynomial functions *x(t)*, *y(t)* of *n* power of the parameter *t* :

$$x(t) = a_0 t^n + a_1 t^{n-1} + \ldots + a_{n-1} t + a_n,$$
$$y(t) = b_0 t^n + b_1 t^{n-1} + \ldots + b_{n-1} t + b_n. \quad (1)$$

Choice of *n* – the power of polynomials depends on complication of the experimental curves to be approximated.

If such approach would be realized it is possible to hold all the substantial form features of the experimental curve and to eliminate the hindrances influencing. In addition, instead of the graph description in the space of initial signals it is possible to operate approximating polynomials in the space of their coefficients. The substantial advantage of such description is its invariance relatively the amount of the measured signals in every graph.

However, if the direct task to build the curve, having the set coefficients of polynomials is trivial, the reverse task – the set coefficients definition of approximating polynomials having the points of the graph – does not have the simple decision [Vishnevskey, 2004]. One of this task decisions is proposed at the article.

Formal problem definition

The graph can be represented by a polyline, the kinks of which form the sequence *of N* experimental values of some function *y(x): $y_Э(x)$={$y(x_0)$,$y(x_1)$,$y(x_2)$,…, $y(x_N)$}*, at the predetermined values $x_0,x_1,x_2,\ldots, x_N$. Let us consider the Bezier curve as an approximating curve because of the most often using in practical applications, parametrically defined polynomial of the third degree.

Under the Bezier curve we understand the arc of flat curve of the third degree [Deniskin, 1999]:

$$Bz(t) = \sum_{j=0}^{3} Br_j(t) \cdot Q_j \quad (2)$$

where $Br_j(t)$ are base Bernstein polynomials of the third degree,

Q_j are the coefficients of curve

t is an real parameter which changes in an interval [0,1].

Bernstein polynomials are defined as:

$$Br_j(t) = C_3^j \cdot (1-t)^{3-j} t^j, \quad 0 \le j \le 3 \quad (3)$$

where $C_3^j, \ 0 \le j \le 3$ are binomials coefficients

$$C_3^j = \frac{3!}{j! \cdot (3-j)!} \quad (4)$$

Thus, the Bezier curve (fig. 1) can be set by two polynomials of the third degree, depending on the parameter *t*:

$$X(t) = x_0 \cdot (1-t)^3 + x_1 \cdot (1-t)^2 \cdot t + x_2 \cdot (1-t) \cdot t^2 + x_3 \cdot t^3$$
$$Y(t) = y_0 \cdot (1-t)^3 + y_1 \cdot (1-t)^2 \cdot t + y_2 \cdot (1-t) \cdot t^2 + y_3 \cdot t^3 \quad (5)$$

The experimental data approximating as Bezier curve has many of advantages. Describing the wide enough class of curves, Bezier curves use reasonable quantity of coefficients - 8. Coefficients (x_0,y_0), (x_3,y_3) are the co-ordinates of initial and eventual points of approximating curve arc, accordingly, and coefficients (x_1,y_1), (x_2,y_2) - co-ordinates of points which control its form (so-called "whiskers"). Changing coefficients or moving points proper

to them on the screen of monitor (both initial and eventual and controls), an user can edit the form of approximating curve in the interactive mode, improving the results of automatic approximation.

Thus, every graph may be represented as a vector $\boldsymbol{v} = \{ x_0,y_0, x_1,y_1, x_2,y_2, x_3,y_3\}$, and its components are the coefficients of Bezier curve equations. The graph description as the vector ***v,*** always has the same amount of components regardless of number of measured experimental points, that makes solving of different tasks, in particular, tasks of the graphs recognition more simple.

It was before noticed, that determination of coefficients of Bezier curve, most exactly approximating one or another graph, is not a trivial task. The iterative method to minimize the sum of distances squares between every point of the graph and the approximating curve is used for determination of Bezier curve coefficients, approximating some experimental sequence of points [2]. Such problem definition follows from presumption, that graphs under consideration are arbitrary, rather than just those which display some unknown function. It is assumed that the graph is the aggregate of unconnected, mutually independent points. To calculate the distance from every point of the graph to the approximating curve proper nearest point on the approximating curve must be found. This task also is iterative as the approximating curve is parametrically defined. When the quantity of points on the graph is large (more than 100), and the amount of iterations is not limited beforehand, such method of task solving resulted in substantial calculable difficulties.

It is proposed to accept the area of the $S(\boldsymbol{v},y(x))$ figure, limited by the contour, formed with the approximating and experimental curves as the measure of likeness of this experimental graph with the approximating curve. So the values of Bezier curve coefficients $\boldsymbol{v}_{opt}$, which approximates certain experimental graph in the best way, correspond to the minimum value $S(\boldsymbol{v},y_Э(x))$:

$$v_{opt} = \arg\min_{v} S(v, y_Э(x)) \tag{6}$$

Algorithm

It is follows from formula 6, that computation of the area, limited by the contour $S(\boldsymbol{v},y_Э(x))$, is basic operation of algorithm, searching the optimum coefficient values $\boldsymbol{v}_{opt}$. The direct calculation $S(\boldsymbol{v},y_Э(x))$ is not possible because of this contour is not simply connected.

Graph $y_Э(x)$ and Bezier curve intersect each other many times. It is difficult to compute the intersections co-ordinates because of Bezier curve is parametrically defined. The simplest decision seems to replace Bezier curve by the polyline *Vt*. The fracture points of the polyline (x_t,y_t) belong to that Bezier curve and are calculated for the $t = 0, \delta, 2\delta, 3\delta,...,1$. The value δ is chosen small enough, to represent initial curve by the polyline *Vt* exact enough for the practical calculations. Possibility to calculate for every value $y(x_n)$ the proper value $V_t(x_n)$ is the result of the accepted simplification. Then the area $S(\boldsymbol{v}, y_Э(x))$ can be calculated as

$$S(v, y_Э(x)) = \sum_{n=0}^{N} \left| V_t(x_n) - y_Э(x_n) \right| \tag{7}$$

The value $\boldsymbol{v}_{opt}$, proper to the minimum value $S(\boldsymbol{v},D(n))$, can be got using the gradient method.

Experimental verification of algorithm

The offered algorithm was experimentally tested more than on 1100 real graphs of spectrographs of the medical preparations of blood, got on the method of Oncotest [Vishnevskey, 2006]. The method of early diagnostics of oncology diseases allows defining patient's presence or absence of oncology disease by the use of spectrograms of medical preparations of blood (Fig.2). At present researches are conducted also and for determination of localization of disease when being it. On fig.3 the spectrograms of preparations of patient's blood are represented. Their diagnoses are the oncology disease of identical localization (stomach). In spite of obvious distinctions of spectrograms on the levels of signals, hindrances, form of curves have undoubted likeness. The form of curves is specific property of the resulted graphs, which can correspond to the diagnosis. The identification task to form a diagnosis by spectrograms supposes additional researches which must be based on treatment of plenty of experimental information. In the same queue, the processing of large data arrays is

impossible without the decision of tasks of their computer treatment, and including the tasks of automatic or automated classification of spectrograms on their form must be decided, using the methods of patterns recognition.

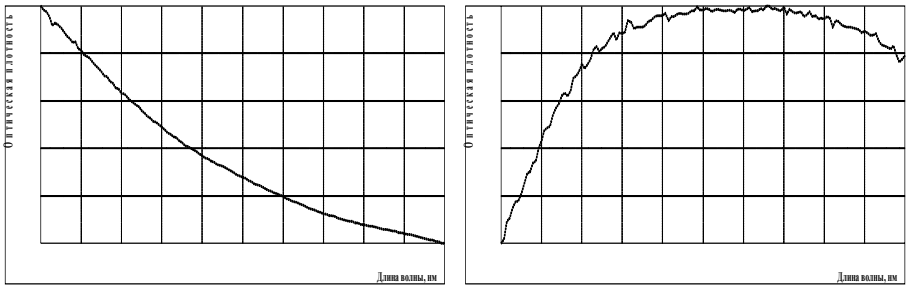


Fig.2. Examples of spectrograms of blood serum preparation:
left - spectrogram of healthy, right - sick man.

The direct use of statistical pattern recognition methods for the automated diagnosis forming in this case is not possible owing to large changeability of scale (dozens of one times), and, also, configuration of function of realizing the same appearance. Therefore it is appeared expedient to find some suitable Bezier curve for every graph, which polynomial coefficients, after adduction to one scale, can be used as pattern parameters, invariant to the changes of scale, amount of measures, noises level.

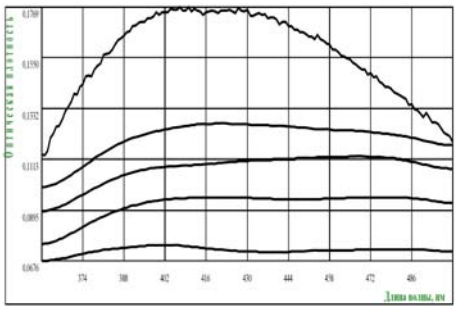


Fig.3 Examples of spectrograms of blood preparation of patients by the oncology disease of the same localization (stomach).

Examples of spectrograms and approximating them Bezier curves are represented on a fig. 4,5. A result was achieved in most cases for 15 - 40 iterations (in especially difficult cases - to 300). As it is obvious from figures, approximating the Bezier curves give the satisfactory approaching.

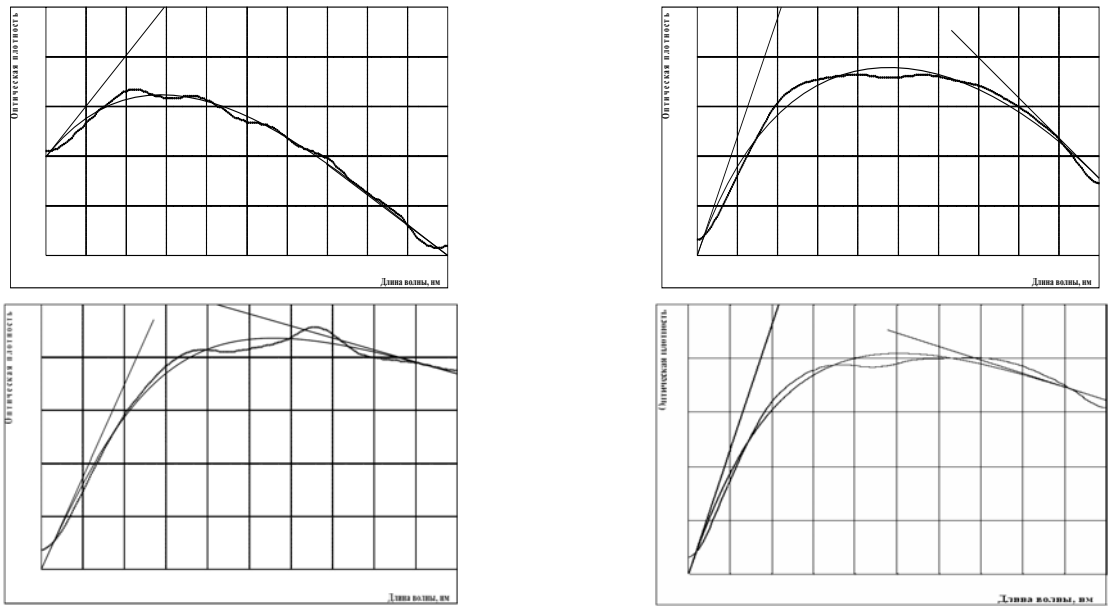


Fig.4. Spectrograms of blood preparations and their approximation by Bezier curves;
localization is a mammary gland

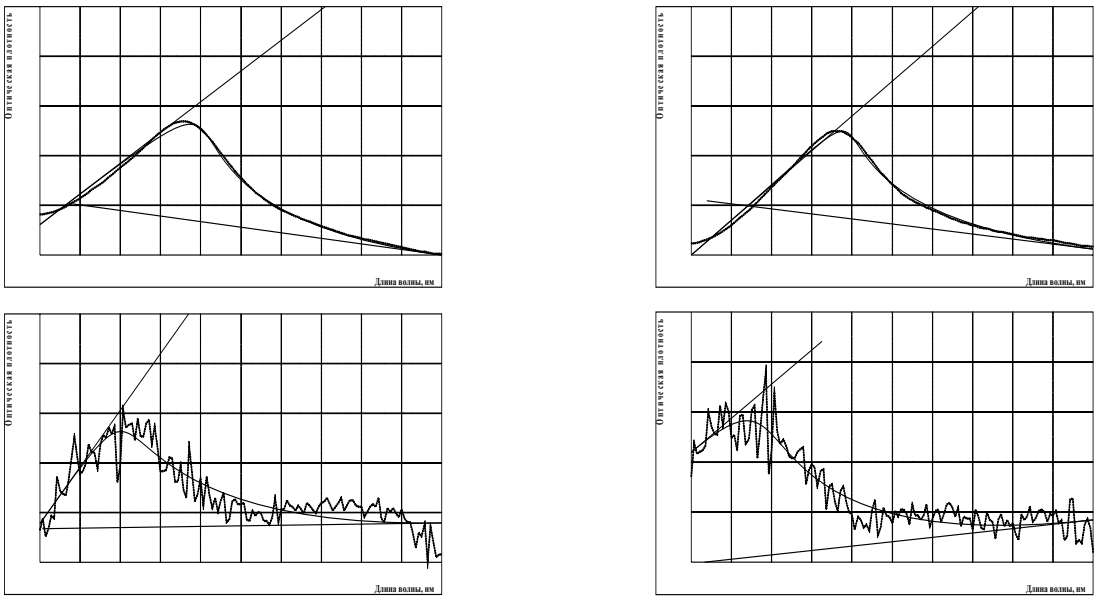


Fig.5. Spectrograms of blood preparations and their approximation by the Bezier curves; localization is a liver

Conclusion

The offered algorithm allows finding the parameters of Bezier curve, which approximates the experimental graph. The examples of approximation of spectrograms are resulted for implementation of the automated diagnostics. It will enable in the future to automatize the process of screening examination of population with the purpose of early detection of malignant tumors.

Bibliography

[Kussul, 2004] Куссуль М.Э. Кодирование контуров, представленных кривыми Безье, в задачах нейросетевой классификации / // Математические машины и системы. – 2004. – № 3. – С.17-30.

[[Vishnevskey, 2004] Вишневский В.В., Рысцов І.К., Волжева М.В Итерационный алгоритм построения кривой Безье по заданными точками //Математические машины и системы. - 2004 №4 - С. 108 – 116.

[Deniskin, 1999] Денискин Ю.И. Особенности аппроксимации обводов параметрическими полиномами в форме Бернштейна //Прикладная геометрия. - 1999. - вып. 2 № 2

[Vishnevskey, 2006] Вишневский В.В., Владимиров В.А., Романенко Т.Н. Программно-аппаратный комплекс "Онкотест-WM-01" // Український журнал телемедицини та медичної телематики. – 2006. – т. 4 №2 С.182-185

Authors' Information

Vitaliy Vishnevskey - *the head of the department, senior research worker, doctor of sciences, Institute of Mathematical Mashines and Systems, 42 Academician Glushkov St., 03680, Kiev 187, Ukraine; e-mail: vit@immsp.kiev.ua*

Vladimir Kalmykov - *senior research worker, doctor of sciences, Institute of Mathematical Mashines and Systems, 42 Academician Glushkov St., 03680, Kiev 187, Ukraine; e-mail: kvg@immsp.kiev.ua*

Tatyana Romanenko - *junior research worker, Institute of Mathematical Mashines and Systems, 42 Academician Glushkov St., 03680, Kiev 187, Ukraine; e-mail: romanenko@immsp.kiev.ua*

VECTOR COMBINATORIAL PROBLEMS IN A SPACE OF COMBINATIONS WITH LINEAR FRACTIONAL FUNCTIONS OF CRITERIA

Natalia Semenova, Lyudmyla Kolechkina, Alla Nagirna

Abstract: *The paper considers vector discrete optimization problem with linear fractional functions of criteria on a feasible set that has combinatorial properties of combinations. Structural properties of a feasible solution domain and of Pareto–optimal (efficient), weakly efficient, strictly efficient solution sets are examined. A relation between vector optimization problems on a combinatorial set of combinations and on a continuous feasible set is determined. One possible approach is proposed in order to solve a multicriteria combinatorial problem with linear-fractional functions of criteria on a set of combinations.*

Keywords*: vector optimization, discrete optimization, linear fractional functions, set of combinations.*

ACM Classification Keywords*: G 2.1 Combinatorics (F2.2), G 1.6 Optimization*

Introduction

A large number of management, planning, designing and other problems are solved with the use of combinatorial optimization models. Such a situation resulted in appearance of variety works devoted to research of combinatorial optimization problems [1,2,4]. Most of the practical decision-making problems are multicriteria. Therefore, it is very much interesting to examine different aspects of solution to multiobjective optimization problems, in particular, to discrete problems [1-8]. The interest in researching a solution to multicriteria discrete optimization models is due to their wide application when important economy, management, ecology, planning problems are dealt with, when different complicated systems are designed, decisions are made under uncertainty and so on. Vector optimization problems emerge if one needs to optimize more than single objective function. Fractional linear objective functions, being the result of division of two linear forms, are widely applied within the problems when it is necessary to optimize some relative indices of quality like a cost price, profitability, productivity and laboriousness. The models using such criteria show the tendencies in continuous lowering of a cost price level as for a unit of products and in increasing of high-quality production indices when production scale is broader. The union of the mentioned problems leads to investigation of multicriteria problems on combinatorial sets. The same problems are complicated each one by itself and they were not investigated earlier when taken totally, and, therefore, to it is quite important and urgent to solve them. It is interesting to consider the latter problems as the ones of combinatorial optimization on a feasible set of combinations.

Most of combinatorial optimization problems are known to be reduced to integer programming problems, but such a case is not always justified since the possibility to take combinatorial properties of a problem into account is lost [1]. The monographs [2,4] show that the convex hull of a combination set is the combination polyhedron, in which the set of vertices $vert\Pi$ is equal to the combination set $C(A)$ of, i.e. $vert\Pi = vert(convC_A)$. Such a combination polyhedron property allows to reduce a solution to an initial problem to a problem solution on a continuous feasible set. Thus, it is quite important and urgent to investigate and construct the methods that solve multicriteria problems and that are based on examining convex hulls of such sets when combinatorial properties of feasible solution domain are taken into consideration, in particular, the same concerns combinations.

At present, there exist many multicriteria problem solution methods, but none of them in the existing form are applicable for combinatorial problems on combinations. That is why it is quite important for the authors to consider the possible approaches used for multicriteria problems based on the combinatorial set of combinations.

The present paper investigates the structural properties of a feasible solution domain and the ones of the sets of Pareto-optimal (efficient), strictly efficient and weakly efficient solutions.

The relation between vector optimization problems on a combinatorial set of combinations and on a continuous feasible set is established. To solve the combinatorial problems on combinations, one possible approach is proposed.

Problem statement. Basic definitions

To state the problem, we use the concept of the multiset A determined by the foundation $S(A)$ and the element multipleness $k(a)$ (the number of the same elements *a*).

Let the multiset $A=\{a_1,a_2,...,a_g\}$ be specified, assume $S(A)=\{e_1,e_2,...,e_n\}$ as its foundation, where $e_j \in R_1 \ \forall j \in N_n=\{1,2,...,n\}$, and the multipleness $k(a_j)=r_j, j \in J_n, r_1+r_2+...+r_n=g,$ is given.

Take an arbitrary $k \in N_g$. Call the ordered k - *selection* from the multiset A as the set

$$a=\left(a_{i_1},a_{i_2},...,a_{i_k}\right), \tag{1}$$

where $a_{i_j} \in A \ \ \forall i_j \in N_k, \ \forall j \in N_k, i_s \neq i_t,$ if $s \neq t \ \ \forall s \in N_k, \ \forall t \in N_k$.

Definition. [2] A set *E(*A), the elements of which are the $k-$ *selections* of type (1) from the multiset *A*, is called the Euclidean combinatorial set, if the conditions $(a' \neq a'') \Leftrightarrow (\exists\, j \in J_k : a'_j \neq a''_j)$ are satisfied for its arbitrary elements $a'=(a'_1,a'_2,...,a'_k)$, $a''=(a''_1,a''_2,...,a''_k)$, that is, the set $E(A)$ the following property: two elements of the set $E(A)$ differ from each other, if, regardless of other differences, they differ in the order of location of the characters that form them.

Let the multiset A be the set. The set of all the k -*selections* from the set $S(A)$ of the type $\left(a_{i_1},a_{i_2},...,a_{i_k}\right)$ is called the Euclidean set of the k -combinations without reiteration of n different real numbers, if the condition $e_{i_1}<e_{i_2}<...<e_{i_k}$ is satisfied. Denote such a combination set by $C_n^k(A)$.

The set of all the k -selections of type (1) from the multiset A is called the general combination set, if the condition $a_{i1} \le a_{i2} \le ... \le a_{ik}$ is satisfied.

The interest in Euclidean combinatorial sets is related to the possibility to consider them as the points of the space R^k, that is, with the possibility to immerse them into the arithmetic Euclidean space R^k. Let $E(A)$ be the combinatorial set and let its element be described by formula (1). The mapping $f: E(A) \to E_f(A) \subset R^k$ is called immersion of the set $E(A)$ into the arithmetic Euclidean space, if f brings the set $E(A)$ into one-to-one correspondence to the set $E_f(A) \subset R^k$ according to the rule: for $a=\left(a_{i_1}, \ ..., \ a_{i_k}\right) \in E(A)$, $x=(x_1...,x_k) \in E_f(A)$, have $x_j=a_{i_j}. \ \forall j \in N_k$

The convex hull for the points of the Euclidean combinatorial set of combinations with reiterations is the polyhedron Π of combinations with reiterations, and such a polyhedron is described the set of the inequalities

$$\begin{cases} e_1 \le x^1, \\ x^i \le x^{i+1}, \ \forall i \in N_{k-1}, \\ x^k \le e^k, \end{cases} \tag{2}$$

where e_i are the elements of the basis $S(A)$ for the multiset A.

Consider the multicriteria optimization problem defined on the combinatorial set of combinations that is immersed into the arithmetic Euclidean space, and the problem has the following form:

$$Z_l(F,X): \ \max\{F(x) \mid x \in X\},$$

where *F(x)* is the vector criterion, $F(x)=(f_1(x),f_2(x),...,f_l(x)),$

$$f_i(x) = \frac{\left\langle c^i, x \right\rangle + c_0^i}{\left\langle d^i, x \right\rangle + d_0^i},\ i \in N_l, \tag{3}$$

$X = \Pi \cap D$, where $\Pi = conv\ S_{gn}^k(A)$, $D = \{x \in R^k \mid Ax \le b\}$, D *is* the convex polyhedron in R^k , $A \in R^{m \times k}$, $b \in R^m$.

The solution to the problem $Z_l(F, X)$ is understood as searching for the elements from one of the following sets: $P(F, X)$, i.e. the Pareto-optimal (efficient) solution set, *Sl(F,X)*, i.e. the Slater-optimal (weakly efficient) solution set, $Sm(F, X)$, i.e. the Smale-optimal (strictly efficient) solution set. According to [3, 6], for any $x \in X$, the following statements are true:

$$x \in Sl(F, X) \Leftrightarrow \{y \in X \mid F(y) > F(x)\} = \varnothing, \tag{4}$$

$$x \in P(F, X) \Leftrightarrow \{y \in X \mid F(y) \ge F(x), F(y) \ne F(x)\} = \varnothing, \tag{5}$$

$$x \in Sm(F, X) \Leftrightarrow \{y \in X \mid y \ne x, F(y) \ge F(x)\} = \varnothing. \tag{6}$$

Evidently,

$$Sm(F, X) \subset P(F, X) \subset Sl(F, X). \tag{7}$$

The finiteness of the feasible domain X yields the non-emptiness of the set $P(F, X)$ and its external stability that means that $\forall y \in X\ \exists x \in P(F, X) : F(x) \ge F(y)$.

Take the combinatorial properties of the combinations polyhedron into account, and the criteria for its vertex, face and adjacency of vertex are formulated in [2], that are expedient to be used for the problem solution method construction.

Vertex criterion. The points $x^i = \left(x_1^i, x_2^i, ..., x_k^i\right) \in R^k$, where $x_j^i = e_j\ \forall j \in N_{k-i+1}$, and $x_{j+k-i+1}^i = e_n$ $\forall j \in N_{i-1}$, $\forall i \in N_{k+1}$, and only they can be the vertices of a polyhedron of combinations without reiterations.

Criterion face. The set of points $x^i = \left(x_1^i, x_2^i, ..., x_k^i\right) \in R^k$ is i-face of polyhedron Π then only then, when it is the decision of the system of arbitrary $k - i$ equalizations, $i \in N_{k-1}^0$ from a set:

$$x_1^i = e_1, x_j^i = x_{j+1}^i, \forall_j \in N_{k-1}, x_k^i = e_n.$$

Adjacency of vertex. Arbitrary two vertices $x^i, x^j \in \Pi, i, j \in N_{k+1}$ are contiguous.

Properties of sets of Pareto of optimal (efficient), strictly efficient, weakly efficient solutions

Theorem 1. The elements of $Sm(F, X)$, i.e. the sets of strictly efficient solutions, $P(F, X)$, i.e. the sets of Pareto-optimal solutions, $Sl(F, X)$, i.e. the sets of weakly efficient solutions to multicriteria problems on the combinatorial sets of combinations are at the vertices of the combination polyhedron Π .

Proof. Consider relation (6) between the defined efficient solution set, take into account the fact that the feasible solution set X is the subset of the combination set $C_{gn}^k(A)$, and the following inclusions are true: $Sm(F, X) \subset P(F, X) \subset Sl(F, X) \subset C_{gn}^k(A)$. In accordance with Theorem 2.4 in [2], the combination set $C_{gn}^k(A)$ coincides with the set of vertices of general combination polyhedron $C_{gn}^k(A) = vert\Pi_{gn}^k(A)$. Therefore, the inclusions $Sm(F, X) \subset P(F, X) \subset Sl(F, X) \subset vert\Pi_{gn}^k(A)$ are true. Theorem is proved.

The present property the combination polyhedron allows reducing the solution to the problem $Z_l(F,X)$ to the solution to the problem $Z_l(F,G)$ on the continuous feasible set $G=\Pi\cap D$. For any point $y\in vert\Pi$ in such a problem, the sufficient optimality conditions [3] for all the mentioned types of the efficient solutions are true.
The following theorems are valid.

Theorem 2. $Sl(F,G)\bigcap vert\Pi\subset Sl(F,X),\ P(F,G)\bigcap vert\Pi\subset P(F,X),\ Sm(F,G)\bigcap vert\Pi\subset Sm(F,X).$

Proof. Since $\text{vert}\,\Pi\bigcap D\subset G,$ the inclusion $Sl(F,G)\bigcap \text{vert}\,\Pi\bigcap D\subset Sl(F,G\bigcap \text{vert}\,\Pi\bigcap D)=Sl(F,X)$ is true.

The relations $P(F,X)=P(F,\text{vert}\,\Pi\bigcap D)\supset P(F,G)\bigcap \text{vert}\,\Pi\bigcap D,$
$Sm(F,X)=Sm(F,D\bigcap \text{vert}\,\Pi)\supset Sm(F,G)\bigcap \text{vert}\,\Pi$ are roved by analogy.

Theorem 3. $\forall x\in \text{vert}\,\Pi: x\in P(F,\Pi)\bigcap D\Rightarrow x\in P(F,X),\quad x\in Sl(F,\Pi)\bigcap D\Rightarrow x\in Sl(F,X)$,
$x\in Sm(F,\Pi)\bigcap D\Rightarrow x\in Sm(F,X)$.

Proof. Since $G=\Pi\bigcap D$, then the following implications are true:
$\forall x\in \text{vert}\,\Pi: x\in P(F,\Pi)\bigcap D\Rightarrow x\in P(F,\Pi\bigcap D)=P(F,G)\Rightarrow x\in P(F,X),$
$x\in Sl(F,\Pi)\bigcap D\Rightarrow x\in Sl(F,X)$, $x\in Sm(F,\Pi)\bigcap D\Rightarrow x\in Sm(F,X)$.

Therefore, Theorems 1-3 establish the relation between the problems $Z_l(F,X)$ and $Z_l(F,G)$, and the latter one is defined on the continuous feasible set. It becomes possible to apply the classical continuous optimization methods for solutions to vector combinatorial problems on combinations. One can develop the new original solution methods using the properties of combinatorial sets and their convex hulls.

As for the strictly quasi-convex objective criterion function, the following theorem is proved in [5].

Theorem 4. If the vector criterion functions $f_i(x),\ i\in N_l$, are strictly quasi-convex and semicontinuous from below on linear segments *X*, then the set *Sl(F,X)* of the weakly efficient solutions to the problem is the union of the efficient sets $P(F,X)$ of the solutions to the subproblems $Z_I(F,X), I\subset N_l, I\neq\varnothing,$ i.e. $Sl(F,X)=\bigcup\{P_l(F,X): I\subset N_l,|I|\leq k+1\}$.

The linear fractional functions are known to be strictly quasi-convex on a convex set, and, therefore, Theorem 4 is valid for the set $Sl(F,X)$ of weakly efficient solutions to the problem *Zl(F,X)*.

Let x^0 be some efficient (Pareto-optimal) solution. Introduce the following denotations:

$P=(p_1,p_2,...,p_l)\in R^{k\times l},\qquad Q=(q_1,q_2,...,q_l)\in R^{k\times l},\qquad p_i=\left(\left\langle d_i,x^0\right\rangle c_i-\left\langle c_i,x^0\right\rangle d_i\right)\in R^k, i\in L,$

$q_i=\left(d_i^0c_i-c_i^0d_i\right)\in R^k, i\in L, y\in R^l, \alpha>0,$ and $\alpha\in R^l$ is some continuous vector, consider the linear programming (LP) of such a form:

(LP): $\max\left\{\langle\alpha,y\rangle\,|\,Q^Tx^0-(P+Q)^Tx+y=0, x\in\Pi\bigcap D\right\}.$

For the problem $Z_l(F,X)$, the following theorem is valid that establishes the interrelations between the solutions to it and the ones to the LP problem.

Theorem 5. The point $x^0\in X$ is the efficient solution to the problem $Z_l(F,X)$ if and only if the LP problem has the optimal solution (x^*,y^*) with $y^*=0.$

There exist a number of the methods used to solve multicriteria problems that allow to find the elements of the indicated of the indicated solution sets. But the problem $Z_l(F,X)$ is becomes more complicated in the presence of the criteria that are the linear fractional functions of type (2). This is the first specific feature of the aforesaid.

The second feature of the problem $Z_l(F,X)$ is the presence of the combinatorial properties that the feasible solution domain has. Therefore, to solve the problem, it is necessary to take all its features into account.

As it is generally known, to solve problems with a linear fractional goal function, there exist many methods conventionally falling into linearization methods, parametric methods, simplex-method modifications, among which one knows Charnse and Kuper method, Gilmori and Gomori algorithm and others. But none of them takes combinatorial terms of feasible solution domain into account.

On the basis of the proved theorems, continuing and developing works [2-5] one possible approach to the considered problem solution is proposed that means to implement the followings stages:

1) at the initial stage, consider some problem with a linear fractional objective function on a combination set;
2) for a separately taken problem, go by turns from a linear fractional objective function to a linear function;
3) a multicriteria problem with linear objective functions is examined, but on a changed feasible solution set;
4) a transition is carried out a multicriteria problem with linear objective functions to the the same thingcriterion with a linear objective function.

Consider the second stage of the proposed approach in more detail. Tho lolve the problem *Z(F,X)*, pass to the problems with linear objective functions. Use the denotations:

$$y_0 = \frac{1}{\left\langle d^i, x \right\rangle + d_0}, \; z_i = x_i y_0, \qquad \forall i \in N_k. \tag{8}$$

Relation (8) specifies the mapping $\psi(X) = X^t \subset R^{k+1}$ for the feasile solution set. If $\left\langle d^i, x \right\rangle + d_0 > 0$, then $y_0 > 0, x_i \geq 0, \forall i \in N_k$, and $\psi(x) = z = (y_0, z_1, ..., z_k)$.

In this case, the problem with a linear fractional objective function is reduced to the linear objective function

$$f'(\mathrm{x}^*) = f(\mathrm{z}^*) = \left\langle c^i, z \right\rangle, \tag{9}$$

under the condition

$$X' = \Pi' \cap D'. \tag{10}$$

It should be noted that, if is $z = (y_0, z_1, ..., z_k)$ is the solution to problem (9), (10) with a linear objective function, the latter meets condition (8), and $x = (x_1^*, ..., x_k^*)$ satisfies the constraints for the problem $Z_l(F,X)$.

Thus, on the general combination set, a combinatorial problem with a linear-fractional function can be reduced to a problem with a linear objective function. The number of variables is increased here by one, and to the system of limitations another equality is added: $\left\langle d^i, z \right\rangle + d_0 y_0^* = 1$.

Conclusions

The paper considers the statement of the multicriteria problem when the combinatorial properties of the feasible solution domain and linear-fractional functions of criteria are taken into account. The paper also establishes the interrelation between the problem with a linear objective function and the vector problem with linear-fractional functions of criteria defined on a feasible combinatorial set of combinations.

It becomes possible to apply the classical optimization methods for solutions to vector combinatorial problems on a combination set and, hence, to develop the new original solution methods when the properties of combinatorial sets and their convex hulls are used.

Proceeding from the proved theorems, the ongoing research and the results developed in [2-5], the approach is proposed that is used to solve the problem $Z_l(F,X)$. The basis for such an approach means to reduce searching for a solution to an initial problem to solving a number of scalar (single-criterion) problems and to verifying optimality of an obtained solution. The single-criterion problem solution methods rely on the ideas of decomposition, Kelly's cutting plane method, relaxation.

Bibliography

[1] Sergienko I.V., Kaspshitskaya M.F. Models and methods solving combinatorial optimization problems on computers. Kyiv, Naukova Dumka, 1981. 287 p. (In Russian)

[2] Stoyan Yu.G., Emets O.O. Theory and methods of Euclidean combinatorial optimization. Kyiv, Institute of System Research in Education, 1993.–188 p. (In Ukraine)

[3] Lebedeva T.T., Semenova N.V., Sergieko T.I. Conditions for optimality and solvability in the problems of linear vector optimization with a convex feasible set. In: Reports of the National Academy of Sciences of Ukraine, 2003, №10, p. 80-85 (In Ukrainian).

[4] Emets O.O., Kolechkina L.M. Combinatorial optimization problems with linear fractional objective functions. Kyiv, Naukova Dumka, 2005. 118p. (In Ukrainian)

[5] C. Malivert, N. Boissard. Structure of efficient sets for strictly quasiconvex objectives. J. of Convex Analysis, 1994, vol. 1, 2, p.143-150.

[6] Lebedeva T.T., Semenova N.V., Sergieko T.I. Stability of vector integer optimization problems: Connection with stability of optimal and nonoptimal solution sets. Kibernetika i Sistemny Analiz, 2005, №4, p. 90-100 (In Russian)

[7] Podinovsky V.V., Nogin V.D. Pareto-optimal solution to multiobjective problems. – M.: Nauka, 1982. 256p. (In Rassian)

[8] T.R. Gulati, M.A. Islam. Efficiency in Linear Fractional Vector Maximization Problems with Nonlinear Constrains // Optimization. – 1989. Vol. 20. - №4. – P.477-482.

Information about authors

Natalia V. Semenova – *V.M. Glushkov Institute of Cybernetics, National Academy of Science of Ukraine, Candidate of Physical & Mathematical Sciences, Senior Scientific Researcher, 40, Prospect Akademika Glushkova, 03187, Kyiv, Ukraine; e-mail: nvsemenova@meta.ua*

Lyudmyla M. Kolechkina – *V.M. Glushkov Institute of Cybernetics, National Academy of Science of Ukraine, Candidate of Physical & Mathematical Sciences, Doctoral Student, 40, Prospect Akademika Glushkova, 03187, Kyiv, Ukraine; e-mail: ludapl@ukr.net*

Alla M. Nagirna – *Ukrainian State University of Finances and Foreign Trade, Senior Teacher, Higher Mathematics and Informatics Department, 3, Chigorin Str., Kyiv; e-mail: vpn2006@rambler.ru*

ONE CLASS OF STOCHASTIC LOCAL SEARCH ALGORITHMS

Leonid Hulianytskyi, Alexander Turchin

Abstract*: Accelerated probabilistic modeling algorithms, presenting stochastic local search (SLS) technique, are considered. General algorithm scheme and specific combinatorial optimization method, using "golden section" rule (GS-method), are given. Convergence rates using Markov chains are received. An overview of current combinatorial optimization techniques is presented.*

Keywords*: combinatorial optimization, stochastic local search, simulated annealing, Markov chains*

Introduction

Approximate algorithms are well-known to solve different combinatorial problems. This based on some facts: firstly, mostly each problem is NP-hard; secondly, result functions have lots of local extremes; and finally, real data often are given with some inaccuracy and this makes serious calculations unnecessary. Also, it must be noticed that key ideas of these algorithms (metaheuristics) allow us to design algorithms, which can solve not one but some classes of optimization problems.

The most known definition of combinatorial optimization problem (according to Papadimitriou and Steiglitz [1,2]) is following: find $x^* \in X$

$$x_* = arg \min_{x \in D \subseteq X} f(x), \quad (1)$$

where X – finite (or possibly countably infinite) solution space, D – subspace defined by problem constraints, $f: X \to R^1$ – objective function.

This definition works perfectly on finite sets, but in case of infinite sets some classification problems nay appear – classification according to structure of solution space may be difficult. To get over these complications, Berge [3] introduced *combinatorial configuration*. Let m,n – natural numbers, $U=\{1, \dots, m\}$, $V=\{v_1, \dots, v_n\}$ – some sets, and some order is given for V: $v_1 < \dots < v_n$ (in other words, V is a chain).

Definition 1. *Combinatorial configuration* is a reflection $\varphi : U \to V$, satisfying some contingencies Λ.

From this definition, if m,n – some fixed numbers, the number of combinatorial configurations is finite. Usage of contingencies Λ allows describing different combinatorial configurations.

Berge's definition can be generalized as follows: Let $Y=\{1, \dots, m\}$, Z – some discrete (may be finite) space called "forming space", φ – homomorphism, $\varphi : Y \to Z$, satisfying some contingencies system Ω. It must be noted, that discrete space is the space of isolated points.

Definition 2. *Combinatorial object* is $\kappa = (\varphi, \tilde{X}, \Omega)$, where $\tilde{X}$ - basic space.

Definition 3. *1st order combinatorial object* is such combinatorial object, where basic space is the forming space:

$$\kappa = (\varphi, X_{(1)}, \Omega),$$

Here $X_{(1)} \equiv Z$. It is easy to see that in case of finite Z these combinatorial objects are the same with Berge's combinatorial configurations [3].

Definition 4. *k-order combinatorial object* is such combinatorial object that:

$$\kappa = (\varphi, X_{(k)}, \Omega),$$

where $X_{(k)} \subseteq X_{(k-1)} \cup X^k$.

After everything stated above, another definition of optimization problem (1) can be stated as follows:

Definition 5. Optimization problem (1) will be *combinatorial optimization problem (COP)*, if solution space X is the space of combinatorial objects.

Also, it must be noted that received results are valid for the case of discrete countable X.

Simulated Annealing (SA) method is well-known effective search technique solving COP [2]. SA algorithms allow finding high accuracy results and can be easily realized on multi-processor systems.

Transaction probability in SA scheme depends on result function changes and realized with parameter T (temperature). Probabilistic transactions (*temperature schedule*) in neighborhoods of worse solutions should decrease with the number of iterations and should tent to zero. Changes of parameter T are stated with *equilibrium conditions*. Such probabilistic mechanism is a key feature of SA scheme and allows avoiding local extremes during first iterations and concentrates on global extremes in time.

At the same time, different researches showed great dependence of received solutions from values of parameters. This fact leads to necessity of probabilistic mechanism generalization. As a result, a new class of *accelerated probabilistic modeling algorithms* (also called *G-algorithms*) was introduced [4], and among them – algorithm with specific probabilistic transaction scheme [5].

G-algorithms

The scheme of accelerated probabilistic modeling algorithms concludes one key idea: algorithm build next element from current solution neighborhood, and if result function of this element is better – we accept it as updated current solution, if result function is worse – element can be accepted with some probability. But, unlike

SA, in *G*-algorithm these "thresholds" are calculated in similar way during entire computational process, but algorithm changes the parameter that defines elimination of worse solution.

To build *G*-algorithm, we should define $\{\mu_t\}, 0 \le \mu_0 < \mu_1 < \ldots \le 1$ – some strictly monotonous sequence of real numbers, that can be describe as similarity of temperature schedule in SA. If x^h is a current solution on iteration *h*, and $L(x)$ is a neighborhood of $x \in X$, then next element $y \in L(x^h)$ with $f(y) > f(x^h)$ (in case of minimization problem) can be accepted as x^{h+1} with some probability $p(x^h,y)$, that depends on current value of μ_t.

Let $F(x,y)$, $0 \le F(x,y) \le 1$– some functional dependent on result function. Transaction probability from *x* to *y* ($x,y \in X$) can be defined in the next way:

$$p \equiv p(x,y) = (1-\mu_t) \cdot F(x,y)$$

The general scheme of G-algorithms presented on Picture 1.

```
procedure  G_Search(x)
begin
       x^0 := some initial solution from X;
       μ_0 : = 0;  h := 0;  t : = 0;
       x_rec:= x^0;  f_rec := f(x^0);
       while neighborhood of current solution L(x^h) is not checked totally do
           begin
               while equilibrium condition is not met  do
                  begin
                       y := GenerateNextNeighbor L(x^h);
                       Calculate F(x^h, y);
                       p := (1- μ_t ) F(x^h, y);
                       ξ  := random[0,1];
                       if  p ≥ ξ  then
h:=h+1;  x^h := y;
if  f_rec > f(x^h)  then
    x_rec := x^h;  f_rec := f (x^h)
end if
                                          end if;
                  end;
                 CalculateNextValue μ_{t+1};
                 t := t + 1;
            end;
          return   x= x_rec;
  end
```

Picture 1. Accelerated probabilistic modeling algorithm (*G*-algorithm)

Here *random*[0,1] – random generator of values from [0,1].

To build *G*-algorithm, these elements must be defined:

- functional $F(x,y)$;
- $\{\mu_t\}$ building mechanism;
- equilibrium conditions;
- stopping rule.

Functional $F(x,y)$ must use monotonous by result function consequences, that met such conditions:

a) $F(x,y) \to 1$, if $f(y) \to f(x)$.

b) $F(x,y) \to 0$, if $f(y) \to \infty$.

For example, next functional can be used as $F(x,y)$:

$$F(x,y)=\begin{cases}\left[\dfrac{f(x)}{f(y)}\right]^{\beta}, if\ \ f(y)\geq f(x),\\ 1, \qquad otherwise.\end{cases}$$

where β > 0 – some real, or natural number particularly.

Usually, it is easy to find the upper bound of result function f_{max} in terms of some COP, because of finite X: $f(x)\leq f_{\max},\ \forall x\in X$. In this case functional may be as follows:

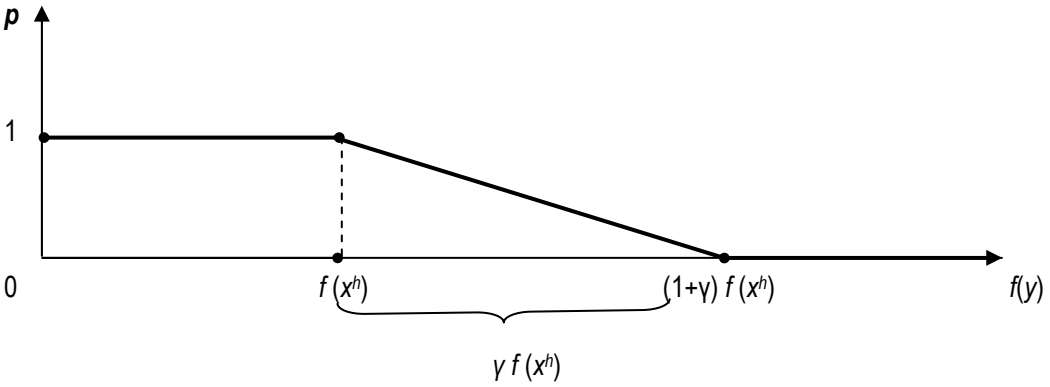
$$F(x,y)=1-\left[\frac{f(y)-f(x)}{f_{\max}-f(x)}\right]^{\beta}.$$

In general case, common G-algorithm uses next piecewise linear functional [5]:

$$F(x,y)=\begin{cases}\min\{1,1-\dfrac{f(y)-f(x)}{\gamma f(x)}\},\ if\ f(y)\geq f(x),\\ 0,\ if\ \ f(y)<f(x),\end{cases}$$

where parameter $\gamma,\gamma>0$ is a real number.

In this case, for current solution x^h transaction probability $p(x^h,y)$ from x^h to $y\in L(x^h)$ represented with piecewise linear functional, that consists from three main parts: it is 1 if $f(y) < f(x^h)$, it decreases from 1 to 0 if $f(x^h) \leq f(y) \leq (1+\gamma)\, f(x^h)$, and it is 0 when $f(y) > (1+\gamma)\, f(x^h)$. This can be illustrated in next way (Picture 2).



Picture 2. Transaction probability in the neighborhood

Thus, parameter γ defines the middle interval $\gamma\cdot f(x^h)$, and if $f(y)$ gets into this interval, y may be chosen as updated solution x^{h+1}. This means that in case of essentially worse f(y) the next neighborhood element y will be far away from acceptance. Also, it should be noted that the length of interval decreases if $f(x^h)$ gets closer to global optimum, and increases otherwise.

Moreover, usage of relative values allows avoiding the dependence from absolute values. Thereby, computational process adapts to dynamics of result function changes.

There are two main approaches to build consequence $\{\mu_t\}$: using some strictly monotonous function called "*G*-function" (that is why algorithms are called *G*-glgorithms [4,5]), and using "golden section" rule [6].

Let G: [0,1] → [0,1] – some strictly monotonous function, then $\mu_{t+1} = G(\mu_t)$. In common cases, G-functions are like this example:

$$G_k(x)=\min\{1,(x^{1/k}+H)^k\},\ \ x\geq 0. \qquad (2)$$

where $k\in\{1, 2, 3\}$, a H, $0 < H < 1$ – some small value defining the convergence rate $G_k(x)\to 1$ if $x\to 1$.

Otherwise, values μ_t may be calculated through division of [0,1] interval with some step and sequential selection of constructed points. Apparently, when consequence reaches 1, proposed algorithm will work like common determinate local search algorithm.

Usage of "golden section" rule became an effective approach to build algorithms for COP with 1-argument continuous result function. In this case μ_t is the left "golden section" point of $[\mu_{t-1},1]$; $\mu_0 = 0$, and $\mu_{t+1}=G(u_t)$. Thus, "golden section" rule defines the speed of left border approaching to 1, and corresponding μ_t are the arguments for $G(x)$ (for example (2)). Algorithm using "golden section" rule is called GS-algorithm.

Setting up the equilibrium condition may be effective with usage of SA experience, and analogy of "temperature" in SA and μ, in G-algorithms. Particularly, possible equilibrium condition can be defined as follows: let ν is some natural number and $\varepsilon > 0$ is real; and realization of ν transactions is called a "run" [7]. If, for current temperature algorithm made k runs and some consequence $f_1,...,f_k$ is received, equilibrium condition is met if

$$| f_{k+1} - f_i | \leq \varepsilon$$

for some $i \in \{1,..., k\}$. In this case f_i may be the average result function value or the best one, received on current run.

Algorithm stops when stopping rule is met. This can be: total check of neighborhood with no transactions done, algorithm computational time limitations, reaching some denoted solution accuracy (in case of known fesult function lower bound) etc. Also, another possible stopping rule may base on comparison of min and max result function values – and min and max result function values with some value of μ: if this correspondence tends to 1, algorithm stops.

In case of some contingencies computational scheme of G-algorithms can easy take these contingencies into account when next point of neighborhood is generated. This approach allows solving some COP classes, and its contingencies can be changed even during algorithm work.

Similarly to Iterated Local Search (ILS) [2], after finishing general GS-algorithm work the received solution can be modified and used as initial solution for built-in GA-algorithm. In other words, we get some metaheuristic method that can be called "Iterated GS-algorithm".

Convergence analysis

Convergence analysis will be made on essential class of COP – permutation tasks with transposition metric. Let $Gr = (V, H)$ will be the complete weighted graph with $V = \{1,...,n\}$ - set of nods and H - set of ribs. The path between two point from solution space X means the consequence of ribs: each nod relevant to $x \in X$, will be connected with nods relevant to $L(x)$ neighborhood elements. Let's consider neighborhoods of minimal radius 1: $L(x) = L_1(x)$. The correspondence between neighborhood $L(x)$ and set $N(v), v \in V$ will be built upon next rule: if two point in space X differs with one transposition, their nods-images on graph Gr will be connected with a rib.

The search process in this case will be imitated with transactions on graph Gr. The degree d of graph is $d = \frac{n(n-1)}{2}$, and diameter $D = n - 1$.

Search process in this case is being modeling with Markov chain because next state depends only on current state. GS-algorithm convergence if corresponding Markov chain will have at least one value corresponding to global optimum.

Transaction probability is limited above with value μ^s. Let $\hat{d} = \left(\frac{1}{d}\right)^{-D}$, $\hat{\mu} = \prod_{i=1}^{D} \mu^i$.

Lemma. Let $v \in V$ - some nod corresponding to some x from solution space. Then expected number of steps to reach the nod, corresponding to the global optimum, not more than $\frac{\hat{d}}{\hat{\mu}}$.

Proof. Let $\underline{x}$ – global optimum of COP (1), and let nod $\underline{v}$ in graph $Gr = (V, H)$ corresponds to $\underline{x}$. There is a path from v to $\underline{v}$ with length $q < D$. In other words, there is a consequence of nods $v_1, v_2, ...v_q$.

If each element of neighborhood can be selected for checking with the same probability, transaction probability of visiting $\underline{v}$ in *q* steps will be at least $\left(\frac{1}{d}\right)^q \times \prod_{i=1}^{q} \mu^i$, and will be valid

$$\left(\frac{1}{d}\right)^q \times \prod_{i=1}^{q} \mu^i \geq \left(\frac{1}{d}\right)^D \prod_{i=1}^{D} \mu^i = \frac{\hat{\mu}}{\hat{d}}.$$

Therefore, probability of visiting $\underline{v}$ starting from some v will be at least $\frac{\hat{\mu}}{\hat{d}}$. This means that expected number of algorithm steps to reach global optimum will be not more than $\frac{\hat{d}}{\hat{\mu}}$.

Lemma is proved.

Theorem. GS-algorithm convergence to the global optimum with probability more than $(1 - \frac{1}{C^k})$ during the number of steps not more than $Ck \cdot \frac{\hat{d}}{\hat{\mu}}$ and this estimation does not depends on initial solution ($C = const > 1$).

Proof. Mathematical induction approach will be used. Let $Q = C\left[\frac{\hat{\mu}}{\hat{d}}\right]$, and let's prove that probability of missing nod $\underline{v}$ in kQ steps is not more than $1/C^k$.

For basic case $k = 1$ and some initial state *x*, according to the Lemma the expected number of steps to reach $\underline{v}$ is not more than $\frac{\hat{d}}{\hat{\mu}}$. From Markov's inequality

$$p(\frac{\hat{d}}{\hat{\mu}} > Q) = \frac{M(\frac{\hat{d}}{\hat{\mu}})}{Q} = \frac{\frac{\hat{d}}{\hat{\mu}}}{C\frac{\hat{d}}{\hat{\mu}}} = \frac{1}{C}.$$

Let's estimate that theorem is valid for all $k \leq K - 1$ and prove it for $k = K$. Let $x_Q, x_{2Q}, ...x_{(K-1)Q}$ - Markov chain built on steps $Q, 2Q, ..., (K-1)Q$ correspondently. Consider two events:

- Event H1: nod $\underline{v}$ is not reached during first Q steps;
- Event H2: nod $\underline{v}$ is not reached during next $(K-1)Q$ steps;

According to these events, probability to miss $\underline{v}$ during KM steps is $p = p(H2 \mid H1) \times P(H1)$. From Markov chain theory, the probability to reach global optimum during some number of steps depends on this number of steps and does not depends on states visited previously. Using this fact, probability $P(H2 \mid H1)$ depends on state on iteration Q and $(K-1)Q$ steps:

$$P = P(H1) \sum_{i \in V} P(H2 \mid x_Q = i) \times P(x_Q = i).$$

Probability $P(H1)$ is not more than $\frac{1}{C}$ according to the basic case, probability $P(H2 \mid x_Q = i)$ is not more than $\frac{1}{C^{K-1}}$ for each $i \in V$ according to the estimation. Finally:

$$P \le \frac{1}{C} \times \frac{1}{C^{K-1}} = \frac{1}{C^K}.$$

Probability to reach global optimum is opposite to miss it. Therefore, with probability more than $(1 - \frac{1}{C^k})$ will be reached during the number of steps not more than $Ck \cdot \frac{\hat{d}}{\hat{\mu}}$, and this estimation does not depends on initial solution.

Theorem is proved.

Corollary. If $k \to \infty$, there is probability convergence of received solutions consequence to the global optimum.

This fact is obvious according to the probability convergence definition.

The key idea of received convergence rates is in generalization of rates received previously for SA [8].

Also, received convergence rates can be used in convergence analysis of other accelerated probabilistic modeling algorithms and other similar search techniques.

Practical application

Proposed combinatorial optimization algorithms can be applied to solve different classes of problems because of its general requests for problem formulation. The complex analysis of algorithms solving popular COP (like TSP and QAP), alongside with specific problems (like location problem, warehousing problem etc) was made.

TSP was the very first application of *G*-algorithms because this problem is a well-known proving ground for new algorithms construction. Paper [9] presents some received results, and these results congruent with best known solutions received on supercomputing systems. Also, some essential experience in combining algorithms construction (based on genetic algorithm and *G*-algorithm hybridization) was accumulated.

QAP (Quadratic Assignment Problem) is also very popular in different branches like economics, production planning etc. Despite of the fact that this problems is well-known and been solved during years, it is hard to find the exact solution for problems with dimension n > 15-20, and even ε-close solution finding is NP-hard problem [1]. Performed results allowed choosing most effective calculation schemes, whose were used to build different algorithms like *GS*-algorithm (using "golden section" rule) and metaheuristics used for TSP and QAP solving [5,9].

In paper [10] some specific problem of optimal cutting was presented. This is a problem of optimal bar layout on semi-infinite tape, when bars have same height and different width. Some algorithms based on local optimization schemes, SA and *G*-algorithms were proposed. Received results proved that *G*-algorithm performed better results comparing with other algorithms.

The warehousing problem is a very important strategic problem in case of different types of products and forecasts about possible needs [11]. Presented *G*-algorithm was compared with three LS and SA algorithms. Received results showed that *G*-algorithm can find high accuracy solutions in less time than other rivals.

Problem of optimal network channel capacity is one of the most important problems in telecommunication network construction with ATM technology. In this case network consists of commutators connected with fixed length network channels. Some requirements about information traffic are given for each pair of commutators, and general information flows are given for each channel. Network channel capacity for some channel depends on basic channel capacity. Problem is to find such number of basic channels that the cost of network will be the smallest and some requirements about service quality will be met. Some algorithms presenting Local Search, Iterated Local Search, Simulated Annealing, *G*-algorithms and Genetic Algorithms were developed. Practical application showed that for different traffic types best solutions in most cases were found be *G*-algorithm and ILS [12].

Conclusion

Some SLS algorithms are presented in this paper, alongside with own modification called *GS-Algorithm*, using "golden section" rule. The combination of probabilistic calculation mechanisms with LS procedure allowed successfully solve different classes of combinatorial optimization problems. An upper bound of global optimum convergence rate is received with usage of Markov chain theory.

Some examples of algorithm applications for some COP classes solving are given. According to received results, proposed algorithm can be used for applied problems solving allow and used as built-in procedure in metaheuristics (first of all, population-based metaheuristics [2]).

Bibliography

1. *Papadimitriu C.H., Steigliz K.* Combinatorial Optimization: Algorithms and Complexiti. – Engiewood Cliffs: Pretice-Hall, 1982.
2. *Hoos H.H., Stützle T.* Stochastic Local Search: Foundations and Applications. – San Francisco: Morgan Kaufmann Publ., 2005.
3. *Berge C.* Principes de combinatoire. – Paris: Dunod, 1968.
4. *Гуляницкий Л.Ф.* Модифицированные алгоритмы вероятностного моделирования в комбинаторной оптимизации //Технология и методы решения задач прикладной математики. – Киев: Ин-т кибернетики им.В.М.Глушкова АН Украины, 1991. – С. 10–14.
5. *Гуляницкий Л.Ф.* Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования // Компьютерная математика. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2004. – №1. – С. 64–72.
6. *Гуляницький Л.Ф., Турчин О.Я.* Использование правила "золотого сечения" в алгоритмах вероятностного моделирования // Журн. вычисл. и прикладной математики. – 1998. – № 1(83). – С.26–29 (укр.).
7. *Golden B.L., Skiscim C.C.* Using simulated annealing to solve routing and location problem // Naval Research Logistics Quarterly. – 1986. – 33, № 6. – P. 273–277.
8. *Rajasekaran S.* Simulated annealing and nested annealing // J.of Global Optimization.– 2000.–16. –P. 43-56.
9. *Hulianitsky L., Turchin A.* On approach to combining genetic algorithms and accelerated probabilistic modeling algorithms /The Experience of Designing and Application of CAD Systems in Microelectronics: Proc. of the VII Int. Conf. CADMS 2003 (18-22 February 2003, Lviv-Slavske, Ukraine). –Lviv: Publ. House of Lviv Polytechnic National University, 2003. – P. 218–219.
10. *Гуляницкий Л.Ф., Гобов Д.А.* О сравнении эффективности алгоритмов решения одного класса задач размещения прямоугольников на полубесконечной ленте //Компьютерная математика. – 2003. – №1. – С. 88-97.
11. *Гуляницкий Л.Ф.* О решении на ПЭВМ одной задачи о складировании продукции // Теория и программная реализация методов дискретной оптимизации. -Киев: Ин-т кибернетики им.В.М.Глушкова АН УССР, 1989. – С.20-25.
12. *Hulianytskyi L., Baklan A.* Optimization of ATM telecommunication Networks // Int. J. "Information theories & applications". – 2005. – 12, N 4. – P. 328–335.

Authors' Information

Leonid F. Hulianytskyi – *V.M.Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkova ave. 40, Kyiv, 03680, Ukraine; e-mail: lh_dar@hotmail.com*

Alexander Y. Turchin - *V.M.Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkova ave. 40, Kyiv, 03680, Ukraine; e-mail: turchin@ua.fm*

THE ANALYSIS AND OPTIMIZATION OF SURVIVABILITY OF MPLS NETWORKS

Mohammadreza Mossavari, Yurii Zaychenko

***Abstract**: The problem of MPLS networks survivability analysis is considered in this paper. The survivability indexes are defined which take into account the specificity of MPLS networks and the algorithm of its estimation is elaborated. The problem of MPLS network structure optimization under the constraints on the survivability indexes is considered and the algorithm of its solution is suggested. The experimental investigations were carried out and their results are presented.*

Introduction

Last years due to fast increase in volumes of the transferred information in computer networks, the necessity to transfer video, audio information and multimedia information, the need to develop new communication technology has appeared that is able to support the transmission of various information types (such as audio, video and data) with required quality of service at high speed.

Technology ATM (Asynchronous Transfer Mode) became the first technology providing integrated transfer of audio, video information and data. However, rigid restriction on the size of transferred cells - of 53 bytes, and also high cost of equipment, in particular ATM switch, precludes its wide usage. Therefore, in the late 1990's instead of it the MPLS technology (Multiprotocol Label Switching) appeared.

This technology gives the common transport mechanism for networks which use protocols TCP/IP, Frame Relay, X.25, ATM. It is based on introduction of streams of various classes of service (CoS), an establishment of priorities in service of various classes and maintenance of demanded quality of service (Quality of Service – QoS) for corresponding classes [1].

The important problem arising at designing of MPLS networks is the problem of the analysis and optimization of survivability indexes.

In E. Zajchenko's works the indicators of survivability for ATM networks have been defined and the method of the analysis of these indicators has been developed.

At the same time specificity of MPLS technology and, in particular, presence of various classes of service (Class of service) and introduction of their priority service do not allow to use directly methods and analysis algorithms of survivability indicators developed for data transmission networks and ATM networks for the analysis of survivability of MPLS networks.

Therefore, the purpose of the present work is development and research of the methods, analysis and optimization algorithms for survivability of MPLS networks.

Statement and model of the analysis of survivability problem

Following work [1] under *survivability of a system* we will understand its ability to keep the functioning and to provide performance of the basic functions (in the reduced volume) at the specified quality of service indicators.

As the basic purpose of a MPLS network is transfer of the specified sizes of input flows of various classes, survivability of MPLS network we estimate by size of the maximum flow which is possible to transfer in a network at failures of its elements - channels and nodes under preservation of the specified level of quality. of service.

There is MPLS network which is described by oriented graph $G = \{X, E\}$ where $X = \{x_j\}$ is a set of nodes,- $E = \{(r,s)\}$ -a set of communication channels (KC); μ_{rs} - capacity of channel (r,s)..

Let us admit, that in a network K classes of flows are transferred (K =$1,\widehat{6}$) (CoS) according to the matrix of requirements, $H(k) = \|h_{ij}(k)\|$ $i = \overline{1,N}$, $j = \overline{1,N}$ (Mbit/s). For each class k the quality of service indicator

(QoS) in the form of the specified value of an average delay $T_{cp,k}$ is introduced which is estimated by following expression [2]:

$$T_{cp,k} = \frac{1}{H_{\Sigma}^{(k)}} \sum_{(r,s)\in E} \frac{f_{rs}^{(k)} \sum_{i=1}^{k} f_{rs}^{(i)}}{\left(\mu_{rs} - \sum_{i=1}^{k-1} f_{rs}^{(i)}\right) \cdot \left(\mu_{rs} - \sum_{i=1}^{k} f_{rs}^{(i)}\right)}, \quad (1)$$

where $H_{\Sigma}^{(k)} = \sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}^{(k)}$, μ_{rs} - capacity of a communication channel (r,s), $f_{rs}^{(k)}$ - a flow value of the-k-th class in the channel (r,s).

It is required to define survivability indicators for the given network.

In work [2] for the analysis of indicators of MPLS networks survivability the following complex indicator has been suggested:

$$P\left\{H_{\Sigma}^{\Phi}(1) \geq r\% H_{\Sigma}^{0}(1)\right\}, P\left\{H_{\Sigma}^{\Phi}(2) \geq r\% H_{\Sigma}^{0}(2)\right\} \ldots P\left\{H_{\Sigma}^{\Phi}(k) \geq r\% H_{\Sigma}^{0}(k)\right\}, \quad (2)$$

where $H_{\Sigma}^{0}(k)$ - flow value of k -th class in failure-free network state;

$H_{\Sigma}^{\Phi}(k)$ is an actual flow value of class k in case of failures, r = (50÷100) % $k = \overline{1,K}$. Thus for an estimation of MPLS networks survivability the vector indicator of kind (1) is used.

As at failures of some channels or the nodes it is not known in advance, what would be the value of the maximum flow of each class at failures, the hypothesis that the general flow structure at failures remains is defined, i.e. the approximate parity on sizes of streams of various classes at failures should remain, namely:

$$H_{\Sigma}^{\Phi}(1) : H_{\Sigma}^{\Phi}(i) : H_{\Sigma}^{\Phi}(K) = H_{\Sigma}^{0}(1) : H_{\Sigma}^{0}(i) : H_{\Sigma}^{0}(K). \quad (3)$$

Algorithm of an estimation of survivability indicators of MPLS networks

Let's consider the MPLS network G = (X, E), consisting of elements (channels and nodes), subjected to environment influence in result of which they fail. It is supposed, that are set reliability characteristics of a network elements- factors of readiness of channels $k_{\Gamma r,s}$ and nodes - $k_{\Gamma r,s}$. $k_{\Gamma,i}$ $(r,s) \in E$ $i = \overline{1,n}$

Let us consider following failure states of network:

1) failure state 1 channel $Z_1 = \{Z_i\}$;
2) failure state 1 node $Z_2 = \{Z_j\}$;
3) failure state 2 channels : $Z_3 = \{Z_r\}$:
4) failure state 1 channel and 1node $Z_4 = \{Z_t\}$;
5) failure state 3 channels: $Z_5 = \{Z_s\}$.

Using model of an active environment, it is possible to define probability of each state $P\{Z_0\}$. For example, if Z_i is the state of channel, $(r_i s_i)$ failure then

$$P(Z_i) = (1 - K_{\Gamma,r,s}) \prod_{(r,s)\neq(r_i,s_i)} K_{\Gamma r,s}, \quad (4)$$

where $K_{\Gamma r,s}$ is a probability of the safe state of channel 1 $(r,s) \neq (r_i,s_i)$, $1 - K_{\Gamma r,s}$ - probability of a channel (r,s) failure.

In work [2] the algorithm of an estimation of survivability indicators of MPLS network has been suggested.

1. We calculate the general intensity of a flow in failure state for all classes of service:

$$H_{\Sigma}^{(0)}(1), H_{\Sigma}^{(0)}(2),..., H_{\Sigma}^{(0)}(K).$$

2. We simulate various failure states: Z_1, Z_2, Z_3, Z_4, Z_5.

For each of them we calculate probabilities $P(Z_i)$ according to (3).

3. We find the value of the maximum flow for all the classes in a condition. $Z_j : H_{\Sigma}^{\Phi}(k, z_j)\ k = \overline{1,K}$

For this purpose we use algorithm of a finding of the maximum flow which have been specially developed in the dissertation.

4. We calculate a complex survivability index for each class of service:

for the first class

$$P\left\{H_{\Sigma}^{\Phi}(1) \geq r\% H_{\Sigma}^{0}(1)\right\} = \sum_{Z_j} P(Z_j), \quad (5)$$

where summation in (5) is performed for all Z_j such that $H_{\Sigma}^{\Phi}(1) \geq r\% H_{\Sigma}^{0}(1)$;

for the k th class

$$P\left\{H_{\Sigma}^{\Phi}(k) \geq r\% H_{\Sigma}^{0}(k)\right\} = \sum_{Z_i} P(Z_i), \quad (6)$$

where summation in (6) is performed for all states Z_i :, $H_{\Sigma}^{\phi}(k) \geq rH_{\Sigma}^{(0)}(k)$

$H_{\Sigma}^{0}(k)$ is intensity of flow of k th class in faultless state of network;

$H_{\Sigma}^{\Phi}(k)$ - actual intensity of a flow of a class k in case of failures, r =(50÷100)%, $k = \overline{1,K}$.

where Z_i is such: that $H_{\Sigma}^{\phi}(r) \geq kH_{\Sigma}^{(0)}(r)$.

Let us construct the received dependences in co-ordinates

$P\{H_{\Sigma}^{\phi}(k)\}$ - $r\% H_{\Sigma}^{0}$ $P\left\{H_{\Sigma}^{\Phi}(1) \geq r\% H_{\Sigma}^{0}(1)\right\}, P\left\{H_{\Sigma}^{\Phi}(2) \geq r\% H_{\Sigma}^{0}(2)\right\}...P\left\{H_{\Sigma}^{\Phi}(k) \geq r\% H_{\Sigma}^{0}(k)\right\}$

Statement of MPLS network optimization problem by survivability indicators

During designing networks by results of the analysis of its survivability indicators **a problem of maintenance of demanded survivability level** appears. It is natural, that this problem can be solved by reservation of its channels and nodes, structural optimization of a network and needs additional expenses.

Therefore, further we will consider statement of the network structural optimization problem by survivability indicators. [3].

Let there be MPLS network which is described by the oriented graph $G = \{X, E\}$ where $X = \{x_j\}$ is a set of network nodes, $E = \{(r, s)\}$ set of communication channels (KC); μ_{rs} - capacity of a channel.

Let us assume, that in the network K flows classes are transferred (To $= 1,\widehat{6}$) (CoS) according to matrixes of requirements, $H(k) = \|h_{ij}(k)\|\ i = \overline{1,N}$ ($j = \overline{1,N}$ Mbit/s). For each class k the quality of service indicator (QoS) in the form of the value of an average delay $k\ T_{cp,k}$. is set up. Let proceeding from a network functional purpose the following values of survivability indicators for the k th class flow are established k ; $P_{0зад}^{(k)}, P_{1зад}^{(k)}, ..., P_{5зад}^{(k)}$.

It is required to define such structure of a network for which for all classes K following restrictions on survivability level will be provided:

$$P\left\{H_{\Sigma}^{\phi}(k) \geq r\% H_{\Sigma}^{(0)}(k)\right\} \geq P_{кзза}, \ r = (50 \div 100)\ \%,\ k = \overline{1,K} \quad (7)$$

and additional expenses would be thus minimum.

We shall provide achievement of demanded level of survivability by introduction of corresponding reservation of the most responsible elements of a network (channels and nodes).

For an estimation of efficiency of channels and nodes reservation the following indicator is suggested:

a) for channels, $\alpha_{r_i s_i} = -\dfrac{\Delta P(Z_i)}{C_{r_i s_i}}$ (8)

where - Z_i a failure state of the channel $(r_i s_i)$;

$\Delta P(Z_i)$ - change of probability of the state Z_i after reservation, $C_{r_i s_i}$ is the cost of such reservation.

The value $\Delta P(Z_i)$ is estimated by the following formula:

$$P_{рез}(Z_i) - P(Z_i) = (P^2_{отк\, r_i s_i} \cdot \prod_{(r,s)\neq(r_i,s_i)} K_{\Gamma r,s} - P_{отк\, r_i s_i} \prod_{(r,s)\neq(r_i,s_i)} K_{\Gamma r,s} = \\ = -(1 - P_{отк\, r_i s_i}) \cdot P_{отк\, r_i s_i} \prod_{(r,s)\neq(r_i,s_i)} K_{\Gamma r,s} = -(1 - P_{отк\, r_i s_i}) \cdot P(Z_i) \quad . \tag{9}$$

The similar formula is used for an estimation of nodes reservation.

Let us notice, that at failures of nodes, all ingoing and outgoing channels fail simultaneously. The indicator $\alpha_{r_i s_i}$ is used for a choice of prime elements (channels and nodes) for reservation. In work the following algorithm of MPLS network optimization by survivability indicators is offered.

Algorithm of MPLS networks optimization by survivability indicators

The algorithm consists of the same iterations, on each of which the next element - the channel or node is reserved.

1st iteration

1. For all channels and nodes we calculate an indicator $\alpha_{r_i s_i}$ by formula (8).

2. We choose channel (r^*, s^*) such that $\alpha_{r^* s^*} = \max\limits_{(r_i s_i)} \alpha_{r_i s_i}$

3. We reserve channel (r^*, s^*) and recalculate survivability indicators for all classes after reservation by the following formula:

$$P^H\left\{H^{\phi}_{\Sigma}(k) \geq r\% H^{(0)}_{\Sigma}\right\} = P\left\{H^{\phi}_{\Sigma}(k) \geq r\% H^{(0)}_{\Sigma}\right\} + \left|\Delta P(Z^*_i)\right|, \tag{10}$$

where $\Delta P(Z^*_i)$ is a change of probability state Z_i after reservation of the channel (r^*, s^*).

4. Check up conditions (11) (restrictions on survivability):

$$P^H\left\{H^{\phi}_{\Sigma}(k) \geq r\% H^{(0)}_{\Sigma}\right\} \geq P_{r,зад}{}^{(k)}, \; r = (50 \div 100)\%, \; k = \overline{1,K}. \tag{11}$$

If restrictions (11) are fulfilled for all r and all classes K then the end, otherwise we go to the 2nd iteration.

We repeat the specified iterations until the condition (11) will be held for all k and r. As on each iteration the values of survivability indicators raise, and their size is limited from top by value 1 the algorithm converges for a final number of iterations.

Experimental researches

The analysis of sensitivity of the received decision to a variation T_{cp}

All experiments were held for factors of readiness of the channels distributed normally at the interval 0.9 – 0.95, and for factors of readiness of the nodes distributed normally at the interval 0.95 – 0.99.

In the first series of experiments sensitivity of survivability indicators of the first class to a variation of restriction on an average delay T_{cp} was estimated. Corresponding results are presented in table 1 and on fig. 1.

Table 1

	T_{cp} = 0,01	T_{cp} = 0,05	T_{cp} = 0,15	T_{cp} = 0,7
P(100%)	0,352779	0,389927	0,410044	0,432404
P(90%)	0,352779	0,389927	0,410044	0,432404
P(80%)	0,482439	0,520454	0,539737	0,565827
P(70%)	0,483985	0,521005	0,541752	0,565827
P(60%)	0,484394	0,521005	0,543298	0,566516
P(50%)	0,484394	0,521005	0,545879	0,566516

In the given series of experiments sensitivity of survivability to a variation T_{cp} for the second class was estimated. Corresponding results are presented on fig. 2

As well as for the class 1 traffic, the variation T_{cp} strongly enough influences on survivability indicators, and at the further increase of T_{cp} the significant improvement of survivability indicators is not observed.

In the following series of experiments sensitivity of survivability to a variation for T_{cp} for the third class was estimated. Corresponding results are presented on fig. 3.

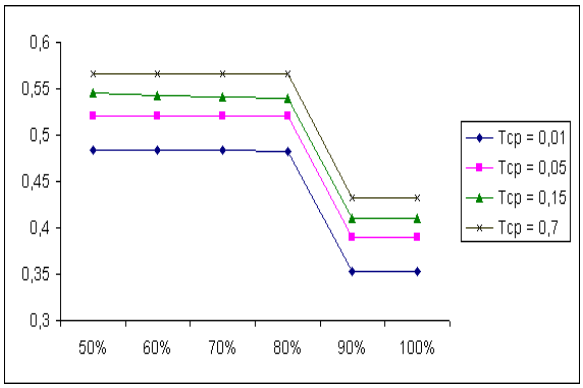


Fig. 1

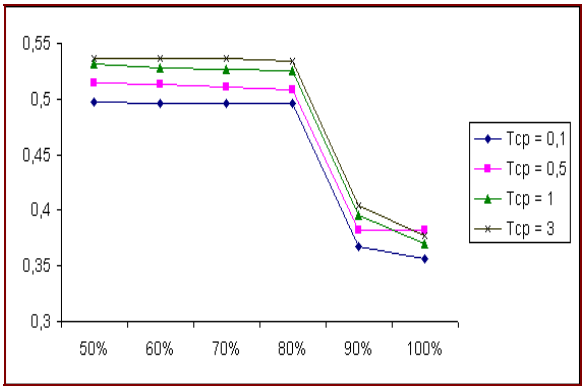


Fig. 2

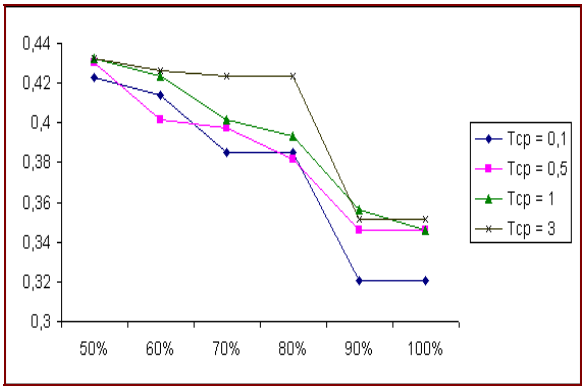


Fig. 3

The analysis of sensitivity of the received solutions to a variation of nodes and channels readiness factors

In the given series of experiments was estimated sensitivity of survivability indicators to a variation of reliable characteristics of network elements factors of readiness of channels and nodes.

Experiment №1. Factors of readiness of nodes are distributed normally on an interval [0,95 – 0,99], factors of readiness of channels – [0,9 – 0,95].

Experiment №2. Factors of readiness of nodes are distributed normally on an interval [0,95 – 0,99], factors of readiness of channels – [0,85 – 0,9].

Experiment №3. Factors of readiness of nodes are distributed normally on an interval [0,9 – 0,95], factors of readiness of channels – [0,85 – 0,9].

Experiment №4. Factors of readiness of nodes are distributed normally on an interval [0,9 – 0,95], factors of readiness of channels – [0,9 – 0,95].

Results for the traffic of class 1 are presented on table 4.

Table 4

	Exper. №1	Exper. №2	Exper. №3	Exper. №4
P(100%)	0,379308	0,212195	0,137087	0,21831
P(90%)	0,379308	0,212195	0,137087	0,21831
P(80%)	0,47672	0,244787	0,166213	0,302938
P(70%)	0,477463	0,24499	0,166213	0,302938
P(60%)	0,478882	0,24499	0,166299	0,303238
P(50%)	0,479821	0,245513	0,166557	0,303238

Conclusions

The problem of the analysis of survivability indicators of MPLS network, in case of failures of its elements-channels and nodes, is formulated.

1. Indicators of networks survivability are defined and the algorithm of their estimation considering the specificity of MPLS networks is offered.
2. The problem of optimization of networks by survivability indicators is formulated and the algorithm of its solution, allowing reaching preset values of survivability indicators is suggested at the minimum additional expenses.
3. Experimental researches of the offered analysis algorithms and optimization of MPLS networks are performed.

Bibliography

1. Зайченко Ю.П., Мохаммадреза Моссавари. Анализ показателей живучести компьютерной сети с технологией MPLS. // Вісник національного технічного університету “КПІ”.Інформатика, управління та обчислювальна техніка.-43,2005,с.73-80.
2. Зайченко Ю.П., Мохаммадреза Моссавари. Оптимизация компьютерных сетей с технологией MPLS по показателям живучести в случае активной внешней среды. // Вісник національного технічного університету “КПІ”.Інформатика, управління та обчислювальна техніка.-.45,2006,с.163-172.
3. Ю.П. Зайченко, А.С. Аникиев, Мохаммедреза Моссавари, Ашраф Абдель-Карим Хилал Абу-Аин. Синтез структуры глобальних компьютерных сетей с технологией MPLS при ограничениях на показатели качества и живучести // Электроника и связь.- 2006.-N 2. - с. 68-71

Authors' Information

Yurii Zaychenko *– professor, PhD, Institute of Applied System Analysis. NTUU. Politechnicheskaya Str. 14, Kiev, Ukraine. phone: +8(044)241-86-93, e-mail: zaych@i.com.ua , baskervil@voliacable.com*

Mohammadreza Mossavari *– PhD Student , Dept. Applied Mathematics, NTUU, Prosp. Pobedyi 37. phone +380677099063*

RECOGNIZING DECOMPOSITION OF A PARTIAL BOOLEAN FUNCTION

Arkadij Zakrevskij

***Abstract**: A hard combinatorial problem is investigated which has useful application in design of discrete devices: the two-block decomposition of a partial Boolean function. The key task is regarded: finding such a weak partition on the set of arguments, at which the considered function can be decomposed. Solving that task is essentially speeded up by the way of preliminary discovering traces of the sought-for partition. Efficient combinatorial operations are used by that, based on parallel execution of operations above adjacent units in the Boolean space*

***Keywords**: Partial Boolean function, non-disjunctive decomposition, weak partition, search by traces, recognition of solution.*

Setting the problem

One of the major problems of the theory of Boolean functions is the problem of functional decomposition, which has useful application in design of logic circuits. Enough to say, that its positive solution can significantly simplify the logic circuits implementation of regarded Boolean functions (for example, at logical synthesis in the basis of LUTs, look up tables). It was set originally in papers [Povarov, 1954], [Ashenhurst, 1959] and [Curtis,1962] as the following problem of *disjoint two-block sequential decomposition*. Suppose a Boolean function $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_n)$ is given, and it is required to represent it as a composition $f(\boldsymbol{x}) = g(h(\boldsymbol{u}), \boldsymbol{v})$ of two functions g and h of smaller number of variables constituting subsets $\boldsymbol{u}$ and $\boldsymbol{v}$. By that $\boldsymbol{x} = \boldsymbol{u} \cup \boldsymbol{v}$ and $\boldsymbol{u} \cap \boldsymbol{v} = \varnothing$.

A necessary and sufficient condition of existing of such a composition was found. Let $\boldsymbol{F}(\boldsymbol{u} \times \boldsymbol{v})$ be the Boolean matrix presenting all values of function f for different values of vector $\boldsymbol{x}$ in such a way, that its rows correspond to values of vector $\boldsymbol{u}$, and its column correspond to values of vector $\boldsymbol{v}$. The condition is formulated as follows: the rows of that matrix can have not more than two different values.

The more general, non-disjoint decomposition was investigated afterwards, at which the given function $f(\boldsymbol{x})$ should be represented as a composition

$$f(\boldsymbol{x}) = g(h(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{w}, \boldsymbol{v}),$$

where three subsets of arguments are connected with the relations $\boldsymbol{x} = \boldsymbol{u} \cup \boldsymbol{w} \cup \boldsymbol{v}$, $\boldsymbol{u} \cap \boldsymbol{w} = \boldsymbol{u} \cap \boldsymbol{v} = \boldsymbol{w} \cap \boldsymbol{v} = \varnothing$, and the couple of sets $\boldsymbol{u}$ and $\boldsymbol{v}$ is regarded as a *weak partition* on set $\boldsymbol{x}$ and is designated $\boldsymbol{u}/\boldsymbol{v}$. Both types of decomposition are illustrated by Fig. 1.

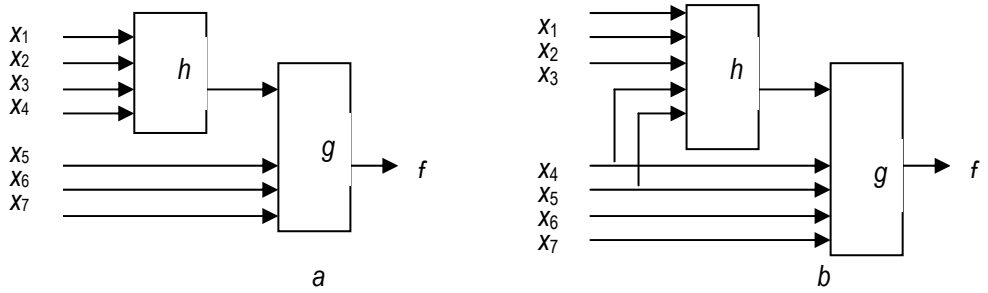


Fig 1. Examples of sequential two-block decomposition of Boolean function $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_7)$: *a*) disjoint, *b*) non-disjoint ($\boldsymbol{u} = (x_1, x_2, x_3)$, $\boldsymbol{w} = (x_4, x_5)$ and $\boldsymbol{v} = (x_6, x_7)$).

To solve the formulated task it is necessary, first of all, to find such a weak partition $\boldsymbol{u}/\boldsymbol{v}$, at which the variables of set $\boldsymbol{u}$ enter in number of arguments of function h only, and variables of $\boldsymbol{v}$ – only in number of arguments of function g. The conditions $|\boldsymbol{u}| > 1$ and $|\boldsymbol{v}| > 0$ should be fulfilled also, otherwise the composition will appear trivial (exists always). Let's name this partition *appropriate*, and the function $f(\boldsymbol{x})$ – *separable*, or *decomposable* at this partition.

The finding of appropriate partition is a difficult task, for which solution an effective combinatorial algorithm was offered [Zakrevskij, 2007a], in case of a completely specified Boolean function. This task becomes even more complicated, when the function f ($\boldsymbol{x}$) appears to be partial, being defined not on all sets of values of variables from set $\boldsymbol{x}$. Just this case is considered below.

Recognizing solution

It] was shown [Zakrevskij, 2007a], that the probability of decomposability of a random completely defined Boolean function fast tends to zero with growth of number of variables n, so already at $n > 9$ such a function, most likely, is not decomposable. In case of partial functions this probability arises with growth of uncertainty, however even in this case it remains small enough, as will be shown below.

Taking into account the given remark, let's assume, that it is known beforehand, that the considered function f ($\boldsymbol{x}$) is separable, being obtained as a result of composition g (h ($\boldsymbol{u}$, $\boldsymbol{w}$), $\boldsymbol{w}$, $\boldsymbol{v}$) of some two Boolean functions g and h on a weak partition $\boldsymbol{u}/\boldsymbol{v}$ on the set of arguments $\boldsymbol{x}$. It is required to detect (to recognize) this partition, after which the obtaining of functions g and h is not a difficult task.

A method of checking a partial Boolean function for decomposability at some given weak partition was offered in [Zakrevskij, 2007b]. An arbitrary Boolean function $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_n)$ was represented there by a "long" Boolean vector $\boldsymbol{f} = (f_0, f_1, \ldots, f_{2^n-1})$, which 2^n components present the values of the function corresponding to values of vector $\boldsymbol{x}$ enumerated in conventional order. For example, vector $\boldsymbol{f}$ = 10011011 represents the function $f(x_1, x_2, x_3)$ taking value 1 on values 000, 011, 100, 110, 111 of vector $\boldsymbol{x}$ and value 0 on values 001, 010, 101. The more convenient for visual perception matrix form of vector $\boldsymbol{f}$ is used below in examples: vector $\boldsymbol{f}$ is divided into parts corresponding to different sets of values of several left components of vector $\boldsymbol{x}$, and these parts play the role of matrix rows.

For example, presenting a Boolean function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ vector

$\boldsymbol{f}$ = **01101101010111100010010000010110110010100111000101000101110100 11**

accepts the following matrix shape, where by thick lines are marked columns and rows where corresponding arguments of the function take value 1:

```
        --------     3
    ----    ----    4
  --  --  --  --   5
 - - - - - - - -  6
  0110110101011110
| 0010010000010110
|  1100101001110001
|| 0100010111010011
1 2
```

In the case, when the appropriate partition is not known a priori, it is possible to organize its search, sorting out different weak partitions and checking the function on decomposability at them. However, such a way is rather labor-consuming, as the number of different weak partitions on the set of variables is approximated from above by value 3^n, fast growing with increase of number of variables n.

In the present paper the method of search for appropriate partition $\boldsymbol{u}/\boldsymbol{v}$ by its traces is suggested, which sufficiently cuts down the number of analyzed partitions. Originally, it was designed for completely specified Boolean functions [Zakrevskij, 2007a], but here it is extended on the case of partial Boolean functions.

Search by traces

The method of decomposition suggested below is based on the following reasons which key moments are given in the form of assertions. They were formulated before for the case of completely specified Boolean functions [Zakrevskij, 2007a], but remain valid when partial Boolean functions are considered.

Suppose two partitions $\boldsymbol{u}/\boldsymbol{v}$ and $\boldsymbol{u}^*/\boldsymbol{v}^*$ are given, such that $\boldsymbol{u}^* \subseteq \boldsymbol{u}$ and $\boldsymbol{v}^* \subseteq \boldsymbol{v}$. Let's speak, that partition $\boldsymbol{u}^*/\boldsymbol{v}^*$ *submits* to partition $\boldsymbol{u}/\boldsymbol{v}$, and call it a *trace* of $\boldsymbol{u}/\boldsymbol{v}$,

Assertion 1. If a partial Boolean function $f(\boldsymbol{x})$ is decomposable at partition $\boldsymbol{u}/\boldsymbol{v}$, it is decomposable as well at partition $\boldsymbol{u}^*/\boldsymbol{v}^*$.

Corollary. If the function $f(\boldsymbol{x})$ is not decomposable at partition $\boldsymbol{u}^*/\boldsymbol{v}^*$, it is not decomposable also at partition $\boldsymbol{u}/\boldsymbol{v}$.

Let's assume $|\boldsymbol{u}| = k$ and $|\boldsymbol{v}| = m$. Partition with $k = 2$ and $m = 1$ we shall term as a *triad*. It is the simplest of partitions, at which some nontrivial decomposition can take place.

Assertion 2. A partial Boolean function is decomposable, if and only if it is decomposable if only at one of triads.

Therefore the search for the partition $\boldsymbol{u}/\boldsymbol{v}$ may be started with the search of its traces on the set of triads, i.e. with looking for an appropriate triad. The needed checking of triads can be fulfilled fast enough, as their number is not large, being significantly less than the number of all weak partitions.

Assertion 3. The number of triads is equal to $C_n^2 (n-2) = \frac{n(n-1)(n-2)}{2}$.

Suppose that some appropriate triad $(x_p, x_q)/x_r$ is detected. If it submits to the required partition $\boldsymbol{u}/\boldsymbol{v}$, we can find the latter, having put for the beginning $\boldsymbol{u} = (x_p, x_q)$ and $\boldsymbol{v} = (x_r)$, and then sequentially expanding these two sets, sorting out remaining variables and testing them on possibility of inclusion into set $\boldsymbol{u}$ or $\boldsymbol{v}$.

By reviewing some concrete triad $\boldsymbol{u}/\boldsymbol{v}$ the Boolean space $M = \{0, 1\}^n$, where the partial Boolean function $f(\boldsymbol{x})$ is presented, is divided into 2^{n-3} intervals corresponding to different values of vector $\boldsymbol{w} = \boldsymbol{x} / (\boldsymbol{u} \cup \boldsymbol{v})$. On each of them the corresponding coefficient f_i of disjunctive decomposition of the function by variables of set $\boldsymbol{w}$ is given. It represents some partial Boolean function of variables x_p, x_q, x_r. As a matter of convenience of subsequent reasoning we shall present each of these coefficients by a ternary matrix size 4×2, which rows correspond to values of the two-component vector $\boldsymbol{u}$, and columns – to values of the one-component vector $\boldsymbol{v}$. Let's designate this matrix $\boldsymbol{T}_i$ and name it a *fragment*. Thus, the 2^n-element ternary matrix representing function $f(\boldsymbol{x})$, is decomposed into 2^{n-3} eight-element fragments specifying functions $f_i(x_p, x_q, x_r)$.

A concrete example of such splitting into eight fragments for a partial Boolean function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ and triad $(x_1, x_2)/ x_6$ is shown below.

```
                  ----------    3
            -----       -----   4
          --    --    --    --  5
        -  -  -  -  -  -  -  - 6
    10 0- 1- 11 0- 10 -1 10
 |  -1 00 -1 1- 00 -1 1- -0
|   0- 1- -0 10 10 0- 11 -1
||  1- 01 11 -1 00 10 -0 11
1 2
```

Assertion 4. The function $f(\boldsymbol{x})$ can be decomposed at triad $(x_p, x_q)/x_r$, if and only if each of the coefficients $f_i(x_p, x_q, x_r)$ is decomposable also at the same triad.

It follows from here, that the probability of decomposability of function $f(\boldsymbol{x})$ at a concrete triad is equal to γ^k, where k is the number of coefficients equal 2^{n-3} and γ – the probability of decomposability of one coefficient. In the case of a completely specified Boolean function the last probability is approximated by the value 1/3, and with growth of uncertainty decreases. Nevertheless, the probability of decomposability of the function $f(\boldsymbol{x})$ quickly decreases with growth of the number of its arguments.

Checking triads for fitness

So, a triad is appropriate, if each fragment of the corresponding splitting of the ternary matrix is suitable. That means, the partial function $f(\boldsymbol{x})$ can be completely defined in such a way, that each fragment will contain no more than two types of Boolean rows (each having equal rows). In other words, a fragment is suitable, if it contains no more than two classes of compatible rows. Remind that two ternary rows are compatible, i.e. they could become equal by changing values "–"of some components for 1 or 0, if they are not orthogonal. It follows from here, that the fragment is suitable, if the graph of orthogonality of its rows is bichromatic [Harary, 1969].

Let us offer the following way of checking fragments with the purpose of detection of suitable ones among them. Any fragment contains four rows, therefore the graph of orthogonality has four vertices. It is bichromatic, if it has no cycle of length three. Let's select arbitrary two different vertices. If such a cycle exists, then one of the selected

vertices will belong to it. Therefore, it is enough to test each of these two vertices on belonging to a cycle of length three. If such belonging will not be revealed, graph is bichromatic, and the triad is suitable.

Necessary and sufficient condition of entering a vertex, i.e. corresponding row, in a cycle of length three could be formulated as follows: among rows orthogonal to the given one, there exist mutually orthogonal rows.

For example, the left of the shown below fragments appears to be suitable, and the right – no, as there is a cycle of length three, composed by three last rows: each of them is orthogonal to the other two (look at Fig. 2).

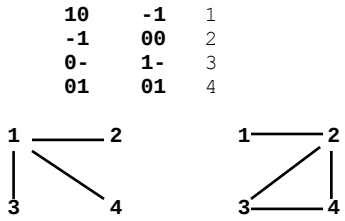


Fig. 2. Graphs of orthogonality of rows of fragments

It is not difficult to check any fragment separately, but the problem is how to check all $2^{|w|}$ fragments for the examined triad and how to find an appropriate partition fast enough. The next part of the paper is devoted to that problem.

Basic operations in Boolean space

A compact and effective set of basic combinatorial operations, which can greatly facilitate the program implementation of the regarded method of search for appropriate partitions, is described below. The parallelism of efficient operations over long Boolean vectors is laid into its foundation, and that essentially accelerates the fulfilled calculations.

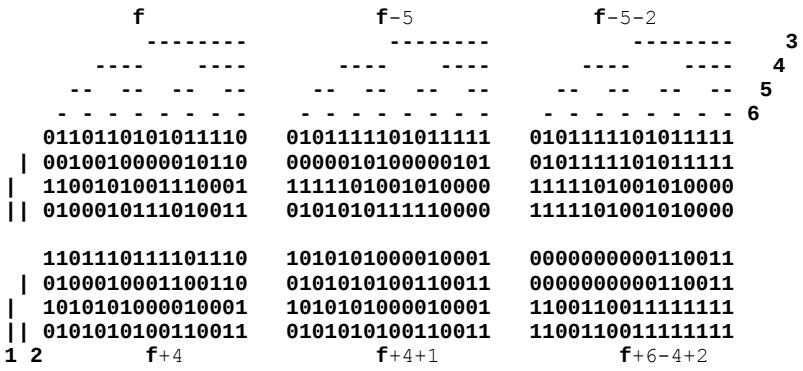
First, let us include into our set the two-place Boolean operations $f \vee g$, $f \wedge g$, $f \oplus g$, $f \sim g$, $f \to g$ which are easily implemented as parallel component-wise operations over corresponding Boolean vectors. They are designated $\boldsymbol{f} \vee \boldsymbol{g}$, $\boldsymbol{f} \wedge \boldsymbol{g}$ (or, simpler, $\boldsymbol{f}\,\boldsymbol{g}$), $\boldsymbol{f} \oplus \boldsymbol{g}$, $\boldsymbol{f} \sim \boldsymbol{g}$, $\boldsymbol{f} \to \boldsymbol{g}$.

Second, we shall supplement them by some useful operations of interaction between adjacent (neighboring) components within the framework of one Boolean vector [Zakrevskij, 2007c].

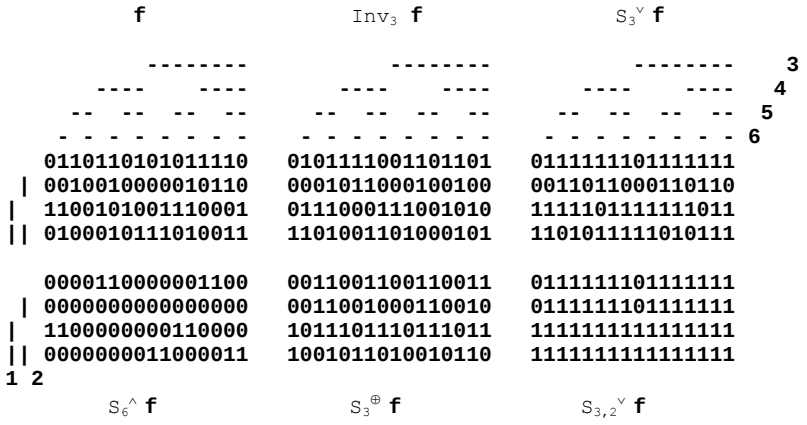
Let's remind, that the function $f(\boldsymbol{x})$ can be represented as Shannon disjunctive decomposition by an arbitrary variable x_i – $f(\boldsymbol{x}) = \overline{x}_i f_{i0} \vee x_i f_{i1}$, which coefficients f_{i0} and f_{i1} are Boolean functions obtained as a result of substitution of values 0 or 1 for variable x_i

Using vector representation of the function, we shall designate these operations of substitution accordingly through $\boldsymbol{f} - i$ and $\boldsymbol{f} + i$. They are easily implemented in the Boolean space on couples of elements adjacent by the variable x_i. When executing the operation $\boldsymbol{f} - i$ both elements of the couple gain the value of the element defined by the condition $x_i = 0$, at execution of the operation $\boldsymbol{f} + i$ – gain the value of the other element corresponding to value 1 of variable x_i.

Let's show examples of such operations, and also of their compositions:



By interaction of adjacent units there are implemented also the operation $\mathrm{Inv}_i\, \boldsymbol{f}$ of inverting the function f at the variable x_i (adjacent elements interchange their values), and so-called operations of symmetrization $S_i^{*}\, \boldsymbol{f}$, in which both elements get value defined by the two-place operation $^{*} \in \{ \vee, \wedge, \oplus, \sim g, \rightarrow \}$ above their initial values [Zakrevskij, 1963]. As a result of these operations the function $f(\boldsymbol{x}) = \overline{x}_i f_{i0} \vee x_i f_{i1}$ is transformed correspondingly into functions $\overline{x}_i f_{i1} \vee x_i f_{i0}$, $\overline{x}_i (f_{i0} * f_{i1}) \vee x_i (f_{i0} * f_{i1})$.
Examples of these operations are shown below.



Here $S_{3,2}^{\vee}\, \boldsymbol{f}$ means composition $S_3^{\vee} (S_2^{\vee}\, \boldsymbol{f})$.

Algorithm of checking triads

The suggested way of checking triads is implemented by the following algorithm, which is remarkable by that it checks on fitness simultaneously all 2^{n-3} fragments generated by the given triad, and finds out by that if the triad is appropriate.

The regarded function $f(\boldsymbol{x})$ is represented by a couple of Boolean vectors $\boldsymbol{f}^0$ and $\boldsymbol{f}^1$, in first of which by 1s are marked the values 0 of the function and in the second - values 1. The splitting of space into fragments is fulfilled by the triad $(x_p, x_q) / x_r$.

To begin with, first rows of fragments are selected, which form together the initial coefficient f^{-} of decomposition of the function $f(\boldsymbol{x})$ by variables x_p and x_q (that coefficient corresponds to values $x_p = 0$, $x_q = 0$). The rows orthogonal to this row, are marked by value 1 in the corresponding parts of computed vector $\boldsymbol{g}$, and their values are fixed by the couple of vectors $\boldsymbol{h}^0$ and $\boldsymbol{h}^1$, checked up further for orthogonality. Alike vectors $\boldsymbol{f}^0$ и $\boldsymbol{f}^1$, they are Boolean vectors with 2^n components.

$\boldsymbol{h}^0 := (\boldsymbol{f}^0 - p) - q$	Getting the initial coefficient f^{-}
$\boldsymbol{h}^1 := (\boldsymbol{f}^1 - p) - q$	
$\boldsymbol{g} := S_r^{\vee} (\boldsymbol{h}^0 \boldsymbol{f}^1 \vee \boldsymbol{h}^1 \boldsymbol{f}^0)$	Finding coefficients orthogonal to f^{-}
$\boldsymbol{h}^0 := S_u^{\vee} (\boldsymbol{f}^0 \boldsymbol{g})$	Computing their intersection
$\boldsymbol{h}^1 := S_u^{\vee} (\boldsymbol{f}^1 \boldsymbol{g})$	

If it turns out that $\boldsymbol{h}^0 \boldsymbol{h}^1 \neq \boldsymbol{0}$, the triad is accepted as not appropriate. In case if $\boldsymbol{h}^0 \boldsymbol{h}^1 = \boldsymbol{0}$ the final rows of fragments are checked, which constitute the final coefficient f^{+} (corresponding to values $x_p = 1$, $x_q = 1$).

$\boldsymbol{h}^0 := (\boldsymbol{f}^0 + p) + q$	Getting the final coefficient f^{+}
$\boldsymbol{h}^1 := (\boldsymbol{f}^1 + p) + q$	
$\boldsymbol{g} := S_r^{\vee} (\boldsymbol{h}^0 \boldsymbol{f}^1 \vee \boldsymbol{h}^1 \boldsymbol{f}^0)$	Finding coefficients orthogonal to f^{+}
$\boldsymbol{h}^0 := S_u^{\vee} (\boldsymbol{f}^0 \boldsymbol{g})$	Computing their intersection
$\boldsymbol{h}^1 := S_u^{\vee} (\boldsymbol{f}^1 \boldsymbol{g})$	

If $\boldsymbol{h}^0 \boldsymbol{h}^1 \neq \boldsymbol{0}$, then the triad is not appropriate. On the other hand, if $\boldsymbol{h}^0 \boldsymbol{h}^1 = \boldsymbol{0}$, the triad is accepted as appropriate.

Example. Let's return to regarding the partial Boolean function $f(x_1, x_2, x_3, x_4, x_5, x_6)$, representing it by a couple of Boolean vectors (rolled up into matrices) $\boldsymbol{f}^0$ and $\boldsymbol{f}^1$:

```
                          ----------    3
                     -----       -----   4
                --    --    --    --  5
              -  -  -  -  -  -  -  - 6
        10 0- 0- 11 0- 10 -1 10
     |  -1 10 -1 1- 00 -1 0- -0
    |   0- -1 -0 00 10 0- 11 -1
    ||  1- -0 11 -1 -0 10 -0 11
   1 2
      01 10 10 00 10 01 00 01   f0
      00 01 00 00 10 00 10 01
      10 00 01 11 00 10 00 00
      00 01 00 00 00 01 01 00

      10 00 00 11 00 10 01 00   f1
      01 10 01 10 00 01 00 00
      00 01 00 00 10 00 11 01
      10 00 11 01 00 10 00 01
```

The check of the function for decomposability at triad $(x_1, x_2) / x_6$ is reduced to testing in parallel all of fragments for fitness. In the given example all of eight fragments are suitable, therefore the function can be decomposed at that triad.

We shall illustrate that operation by the case of testing one of the fragments, third at the left, demonstrating initial values of appropriate components of the ternary vector $\boldsymbol{f}$ and of vectors obtained sequentially by the algorithm:

$$\mathbf{f}^0,\ \mathbf{f}^1,\ \mathbf{h}^0,\ \mathbf{h}^1,\ \mathbf{h}^0\mathbf{f}^1,\ \mathbf{h}^1\mathbf{f}^0,\ \mathbf{g},\ \mathbf{f}^0\mathbf{g},\ \mathbf{f}^1\mathbf{g},\ \mathbf{h}^0 \text{ and } \mathbf{h}^1.$$

The check is carried out first on initial coefficient f^-, and then on finite coefficient f^+.

```
f   f0  f1   h0  h1  h0f1 h1f0 g  f0g  f1g  h0  h1
0-  10  00  10  00  00  00  00  00  00  00  11   Initial
-1  00  01  10  00  00  00  00  00  00  00  11  coefficient
-0  01  00  10  00  00  00  00  00  00  00  11    f-
11  00  11  10  00  10  00  11  00  11  00  11
                                              h0h1 = 0
            00  11  00  10  11  10  00  11  00   Final
            00  11  00  00  00  00  00  11  00  coefficient
            00  11  00  01  11  01  00  11  00    f+
            00  11  00  00  00  00  00  11  00
                                              h0h1 = 0
The triad is appropriate
```

Search for appropriate partition

If the considered triad $(p, q) / r$ has appeared suitable, it is possible to assume, that it is a trace of the sought-for appropriate partition. In this case the latter can be found by moving along the track generated from the found trace. By that the value of vector $\boldsymbol{g}$ obtained at the previous stage is used, and sets $\boldsymbol{u}$ and $\boldsymbol{v}$ are sequentially expanding, beginning with initial values $\boldsymbol{u} = (p, q)$ and $\boldsymbol{v} = (r)$.

Expanding set $\boldsymbol{v}$. Let's begin from set $\boldsymbol{v}$. Sorting out sequentially all elements s from set $\boldsymbol{x} \setminus (\boldsymbol{u} \cup \boldsymbol{v})$, we shall discover among them such ones, at which inclusion in set $\boldsymbol{v}$ the partition $\boldsymbol{u}/\boldsymbol{v}$ remains appropriate. With this purpose three operations are fulfilled for each element s:

$$\boldsymbol{e} := S_s^{\vee}\boldsymbol{g}$$
$$\boldsymbol{h}^0 := S_{\boldsymbol{u}}^{\vee}(\boldsymbol{f}^0\boldsymbol{e})$$
$$\boldsymbol{h}^1 := S_{\boldsymbol{u}}^{\vee}(\boldsymbol{f}^1\boldsymbol{e})$$

and if $\boldsymbol{h}^0\boldsymbol{h}^1 = \mathbf{0}$, then s is included into $\boldsymbol{v}$ by implementing operations

$$\boldsymbol{v} := \boldsymbol{v} \cup \{s\},\ \boldsymbol{g} := \boldsymbol{e}.$$

So the final value of set $\boldsymbol{v}$ is found.

Expanding set $\boldsymbol{u}$. The maximum expansion of set $\boldsymbol{u}$ is found similarly. If it is known, that the required partition $\boldsymbol{u} / \boldsymbol{v}$ is strict (i. e. $w = \varnothing$), it is possible to put $\boldsymbol{u} = \boldsymbol{x} / \boldsymbol{v}$ and, probably, to test the function for decomposability, as the

algorithm used is heuristic. Let's remark, however, that the probability of obtaining by this algorithm erroneous solution fast tends to zero with growth of the number of variables n.

If the required partition could be non-strict, it is necessary to test all elements from initial value of set $\boldsymbol{x} \setminus (\boldsymbol{u} \cup \boldsymbol{v})$ for the possibility of including them into set $\boldsymbol{u}$.

Check of the immediate element s can be fulfilled by the following heuristic algorithm, which partly implements the procedure circumscribed in [Zakrevskij, 2007a]. The algorithm considers the initial coefficient f^{-} of the function f decomposition by the current value of set $\boldsymbol{u}$, finds orthogonal to it coefficients, checks them for compatibility and, in case of compatibility, includes element s in set $\boldsymbol{u}$ without further check.

$\boldsymbol{e} := \boldsymbol{u} \cup \{s\}$
$\boldsymbol{h}^0 := \boldsymbol{f}^0 - \boldsymbol{e}$
$\boldsymbol{h}^1 := \boldsymbol{f}^1 - \boldsymbol{e}$
$\boldsymbol{g} := S_{\boldsymbol{v}}^{\vee}(\boldsymbol{h}^0\boldsymbol{f}^1 \vee \boldsymbol{h}^1\boldsymbol{f}^0)$
$\boldsymbol{h}^0 := S_{\boldsymbol{u}}^{\vee}(\boldsymbol{f}^0\boldsymbol{g})$
$\boldsymbol{h}^1 := S_{\boldsymbol{u}}^{\vee}(\boldsymbol{f}^1\boldsymbol{g})$

If $\boldsymbol{h}^0\boldsymbol{h}^1 = \boldsymbol{0}$, then s is included into $\boldsymbol{u}$ by operation $\boldsymbol{u} := \boldsymbol{e}$.

In such a way the set $\boldsymbol{u}$ is found and, therefore, the whole partition $\boldsymbol{u}/\boldsymbol{v}$.

Note, that the operation of looking for coefficient f^{-} is presented in this algorithm in abbreviated form, by expressions $h^0 := \boldsymbol{f}^0 - \boldsymbol{e}$ and $\boldsymbol{h}^1 := \boldsymbol{f}^1 - \boldsymbol{e}$, instead of more detailed

$$\boldsymbol{h}^0 := (\ldots((\boldsymbol{f}^0 - e_1) - e_2) - \ldots) - e_t,$$
$$\boldsymbol{h}^1 := (\ldots((\boldsymbol{f}^1 - e_1) - e_2) - \ldots) - e_t,$$

where $e = (e_1, e_2, \ldots, e_t)$.

Results of experiments

To estimate the efficiency of suggested methods and algorithms and determine the area of their practical application, an experimental system was used based on principles described in [Zakrevskij, 2006a]. It includes a generator of a flow of random examples of initial data, which essentially supplements the well known and widely spread mechanism of Benchmarks, because it enables statistical investigation of regarded algorithms.

The suggested heuristic algorithm was programmed in C++ and tested on computer (Pentium IV, 2.8 GHz) [Zakrevskij, 2006b]. In a series of experiments with completely defined Boolean functions the values n, $k = |\boldsymbol{u}|$, $m = |\boldsymbol{w}|$ were set, a random partition $\boldsymbol{u}/\boldsymbol{v}$ on the set $\boldsymbol{x}$ and functions g, h were generated, then function $f(\boldsymbol{x})$ was calculated. After that the given algorithm was fulfilled, which found partition $\boldsymbol{u}/\boldsymbol{v}$ for the function $f(\boldsymbol{x})$, the number q of triads scanned by search for traces was fixed, and the total time t (in seconds) spent during search for the partition was measured.

The obtained results are represented in the following table, which right part corresponds to splitting of the set of arguments $\boldsymbol{x}$ in three parts $\boldsymbol{u}$, $\boldsymbol{w}$ and $\boldsymbol{v}$, whenever possible the same size, and left part - in two: $\boldsymbol{u}$ and $\boldsymbol{v}$.

It should be noted that the table begins with $n = 14$, because $t \le 0.00$, if $n \le 14$.

Table 1. Results of experiments over 15 examples
with increasing number of variables n from 14 up to 28

n	k/m	q	t	k/m	q	t
14	7/7	3	0.00	5/5	39	0.00
15	8/7	2	0.00	5/5	78	0.02
16	8/8	1	0.00	6/5	18	0.01
17	9/8	11	0.02	6/6	34	0.05
18	9/9	4	0.02	6/6	42	0.09
19	10/9	3	0.03	7/6	39	0.17
20	10/10	9	0.11	7/7	3	0.16
21	11/10	1	0.11	7/7	3	0.39
22	11/11	3	0.48	8/7	6	2.41
23	12/11	9	2.17	8/8	16	7.05
24	12/12	3	2.48	8/8	26	18.17

25	13/12	4	5.61	9/8	19	33.16
26	13/13	4	11.64	9/9	3	52.11
27	14/13	19	60.13	9/9	36	187.55
28	14/14	19	1280.67	10/9	3	1585.03

As can be seen from the table, the regarded task of Boolean function decomposition is solved in less than one minute, when the number of arguments does not exceed 26. The amount of memory for representation of Boolean function $f(\boldsymbol{x})$ grows quickly, reaching, for example, $2^{28} = 268\,435\,456$ bits for representation of one long Boolean vector, when $n = 28$. Because of the restrictions on the used operation memory that leads to an essential decrease of the calculation speed.

Conclusion

In this paper, the heuristic algorithm is offered for finding such weak two-block partition on the set of variables of a partial Boolean function, on which the function can be decomposed. The algorithm is effective, if there exists a good solution "hidden" in vector representation of the function of many variables. In this case the search of the partition is reduced to recognition of the latter.

Acknowledgement

This work was supported by Belarusian Republican Fond of Fundamental Researches (Project Ф07МС-034).

Bibliography

[Ashenhurst, 1959] Ashenhurst R.L. The decomposition of switching functions. – Proc. International Symposium on the Theory of Switching, Part 1. – Harvard University Press, Cambridge, 1959, pp. 75-116.

[Curtis,1962] Curtis H.A. Design of switching circuits. – Van Nostrand, Princeton, N. J., 1962.

[Povarov, 1954] Povarov G.N. About functional decomposition of Boolean functions. – Reports of the AS of USSR, 1954. – V. 4, No 5 (in Russian).

[Harary, 1969] Frank Harary. Graph theory. – Addison-Wesley Publishing Company : Reading, Massachusetts; Menlo Park, California; London; Don Mills, Ontario. 1969.

[Zakrevskij, 1963] Zakrevskij A.D. Universal system for solving problems the type relay system synthesis. – Annals of Siberian Physical-Technical Institute, 1963. – Issue 42, pp. 9-37 (in Russian).

[Zakrevskij, 2006a] Arkadij Zakrevskij. Combinatorial methods of solving Boolean problems. – Boolean Problems. 7th International Workshop (ed. by B. Steinbach). – September 21-22, 2006, Freiberg, pp.1-10.

[Zakrevskij, 2006b] Arkadij Zakrevskij. A new heuristic algorithm for sequential two-block decomposition of Boolean functions. – Proceedings of 3rd IFAC Workshop on Discrete Event System Design DESDes'06. September 26-28, 2006, Rydzyna, Poland. University of Zielona Gora. – pp. 13-17.

[Zakrevskij, 2007a] Arkadij Zakrevskij. Decomposition of Boolean functions – recognizing a good solution by traces. – International Journal "Information Theories and Applications", vol. 14, 2007, pp. 359-365.

[Zakrevskij, 2007b] Zakrevskij A.D. Decomposition of partial Boolean functions – checking for decomposability at a given partition. – Informatics, 2007, 1(13), pp. 16-21 (in Russian).

[Zakrevskij, 2007c] Zakrevskij A.D. Parallel operations over neighbors in Boolean space. – Proceedings of the Sixth International Conference CAD DD-07, - Minsk 2007, vol. 2, pp. 6-13.

Author's Information

Arkadij Zakrevskij - *United Institute of Informatics Problems of the NAS of Belarus, Surganov Str. 6, 220012 Minsk, Belarus; e-mail: zakr@newman.bas-net.by*

DETECTION OF LOGICAL-AND-PROBABILISTIC CORRELATION IN TIME SERIES[1]

Tatyana Stupina

Abstract. *An application of the heterogeneous variables system prediction method to solving the time series analysis problem with respect to the sample size is considered in this work. It is created a logical-and-probabilistic correlation from the logical decision function class. Two ways is considered. When the information about event is kept safe in the process, and when it is kept safe in depending process.*

Keywords*: the prediction of heterogeneous variables system, the adaptive method, multidimensional time series, logical decision function.*

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

Introduction

The problem of detection correlations by data, which is presented by time series, is used in different intellectual analysis domains. We have the most difficult problem, when any prior information about process or object is absent. In addition to that several attendant problems are appeared. Firstly, it is necessary to define a class of decision functions (models). Secondary, we must work up a method of plotting optimal decision function, in other words to define optimality criterion by sample. In the third place, we must test our model on adequacy and effectiveness (capacity for general conclusion or statistical stability).

At present time there are many well-known scientific schools, what make researches to that line of investigation [Lukashin Y.P., 2003, Bezruchko B.P., Smirnov D.A., 2003, Lbov G.S., Starceva N.G, 1999]. However universal method is not exists. Several suppositions and priory information are used by every method. It says that problem is actual problem. The method is preferred if it uses lame suppositions to respect with decision function class and if it has capability to retarget model during learning on sample data. At present time so methods use the neural-network technological, the pyramidal-network, the wavelet analysis, the logical structures and others approaches. Such methods we can name as adaptive methods. The conception of adaptive has more comprehensive sense [Lukashin Y.P., 2003, Lbov G.S., Starceva N.G, 1999].

We will interpret concept of adaptive as consecutive selection of model's structure during process of learning on sample data in order to take effective prediction by time series. At the same time it is appeared additional problem – detection a time moment of changing model's structure (criterion of adaptive).

In this paper one is suggested two ways to joint analysis of several unvaried time series by using MLRP-method. When the information about event is kept safe in the process, and when it is kept safe in depending process. That method was applied to prediction of multivariable heterogeneous time series [Stupina T.A., Lbov G.S., 2006]. The solving of practice problem from hydrological domain is presented here by MLRP-method. Model from the logical decision function class we will name as a logical-and-probabilistic correlation [Lbov G.S., Starceva N.G., 1999].

Problem Statement

Let us consider terminal time series $\{x(t), t \in T\}$, it is realization of any time-dependent random process $\eta(t)$. One is supposed that simultaneous distribution $p(\eta_1)$, $p(\eta_1, \eta_2)$, $p(\eta_1, \eta_2, \eta_3)$,…, $p(\eta_1,\ldots, \eta_T)$ is exist. The value set $D_{\eta(t)}$ of variables may be quantitative, nominal and ordinal type in a more case. Let the values of random process $\eta(t)$ are measured at consequent moments of the time with the gap $\Delta t = t_k - t_{k-1}$. Denote this set of moments as $T = \{t_1,\ldots,t_k,\ldots,t_N\}$, N << ∞.

[1] This work was financially supported by Lavrentiev's Grant № 7 of Youth Science Concurs and RFBR 07-01-00331-a

Classical problem of prediction time series is consist in that we must take prediction at time moment $t = t_R$ on time period $t_{R+\tau}$ by analyzing prehistory $\{x(t_k)\}$, $k = 1,...,d$, with length d. As a rule the value τ is named as forestalling. Let us denote the set of every possible prehistory, that have length d, as a D_X, and the set of every possible all forestalling sets as a D_Y. Let us understand a prediction decision function as a f mapping of the D_X set on the D_Y set, i. e. $f : D_X \to D_Y$, $\dim D_X = d$, $\dim D_Y = \tau$. Model's construction f of prediction is defined by decision function class Φ.

If the simultaneous distribution is known than optimal decision function, constructing predicts to time $t + \tau$, is conditional average of distribution $E(\eta_{t+\tau} \mid \eta_{t-d},...,\eta_t)$. In order to solve this problem it is necessary to restore conditional distribution. But that way is not practical because we have not enough size of sample in applied tasks. Therefore it is possible to offer a different depending on specified suggestions targets setting (concerning properties of random process) and the different methods (concerning decision function class) of their decision accordingly.

At present time it was developed many method for prediction depended on time random process (probabilistic characteristics of process are not changed on time). Its methods are based on constructing several models, which usually use some suggestion. For example, if we want to do long-time prediction than the best offer (concerning error variance value) is global model, if we want to do short-term prediction, than the best choice is a local model [Bezruchko B.P., Smirnov D.A., 2003]. Note that most models accomplish solitary prediction and as a rule it is at next time $t + \Delta t$ or at time moment $t + \tau\ \Delta t$, $\tau = 2,...,N - d$.

We propose model, that accomplishes prediction on all forestalling term τ, in other words, to time moments $t + k\ \Delta t$, $k = 1,...,\tau$. That prediction allows to take one decision function (structure of model) and to do simultaneously several predictions on future by one prehistory.

For that problem statement it is important to analyze several steps:

- The detection time moment of changing model's structure (criterion of adaptive);
- The optimization of prehistory length d;
- The optimization of forestalling term τ.

In order to solve these items we will use class of logical decision function. We will consider two ways: a) when the information about event is kept safe in the process, and b) when it is kept safe in depending process. We will perform the primary ideas of these ways in following paragraphs.

Analysis unvariate time series problem

Let we have unvaried time series $\{x(t)\}$ of any random process $\eta(t)$. It is necessary to solve a problem of constructing function f by empirical data, that is presented as terminal points N for given prehistory length d and forestalling term τ. We will construct decision function from the logical decision function Φ_M by sample data, which is made from points of discrete time series. The procedure of building data table $v = \{v_x, v_y\}$ depends on problem statement and on data generally.

For example, it may be

a) Shift of prehistory window step-by-step on time series,

b) Shift of prehistory window to some position on time series,

c) Building prehistory window from the series points, that is positioned on some distance.

Also we can consider some combination of items indicated above. The visual illustration of the unvaried time series and principle of building sample table are presented on figure.1

Not lose commonality let us consider, that $\Delta t = 1$, then prehistory table is built as $v_x = \{x_{kj}\} = \{x(j + k - 1)\}$, where $j = R - d + 1,...,R$, $d \le R \le N - \tau$, $k = 1,...,N - R - \tau$, and forestalling term table (future predictions) is built as a $v_y = \{x_{kj}\} = \{x(j + k)\}$, where $j = R + 1,...,R + \tau$, $k = 1,...,N - R - \tau$, for case (a) as above. With the

help of data table $v = \{v_x, v_y\}$ of the size $N - R - \tau$ we will construct sample decision function $\bar{f}$ from the class Φ_M by the MLRP-method. So we have that choice of optimal length d^* of prehistory corresponds to the choice of informative characteristic subset. A choice of optimal forestalling term length τ^* will be correspond to definition of likely problem size (complexity) for a given sample size.

We define a time moment of changing model's structure (adaptive) as a time moment $t^* = t_{R-d}$ for which the condition $|F(\bar{f}) - F^*| \geq h$ is carried out, where the value F^* is threshold value of model quality, h is admissible value of deviation for established quality.

Below we will consider logical decision function class Φ_M and its properties. We will define criterion of quality $F(f)$ for decision function f.

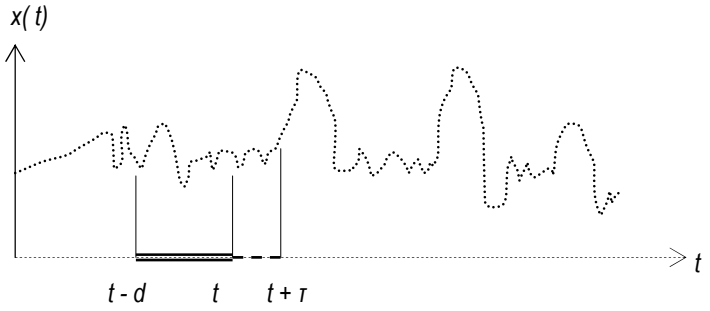


fig.1. Analysis unvariate time series.

The prediction with respect to other time series

This paragraph is devoted to detection of correlation between two unvaried time series. That problem statement is well known and is commonly applied for solving practice problem [Bezruchko B.P., Smirnov D.A., 2003]. However the most methods indicate some power of correlation for the given time point.

The suggested method is founded on constructing function for that a definitional domain is assigned in domain of realizations of one time series $\{x(t)\}$, and a value domain (domain of prediction point $t + \tau\ \Delta t$, $\tau = 1,...,N - d$) is assigned in domain of realizations of other time series $\{y(t)\}$. It is supposed that one time series with respect to other process. The visual illustration of two dependent time series and the principle of building sample table are presented on figure.2.

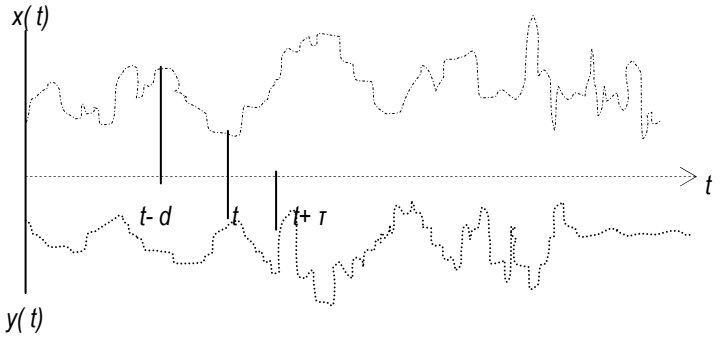


fig.2. Two dependent time series.

The data table is constructed by principle like above. The power of correlation f is defined by quality value $F(f)$ a) on the learning sample and b) on the control sample. We will construct sample decision function $\bar{f}$ (logical-and-probabilistic correlation) from the logical decision function class Φ_M by the MLRP-method like above.

MLRP-method of creating logical-and-probabilistic model

From the beginning we consider a commonly probabilistic statement problem. Let the value (x,y) is a realization of a multidimensional random variable (X,Y) on a probability space $< \Omega, \mathrm{B}, \mathrm{P} >$, where $\Omega = D_X \times D_Y$ is μ-measurable set (by Lebeg), B is the borel σ-algebra of subsets of Ω, P is the probability measure (we will define such as c, the strategy of nature) on B, D_X is heterogeneous domain of under review variable, $\dim D_x = n$, D_y is heterogeneous domain of objective variable, $\dim D_y = m$. The given variables can be arbitrary types (quantitative, ordinal, and nominal). For the pattern recognition problem, for example, the variable Y is nominal. Let us put $\Phi_\circ$ is a given class of decision functions. Class $\Phi_\circ$ is μ-measurable functions that

puts some subset of the objective variable $E_y \subseteq D_Y$ to each value of the under review variable $x \in D_X$, i.e. $\Phi_\circ = \{f : D_x \to 2^{D_y}\}$. For example the domain E_y can contains the several patterns $\{\omega_1,...,\omega_k\}$ for pattern recognition problem.

The quality $F(c,f)$ of a decision function $f \in \Phi_\circ$ under a fixed strategy of nature c is determined as $F(c,f) = \int_{D_x} (P(E_y(x)/x) - \mu(E_y(x)))dP(x)$, where $E_y(x) = f(x)$ is a value of decision functions in x, $P(y \in E_y(x)/x)$ is a conditional probability of event $\{y \in E_y\}$ under a fixed x, $\mu(E_y(x))$ is measurable of subset E_y. Note that if $\mu(E_y(x))$ is probability measure, than criterion $F(c,f)$ is distance between distributions. If the specified probability coincides with equal distribution than such prediction does not give information on predicted variable (entropy is maximum). On the nominal-real space $\Omega = D_н \times D_в$ a measure μ is defined so as any $E \in \mathrm{B}$, $E = \bigcup_{j=1}^{|E_н|} E_в^j \times \{z^j\}$, $\mu(E) = \sum_{j=1}^{|E_н|} \frac{\mu(E_в^j)}{|D_н|\mu(D_в)}$, were $E_н$ is projection of set E on nominal space $D_н$, z^j - item of $E_н$, $E_в^j$ - set in $D_в$ corresponding to z^j, $\mu(E_в^j)$ - lebeg measure of set $E_в^j$. For any subset of domains D_X or D_Y the measure μ is assigned similarly. Clearly, the prediction quality is higher for those E_y whose measure is smaller (accuracy is higher) and the conditional probability $P(y \in E_y(x)/x)$ (certainty) is larger. For a fixed strategy of nature c, we define an optimal decision function $f_\circ(x)$ such as $F(c,f_\circ) = \sup_{f \in \Phi_\circ} F(c,f)$, where $\Phi_\circ$ is represented above class of decision functions.

In commonly when we solve this problem in practice the size of sample is small and type of variables may be different. In this case class of logical decision function Φ_M complexity M [Lbov G.S., Starceva N.G, 1999] is used. For the prediction problem of the heterogeneous system variables class Φ_M is defined as $\Phi_M = \{f \in \Phi_\circ \,|\, f \sim <\alpha, r(a)>, \alpha \in \Psi_M, r(\alpha) \in R_M\}$ (the mark '~' denotes the correspondence of pair $<\alpha, r(a)>$ to symbol *f*), were Ψ_M is set of all possible partitioning $\alpha = \{E_X^1,...,E_X^M \,|\, E_X^t = \prod_{j=1}^n E_{X_j}^t,\ E_{X_j}^t \subseteq D_{X_j}, t = \overline{1,M}, \bigcup E_X^t = D_X\}$ of domain D_X on M noncrossing subsets, R_M is set all possible decisions $r(\alpha) = \{E_y^1,...,E_y^M \,|\, E_y^t \in \Im_{D_Y}, t = \overline{1,M}\}$, $\Im_{D_Y}$ - set of all possible *m*-measuring intervals. For that class the measure $\mu(E_y(x)) = \frac{\mu(E_y)}{\mu(D_Y)} = \prod_{j=1}^m \frac{\mu(E_{y_j})}{\mu(D_{y_j})}$ is the normalized measure of subset E_y and it is introduced with taking into account the type of the variable. The measure $\mu(E_y(x))$ is measure of interval, if we have a variable with ordered set of values and it is quantum of set, if we have a nominal variable (it is variable with finite non-ordering set of values and we have the pattern recognition problem). A complexity of Φ_M class is assigned as M if we have invariant prediction (decision is presented by form: if $x \in E_X^t$, than $y \in E_y^t$), $M_\Phi = M$, and it is assembly $(k_1,...,k_M)$ if we have multivariate, i.e. $E_y^t = \bigcup_{i=1}^{k_t} E_y^i$, $t = 1,...,M$ and $E_y^i \cap E_y^j = \varnothing$ for $i \neq j$ (decision is presented by form: if $x \in E_X^t$, than $y \in E_y^1 \vee E_y^2 \vee ... \vee E_y^{k_t}$). The class of logical decision function has universal property.

Statement. For any function $f \in \Phi^\circ$ and $\varepsilon > 0$ there are M and several logical decision function $f_M \in \Phi_M$ so that $|F(c,f) - F(c,f_M)| \leq \varepsilon$.

Others good properties of the logical decision function class are presented in work [Stupina T.A. 2006] for prediction system heterogeneous variables problem.

If the strategy of nature is unknown the sampling criterion $F(\bar{f})$ is used by method $Q(v_N)$ of constructing sample decision function $\bar{f}$, $\bar{F}(\bar{f}) = \sum_{t=1}^{M'} \bar{p}_x^t (\bar{p}_{y/x}^t - \bar{\mu}_y^t)$, where $\bar{p}_x^t = \frac{N(\tilde{E}_X^t)}{N(D_X)} = \frac{N^t}{N}$, $\bar{p}_{y/x} = \frac{N(\hat{E}_y^t)}{N(\tilde{E}_X^t)} = \frac{\hat{N}^t}{N^t}$, $\bar{\mu}_y = \mu(\hat{E}_y)$,

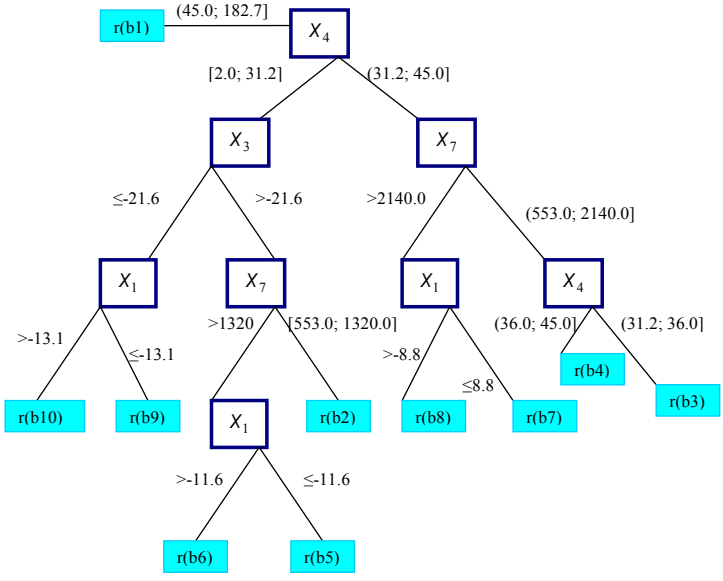
$N('*')$ is number of sample points, generating the set "*", $\bar{f} \sim <\alpha, r(\alpha)>$, $\alpha = \{\tilde{E}_X^1, ..., \tilde{E}_X^{M'}\} \in \Psi_{M'}$, $r(\alpha) = \{\hat{E}_y^1, ..., \hat{E}_y^{M'}\} \in R_{M'}$. The optimal sample decision function is $\bar{f}^* = \arg \max_{\alpha \in \Psi_{M'}} \max_{r(\alpha) \in R_{M'}} \bar{F}(\bar{f})$. In order to solver this extreme problem we apply the algorithm *MLRP* of step-by-step increase attachments of decision trees. It do the branching of top point on that value criterion $\bar{F}(\bar{f})$ is maximum and the top point is divisible or $\bar{F}(\bar{f}) \geq F^*$. The top point is indivisible if 1) number of final top point is $M' = M^*$ or 2) $\hat{N}^t \leq N^*$. That criterion and parameters F^*, M^*, N^* assign method of constructing sample decision function.

In order to estimate the MLRP - method quality we did statistical modeling. The average of the criterion of sample decision function on samples of fixed size $m_F(c) = \mathrm{E}_{V_N} F(c, \bar{f})$ is estimated for fixed nature strategy. Moreover we researched the averaging-out empirical functional quality $\varepsilon_N(c) = \mathrm{E}_{V_N} F(c, \bar{f}) - \mathrm{E}_{V_N} \bar{F}(\bar{f})$ for given strategy of nature with the purpose of estimating decision quality, and maximal removal of empirical functional quality average of distribution $\varepsilon_N^*(c) = \sup_{c: \bar{F}(\bar{f}) = F_o} \varepsilon_N(c)$ for a given empirical quality value $F_\circ$. It was taken for some parametric nature strategy class, for given nature strategy complexity M, decision function complexity M'. The decision function is built by MLRP-method on sample of size N. Parameters n, m (dimensions of domains D_X and D_Y) and number of fixed type variables were considered in problem statement on the whole. It's defined the complexity of nature strategy and complexity of decision function. The GenMLRP-algorithm was developed for modeling nature strategy parameters. Generation nature strategies were realized in accordance with definition, where parameters are established by random in the given interval. The properties of functional quality are presented in work [Stupina T.A., 2006] for uniform distribution on set D_Y.

The MLRP-method was applied for prediction multivariate time series. Three random processes were simultaneously considered instead of one. Feature's systems (under review and predicted) were established. Procedure of building data table is offered in work [Stupina T.A., G. S. Lbov, 2006]. The example of solving practical problem is presented in next paragraph.

Application MLRP-method to prediction multivariate time series task

This paragraph is devoted to some of practical problem from hydrological area. It consists in the prediction of the columbine $(k = 1)$, transmitting across Oby riverbed, the average monthly temperature $(k = 2)$ and the atmospheric precipitates $(k = 3)$ by like hydrometeor data (in the course of the 86 years) in control post of the city Kolpashevo. In order to construct decision function of prediction variable system (y_1, y_2, y_3) in April by variable system in the course November, December and January the average monthly data was



$r(b6) =$"$Y_1 \in [-6.3; 1.8]$" $\wedge$ "$Y_2 \in [13.0; 34.0]$" $\wedge$ "$Y_3 \in [1120.0; 4850.0]$"

fig. 3. The decision tree of hydrological situations is presented.

worked up in the course of November ($i = 1$), December ($i = 2$), January ($i = 3$) and April ($j = 1$) in control post. The sample decision function was constructed by learning data $\{x_k(t_{k_i+12}), y_k(t_{k_j})\}$ of the size 76.

Estimation of quality criterion (probability estimation of veritable decision by rule $\bar{f}$) was taken by control sample of the size 10 (it is last years of time series) and it was equal 0.8 that is satisfactory result.

Four important features were chosen from initial nine features for consecutive constructing decision tree (the decision function from logical decision function may be presented by dichotomous count). In compliance with construction decision function the average monthly temperature in November (x_1) and in January (x_2), the atmospheric precipitates in November (x_4), the columbine in November had the most influence on prediction quality. Visual illustration of some decision tree of hydrological situations and decision example on top $r(b\ 6)$ is presented on figure.3 with complexity M = 10, $N^* = 3$.

Conclusion

In that paper two ways to solving analysis unvaried time series problem was considered. It was founded on the MLRP-method constructing logical-and-probabilistic model for prediction heterogeneous variables system. The idea's approach and ways of realizations was formulated here.

The decision was constructed by MLRP-method from the logical decision function class. It allows taking optimal parameters as such prehistory length and forestalling term for unvaried time series. Practical problem from hydrological area was decided by MLRP-method for prediction variable system. Simulation of the different type time series is matter of future researches.

We want to note that proposed approach to joint analyses of some time series can have more than enough applications. For example, we can solve problem of statistically important correlation detection between seismic processes arising on the most distant region. It allows us to understand and even perhaps to detect earthquake precursors.

Bibliography

[Box Dj., Dgenkins G., 1974] Box Dj., Dgenkins G. Analysis time series. Prediction and direction. Publ. Moskow, Wold.– 1974 - 242 pp.

[Lukashin Y.P., 2003] Lukashin Y.P Adaptive short time prediction methods for time series. Publ. Moskow, Finances and Statistic – 2003 – 416 pp.

[Bezruchko B.P., Smirnov D.A., 2003] Bezruchko B.P., Smirnov D.A. The modern modeling by time series. http://www.nonlinmod.sgu.ru/doc/review.pdf.

[Lbov G.S., Starceva N.G, 1999] Lbov G.S., Starceva N.G. Logical Decision Functions and Questions of Statistical Stability. Inst. of Mathematics, Novosibirsk.

[Stupina T.A., Lbov G.S., 2006] T.A. Stupina. G. S. Lbov. Application of the multivariate prediction method to time series. International Journal ITHEA, Vol 13, No 3. - 2006 - pp.278-285.

[Stupina T.A., 2005] Stupina T.A. Estimation of quality removal for prediction multivariate heterogeneous variable problem. Proceeding of the Russian conference "Mathematical methods of pattern recognition (MMPO-12)", Moskow, 20–26 November 2005 – pp. 209-212.

[Stupina T.A., 2006] T.A. Stupina. Recognition of the Heterogeneous Multivariate Variable. Proceeding of the international conference, 2006 (KDS'2006), Varna (Bulgaria), Vol 1 – pp. 199-202.

Author's Information

Tatyana A. Stupina *– Institute of Mathematics SBRAS, Koptuga 4 St, Novosibirsk, 630090, Russia; e-mail: stupina@math.nsc.ru*

PARAMETRIC IDENTIFICATION AND DIAGNOSIS OF INTEGRATED NAVIGATION SYSTEMS IN BENCH TEST PROCESS

Ilya Prokoshev, Alexander Chernodarov

***Abstract**: Growth of complexity and functional importance of integrated navigation systems (INS) leads to high losses at the equipment refusals. The paper is devoted to the INS diagnosis system development, allowing identifying the cause of malfunction. The proposed solutions permit taking into account any changes in sensors dynamic and accuracy characteristics by means of the appropriate error models coefficients. Under actual conditions of INS operation, the determination of current values of the sensor models and estimation filter parameters rely on identification procedures. The results of full-scale experiments are given, which corroborate the expediency of INS error models parametric identification in bench test process.*

***Keywords**: fault detection, integrated navigation systems, state control, sensors, model of errors, parametric identification, supervision, monitoring, fault diagnosis, diagnostic reasoning*

***ACM Classification Keywords**: B.8.1 Reliability, Testing, and Fault-Tolerance; J.2 Computer Applications: Physical Sciences and Engineering: Aerospace*

Introduction

Most model-based methods for fault detection and diagnosis rely on the idea of analytical redundancy that is the comparison of the actual behavior of a system to the behavior predicted on the basis of the mathematical system model. Typical model-based fault detection process consists of two steps: residual generation and residual assessing/classification. The decision making is actually a process of classifying the residuals into one of two classes: normal and fault.

Nowadays, the necessity for an inertial support of the operation of integrated navigation systems is considered to be proved. Such a support forms is the basis for the highly maneuverable objects continuous navigational support. However, as for the implementation of the potentialities of inertial navigation systems (INSs), the problem of improving their operational characteristics still remains topical. Among such characteristics which significantly affect the navigational safety we may reckon the INS operational-readiness time, INS accuracy, and INS reliability.

Regarding INSs, traditional approaches [1] to the solution of the above problem rely on the hardware modernization of existing sensors and on the development of new types of sensors such as a gyroscope and an accelerometer. Approaches [1], which involve INS error estimation from data obtained from satellite navigation systems (NSs) and from other external NSs are also deemed to be promising. Furthermore, insufficient attention is given, in our opinion, to the study of the capabilities of combined INSs, built around sensors that are different in the: principle of operation. At the same time, available engineering solutions [2] of such a problem provide the necessary basis in order for studies in this particular field to be conducted.

The evolution of INS relies on improvements both in hardware and in the methods of integrating this hardware. The potentialities of INSs [1] are based on the Kalman filtering technology and on the mathematical INS sensors error models. In order for the INS state to be estimated reliably, the parameters both of models and of an optimal Kalman filter (OKF) must reflect actual measuring processes and noise conditions adequately. Therefore, it is essential that during use of INSs, any changes in noise statistics as well as in dynamic and accuracy sensors characteristics be taken into account. This can be done through the identification and retuning of the appropriate coefficients in an algorithm for data processing.

The potentialities of such a technology make it possible

- to combine dissimilar measurement aids into an integrated structure and to improve the accuracy and reliability of navigational determinations on this basis;
- to implement the mutual support of INSs in the interests of ensuring their integrity;

- to estimate both INS errors and sensor errors from indirect measurements and through the use of correlations;
- to form procedures for the monitoring, diagnosis, and control of the INS technical condition.

However, the effectiveness of OKF application as a kernel of INSs essentially depends on the goodness of fit of the mathematical INS error models and sensor errors to actual measuring processes.

Problem Statement

Generally, the structure of navigating system can be presented in the form of three modules, namely: information sensors, information signal converters from the analogue form into digital and digital processing devices (see figure 1).

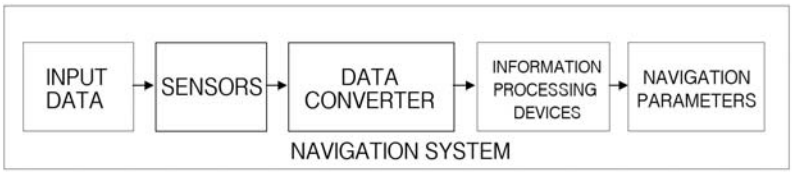


Fig. 1 The structure of typical navigation system

The experience of inertial navigation systems development shows that the intrinsic error of these units defining their functional reliability is the random parametric drift called by dynamically-tuned gyros, interface electronic cards, control cards and couplers. The given task solution is impossible without more profound analysis of occurrence reasons and influence of design and technological parameters on values and stability of random drift.

According to stated, the research of the factors influential in involuntary drift of system and creation of the effective diagnostic technique permitting to estimate current technical condition of INS is the actual task.

The main work purpose is development of algorithms for the INS diagnosis, permitting to reveal reasons of refusals and faults on the data on the basis of structural adapting and navigation model parameters identification.

The offered solution technology includes the following stages:

- the structural adapting of the INS equations in view of the detected disorder and model defect in parametric type;
- retrospective estimation of the extended state INS error vector, originating because of defects;
- correlation processing of the received estimations of errors;
- solution of the algebraic equations on parameters, approximating correlation function and included in diagnostic model;
- INS state handle in view of the current state of meters, namely - retargeting of parameters of error model and INS working capacity restoring.

Given technology will allow solving the following problems:

- optimization malfunctions search strategy;
- separate system units technical condition estimation.

According to the purpose of work it is possible to solve the following research problems:

- the statistical analysis of INS units parameters accuracy not meeting the quality specifications requirements;
- refusals database development of INS interconnected units not past a trial stages;
- open architecture development for processing information from various data sources;
- development realizing automated information capturing for its subsequent processing.
- Parametric Identification of the Error Models of INS Sensors

The full analysis of various methods has led to expediency of application of complex monitoring systems which use different by the physical nature research methods that, in turn, will allow excluding lacks of one method and use advantages of other methods to realize thus a principle of "redundancy" increasing reliability of INS systems.

The integration of navigation systems relies on a priori known models of the errors of sensors, namely, of accelerometers, gyroscopes, pseudorange sensors, and so on. However, when INSs are in use, the sensors characteristics undergo certain changes. This has necessitated taking into account these changes through the models of errors of the sensors. Thus, the problem of identifying the models of sensor errors parameters arises. The above problem can be solved on the basis of correlative processing of the estimates of sensor errors both in real time and in the postprocessing INS diagnosis.

The equations describing INS functioning in an operating mode can be represented in a general view

$$\dot{y}(t) = F(y,t); \tag{1}$$

$$\dot{y}_p(t) = F(y_p,t) + G(t)\xi(t), \tag{2}$$

where $y(t)$ – n-dimensional state parameter vector of ideal (undisturbed) INS;

$y_p(t)$ – n-dimensional state parameter vector of real (disturbed) INS;

$F(y_p,t)$ – vector function;

$\xi(t)$ – r-dimensional INS disturbance vector;

$G(t)$ – link coefficient matrix of $\xi(t)$ vector with $y_p(t)$ vector ($n \times q$ matrix of variable coefficients describing the INS sensors noise intensity).

The following INS error equation could be put in conformity to the correlations (1) and (2):

$$x(t) = A(t)x(t) + G(t)\xi(t); \tag{3}$$

where $x(t)=y_p(t)-y(t)$ - INS error vector (n-dimensional INS disturbed vector parameters deviation from undisturbed INS vector parameters);

$$A(t) = \left.\frac{\partial F(y,t)}{\partial y}\right| y = y_{P(i)} \text{ - } n \times n \text{ partial derivative matrix.} \tag{4}$$

In the onboard implementation of the models of INS errors, it is deemed possible to have an approximate description of the gyro random drift and the accelerometer random displacement as the Markov Gaussian first-order process

$$\Delta\dot{\mu} = -\frac{1}{\tau_\mu}\Delta\mu + \xi\sigma_\mu\sqrt{\frac{2}{\tau_\mu}} \tag{5}$$

with the exponential correlation function

$$R_\mu(t) = \sigma^2{}_\mu e^{-\alpha_\mu|t|}, \tag{6}$$

where t – running time, $\alpha_\mu = 1/\tau_\mu$; τ_μ is the correlation time; $R(0) = \sigma^2{}_\mu$ is the error variance; σ – mean-square deviation of random error; $\xi \in N(0,1)$; μ – sensor related index ($\mu = a$ – accelerometer; $\mu = \omega$ – gyroscope).

Taking into account $A_\mu = \tau_\mu{}^{-1}$; $G_\mu = \sigma_\mu\sqrt{2/\tau_\mu}$ the equation (5) could be embedded in general structure INS error equation (3).

In the equations, coefficients A_μ, G_μ are a priori defined for sensor nonfault states. The change in the technical condition of sensors during use of INSs has an influence both on the parameters and on the structure of the models of errors. Therefore, the need arises for refinement of the models during bench tests. For this purpose, we propose that the technology of structural parametric identification of the models should be used. Such a technology is based on the requirement that the base models of sensor errors be in agreement with the results of correlative processing of estimates in real time. Provision is made for the extension of the sensor errors models, which is adequate to the form of correlation functions when the postprocessing of bench test data is performed.

In relations (1) and (2), the quantity α_μ is the parameter that is to be identified. In this case, the problem can be reduced to the finding of the α_μ value, which minimizes the quadratic function

$$\hat{\alpha}_\mu = \arg\min \sum_{j=1}^{N} (\hat{R}_{\mu j} - \sigma_\mu^2 \, e^{-\alpha_\mu \tau_j})^2 , \tag{7}$$

where $\hat{R}_{\mu j}$ is the correlation function that is determined from experimental data in the following way:

$$\hat{R}_k = \frac{1}{N} \sum_{i=k+1}^{N+k} \overset{\circ}{x}_i \, \overset{\circ}{x}_{i-k} \quad , k = \overline{0, N} ; \quad \overset{\circ}{x}_i = x_i - m_x ; \quad m_x = \frac{1}{N} \sum_{i=1}^{N} x_i \ , \tag{8}$$

where $x_i = x(t_i)$ – estimated error of corresponding sensor; N –number of retrospective counts;

$$\tau_j = j\Delta t_i ; \ \Delta t_i = t_i - t_{i-1} ,$$

t_i – discrete instants of time.

As for NSs, the present state of their hardware support and mathematical-and-software support makes it possible to extend the field of application of the methods of integrated data processing by "entrusting" these methods with the solution of unconventional problems. Among such problems we may reckon monitoring, diagnosis, identification, and estimation of the NS technical condition from bench tests data.

A traditional estimation scheme is an open-loop one intended to compensate for the estimates of INS errors; this scheme is shown in fig. 2, where the following notation is introduced: y_{DS} is the vector of parameters that are reckoned by the INS; y_{EIS} is the vector of parameters that are reckoned by the external information sensors; z is the vector of observations; $\hat{x}$ is the vector of estimates of INS errors, $\omega(t)$ - dynamic system disturbance vector; $\vartheta(t)$ - disturbance vector in observation channel.

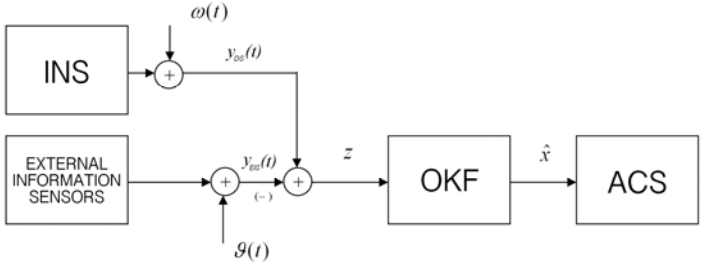


Fig. 2 Open-loop scheme for the damping of INS errors

Using inertial navigation signals and external navigation sensors observations, pseudoranges and velocities measurements the INS error vector estimation could be performed in the following way:

$$z_k = [\varphi \lambda \overline{V}]^T{}_{INS} - [\varphi \lambda \overline{V}]^T{}_{EIS} \tag{9}$$

In the equation (9): φ, λ – geographic latitude and longitude location of moving object; $\overline{V}$ – relative velocity vector of moving object.

Functional monitoring of integrated NSs relies on the technique of channel-wise (element-wise) processing of the vector $z_i = \{z_1,...,z_l\}$ of observations. Based on such a technique, it is apparently possible to check an NS by means of generalized parameters that characterize the state of each of the l measuring channels. For instance, in order to check the j-th measuring channel, use can be made of the normalized residual $\beta_j = \nu_j / \alpha_j$, where α_j is a scaling parameter; $j = \overline{1, l}$; l is the dimensionality of the vector z_i of observations. When observations are processed in "forward" time, the residual ν_j is the difference $\nu_j = z_j - \hat{z}_j$ in value between the actual observation z_j and the predicted observation $\hat{z}_j = H_j \hat{m}_j$, where m_j, $\hat{x}_{i/i}$ is the estimate of the NS error

vector x_i at the i-th step after the j-th component and the whole of the vector z_i of observations are processed, respectively; H_j is the row vector of coupling coefficients. Statistical properties of the above-mentioned residuals can be used for the construction of decision rules.

As is known [4] in the absence of discrepancy between the predicted and real observation, the square of the normalized residual β_j^2 is distributed as χ^2, and the quotient of the actual variance $\hat{\alpha}_j^2$ and the predicted variance α_j^2 has the ϑ^2 distribution.

For the distributions in question, the mathematical expectation and variance have tabulated values. These can be used for the formation of tolerances and for the classification of the types of technical condition of an integrated navigation system [5], i.e., of the good condition, operable condition, etc.

Necessary conditions for the good state of the integrated navigation system in reference to the j-th component z_i of the vector of observations follow from the properties of the residual ν_j, and they have the form

$$\nu_j \in N(0,\alpha^2); \qquad \beta_j^2 = \nu_j^2 / \alpha_j^2 \in \chi^2(1,2); \qquad F_j = \hat{\alpha}_j^2 / \alpha_j^2 \in \vartheta(a,b),$$

whereby $\hat{\alpha}_j^2$ is a true value of the variance of the j-th residual, computed on a moving time interval; *a, b* are the tabulated values of the mathematical expectation and variance for the ϑ^2 distribution.

Using the "three-sigma rule" as well as the properties of the χ^2 and ϑ^2 distributions, one can form the tolerances γ_j^2 and η_j^2 respectively on the good and operable condition of the integrated NS in reference to the j-th vector of observations channel, i.e.,

$$\beta_j^2 \le \gamma^2 = 1 + 3\sqrt{2} \approx 5.2; \qquad F_j \le \eta^2 = a + 3\sqrt{2b}.$$

The parameter β_j^2 is formed using the current residual and it reflects the current status of j-th channel of the vector of observations. If it is out of the tolerance γ^2, this fact may be associated both with outliers and with failures. The parameter F_j is the quotient of the actual and predicted variance of the residual. It is formed over an averaged range of values of the residual on a moving time interval. Therefore, if it is out of the tolerance η^2, this fact may be associated with a gradual failure.

The above method intended for the estimation and functional monitoring permits one to establish only the fact that there is a discrepancy between the output signals of the NSs being united, and this fact manifests itself by means of the appropriate components of the vector of the residuals ν_i. Because of this, in order for the diagnosis to be performed, it is apparently expedient to make use of generalized parameters such that the discrepancy would be ascertained for each component of the state vector of an integrated navigation system. In what follows, we show that in order to localize a trouble for the depth of a sensor, namely, of an accelerometer, a gyroscope, it is possible to use some of the combinations of estimates that were obtained in the processing of observations in "forward" and "backward" time.

Analysis of the Results of Studies

The INS-2000 integrated inertial satellite navigation system [3], developed by the RDC (Ramenskoye) has been the object of experimental studies. One of the experiments has been carried out using a bench set of the INS-2000 system, mounted on a geodetically tied-in rotary table. The rotary table was considered as a reference base intended for determination of the actual phase path. The Poisson algorithm for the reckoning of the geographical position from direction cosines and of the projections of the vector of relative velocity on the axes of an inertial measurement unit (IMU) is a navigational kernel of the INS. The basic state vector was comprised of 18 parameters, namely: errors of the IMU angular position; errors in the reckoning of the components of the vector of relative velocity in the direction of the IMU axes; errors in the reckoning of direction cosines; IMU angular drifts and displacements of accelerometers. The results of INS state estimation are shown on fig. 3

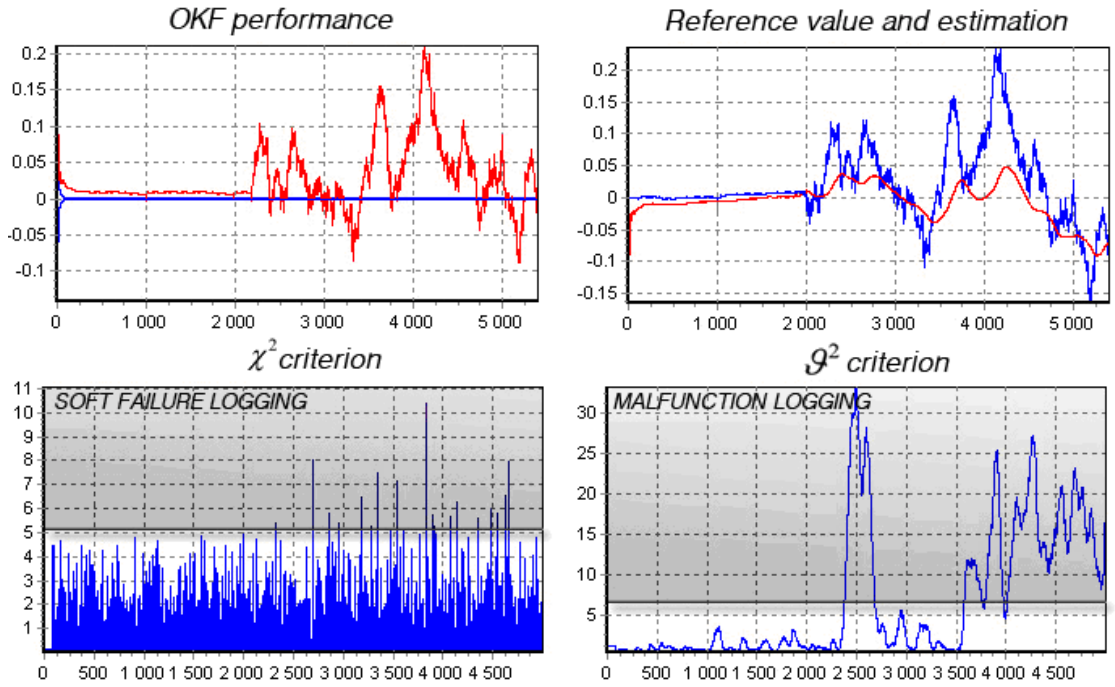


Fig. 3 Results INS state estimation and malfunction logging

Conclusion

In the paper presented here, the authors draw your attention to the importance of systems approach to the construction of mathematical and software support for integrated navigation systems (INSs). Such an approach enables us to combine the capabilities of algorithmic and hardware means intended to improve the accuracy and reliability of INSs. The algorithms considered can form a basis in the construction of a unified technological process, meant for estimation, identification, and control of the INS state. The unified technological process implementation is of great importance in the creation of INSs that provide safety in the use of moving objects.

Bibliography

1. **Schmidt G.T.** INS/GPS Technology Trends. – In: Advances in Navigation Sensors and Integration Technology. RTO Lecture series 232 (2004). Preprints, pp. 1/1 – 1/16.
2. **Fitzgerald R.J.** Divergence of the Kalman filter // IEEE Trans. on Automatic Control. 1971.Vol.16. № 6.– P. 736-747.
3. **Titterton D.N., Weston J.L.** Strapdown Inertial Navigation Technology, Second Edition. Progress in Astronautics and Aeronautics Series, Vol. 207, 2004, 574 pp.
4. **Kailath T.** An innovations approach to least squares estimation. Part 1: Linear filtering in additive white noise // IEEE Trans. on Automatic Control, 1968, Vol. 13, № 6. pp. 646–655.
5. **Sindeev I.M.** (editor). Diagnosis and prediction of the technological condition of aircraft equipment [In Russian].- M: Transport, 1984.

Authors' Information

Ilya V. Prokoshev *- A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, leading engineer, candidate of science, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, +7(495)1352591, E-mail: eldream@e-music.ru*

Alexander V. Chernodarov *- Zhukovsky Air Force Engineering Academy, candidate of science, docent, P.O.Box: Planetnaya St., 125190, Moscow, Russia, E-mail: chernod@mail.ru*

DIGRAPHS DEFINITION FOR AN ARRAY MAINTENANCE PROBLEM

Angel Herranz, Adriana Toni

***Abstract**: In this paper we present a data structure which improves the average complexity of the operations of updating and a certain type of retrieving information on an array. The data structure is devised from a particular family of digraphs verifying conditions so that they represent solutions for this problem.*

***Keywords**: array maintenance, average complexity, data structures, models of computation*

Introduction

Let A be an array of fixed length N with elements belonging to a commutative semigroup an let us consider two operations, Update and Retrieve, with the following intended effect:

- Update(i, x) increments the i-th element of A in x (A(i):= A(i)+ x;).
- Retrieve(i, j) outputs A(i)+ A(i + 1) +... + A(j).

The less space consuming and, likely, the most natural data structure for implementing both operations is the array itself (from now on, expression i..j denotes the set $\{k \in \mathbb{N} \bullet i \leq k \wedge k \leq j\}$):

Example 1

Update(i,x):
begin
A(i) := A(i) + x;
end;

Retrieve(i,j):
begin
return $\sum_{k \in i..j} A(i)$
end;

Running in a random access memory machine, the complexity of Update(i, x) is constant whilst, in the worst case, the complexity of Retrieve(i, j) is linear on N. To improve the complexity of Retrieve the data structure can be reified as an array S of length N + 1 with the property $S(i) = \sum_{k \in 1..N} A(i)$. Then programs must be adapted:

Example 2

Update(i,x):
for** k **in** i..N **loop
S(k) := S(k) + x;
end loop;

Retrieve(i,j):
begin
return S(i)-S(j+1);
end;

For this implementation the complexity of Retrieve is constant whereas, in the worst case, the complexity of Update is linear on N. The design in Example 2 assumes the existence of - (the inverse of +) in the model. This consideration aside, under any execution sequence of operations Update and Retrieve, both implementations are indistinguishable from a functional point of view. This means that Example 1 and Example 2 are different solutions to the same problem definition.

In this paper we are interested in designs with a good average complexity of Update and Retrieve operations when the program variables store elements of a commutative semigroup. Obviously, programs must yield the correct result irrespective of the particular semigroup. Uniform probability of Update and Retrieve execution in programs is assumed in order to improve what we have called average complexity, in other words, we are trying to minimise the sumof the costs of all executions.

In the next Section the RQP (Range Query Problem) and its solutions in terms of digraphs are formalised. Then a particular family of digraphs that represent solutions to the RQP will be presented in an informal way.

Range query Problem

In this section, the range query problem and its solutions are formalized. In this formalization, arrays store elements that belong to a commutative semigroup S. Let us start with the definition of arrays used in this paper.

Definition 1. An array A of length N is a total function from 1..N into S.

Criterion 2. Let A be an array of length N interpreted as a function from 1..N into S: A. 1..N. S. |A| denotes N, dom A denotes 1..N and ran A denotes S.

Definition 3. The Range Query Problem of size N (N-RQP) is the analysis and design of data structures for the implementation of the operations Update and Retrieve where both operations are interpreted as higher order functions:

$$Update : (1..N \to \mathbf{S}) \times 1..N \times \mathbf{S} \to (1..N \to \mathbf{S})$$

$$Update(A,i,x)(j) = \begin{cases} A(j) + x \ \textbf{\textit{if}}\ i = j \\ A(j) \qquad \textbf{\textit{otherwise}} \end{cases}$$

$$Retrieve : (1..N \to \mathbf{S}) \times 1..N \times 1..N \to \mathbf{S}$$

$$Retrieve(A,i,j) = \sum_{k \in i..j} A(k)$$

Definition 4. A N-RQP design is a triple (Z, U, R) where Z is an array of length M with N less or equal M, U is a family of subsets of 1..M indexed on 1..N and R is a family of subsets of 1..M indexed on 1..N × 1..N. Given a N-RQP design (Z, U, R), the implementation of the operations Update and Retrieve is:

```
procedure Update
  (i : 1..N, x : S) is
begin
  for k in U_i loop
    Z(k) := Z(k) + x;
  end loop;
end Update;
```

```
function Retrieve
  (i : 1..N, j : 1..N)
  return S is
begin
  return ∑_{k∈R_ij} Z(i)
end Retrieve;
```

Lemma 5. The complexity of the implementation of Update(i, x) and Retrieve(i, j) in Definition 4 is linear on the cardinal of Ui and Rij, respectively, when running on a random access memory machine.

PROOF. Trivial

Definition 6. A N-RQP design (Z, U, R) is a N-RQP solution if and only if for every $i, j, k \in N$ and $x, y \in S$ the following triplets in the programming logic (annotated programs) are totally correct:

```
--  i ≤ k  ∧  k ≤ j
--  ∧ Retrieve(i,j) = x
Update(k,y);
--  Retrieve(i,j) = x + y
```

```
--  (k < i  ∨  k > j)
--  ∧ Retrieve(i,j) = x
Update(k,y);
--  Retrieve(i,j) = x
```

Lemma 7. A N-RQP design (Z, U, R) is a N-RQP solution if and only if

$$\forall i,j,k \in 1..N \ \bullet\ |R_{ij} \cap U_k| = \begin{cases} 1 \ \textbf{\textit{if}}\ i \le k \ \wedge\ k \le j \\ 0 \ \textbf{\textit{otherwise}} \end{cases}$$

PROOF. This is a well known result and a proof can be found in [1].

Average Complexity in RQP Solutions

As we mentioned in previous sections, we will try to minimize the sum of the costs of all different executions of Update and Retrieve. A uniform probability distribution for each Update possible execution (N operations) and Retrieve possible execution ($\left(\frac{N+1}{2}\right)$ operations) is assumed.

Definition 8. The average complexity of a N-RQP design (Z, U, R) is

$$\frac{\sum_{i=1,j=i}^{N,N} |R_{ij}| + \sum_{i=1}^{N} |U_i|}{N + \binom{N+1}{2}}$$

Minimizing function ϕ below, is enough to minimize the average complexity function above.

$$\phi(N) = \sum_{i=1,j=i}^{N,N} |R_{ij}| + \sum_{i=1}^{N} |U_i|$$

RQP Solutions as Graphs

N-RQP designs can be described in terms of graphs where the content of Rij and Ui are represented as graph vertices and edges (Definition 14). Let us start with some basic definitions.

Definition 9. A digraph, or directed graph, G is a pair (V, E), where V is a finite set (vertex set) and E is a binary relation on V (edge set).

Criterion 10. Notation $u \to v$, instead of (u, v), is used to denote the edges

Definition 11. Let G =(V, E) be a digraph, the out-degree of a vertex u is $\left|\{v \in V \mid u \to v \in E\}\right|$ and the in-degree of a vertex v is $\left|\{u \in V \mid u \to v \in E\}\right|$.

Definition 12. If there is a path from v1 to v2 in a digraph G =(V, E) we say that v2 is reachable from v1. Functions Successors and Ancestors are defined as:

$$Successors(G,u) = \{v \in V \mid v \text{ is reachable from } u\}$$
$$Ancestors(G,v) = \{u \in V \mid v \text{ is reachable from } u\}$$
$$Successors*(G,u) = \{u\} \cup Successors(G,u)$$
$$Ancestors*(G,v) = \{v\} \cup Ancestors(G,v)$$

Definition 13. An acyclic digraph G =(V, E) is a N-RQP graph if the following conditions hold:

$$(1)\ V = 1..M \text{ with } N \leq M.$$
$$(2)\ \text{For every vertex } v \leq N, \text{ its out-degree is } 0.$$
$$(3)\ \text{For every vertex } v > N,\ Successors(G,v) \cap 1..N \neq \emptyset.$$

Definition 14. Given a N-RQP graph G =(V, E), the N-RQP design (Z, U, R) is a N-RQP design in terms of G if it verifies the following properties:

$$(1)\ |Z| = |V|$$
$$(2)\ U_i = Ancestors^*(G,i)$$
$$(3)\ R_{ij} \text{ is the set of vertices with the smallest cardinal that verifies:}$$
$$\bigcup_{u \in R_{ij}} Successors^*(G,u) \cap 1..N = i..j$$
$$\bigcap_{u \in R_{ij}} Successors^*(G,u) \cap 1..N = \emptyset$$

The existence of Rij is guaranteed for every i, j such that $1 \leq i \leq j \leq N$ because in the absence of a set Rij with a cardinal less than j. i +1 we would end up with Rij = i..j. With respect to the uniqueness of Rij several sets could

exist with a smallest cardinal verifying the conditions in Definition 16 so an arbitrary criterion should be given (lexicographic order, for instance).

The following theorem states that N-RQP graphs represent N-RQP solutions:

Theorem 15. Let G =(V, E) be a N-RQP graph, a N-RQP design in terms of G is a N-RQP solution.

PROOF. Let us consider the execution of an arbitrary program:

r := Retrieve(i,j);
Update(k,x);
r' := Retrieve(i,j);

with $i,k \in 1..N$ and $j \in i..2N$. We have to prove that if $i \leq k \leq j$ then $r' = r + x$; otherwise $r' = r$. The proof is based on the following obvious fact: $r' = r + \left|U_k \cap R_{ij}\right| x$ (observe that *Retrieve*(i, j) is $Z(u_1)+ \ldots + Z(u_n)$ where $R_{ij} = \{u_1,\ldots,u_n\}$).

- Case $k < i \vee j < k$: in this case $U_k \cap R_{ij} = \varnothing$ therefore $r' = r$.
- Case $i \leq k \leq j$: in this case $|U_k \cap R_{ij}| = 1$ therefore $r' = r + x$.

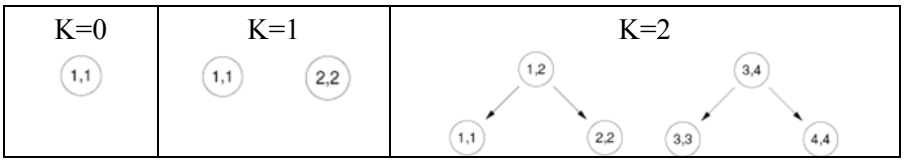


Fig. 1. 2^K-RQP graphs for $K \in 0..2$

Constructing RQP Solutions

The inspiration of our approach comes from a particular family of N-RQP graphs where N is a power of 2. In the solution designs in terms of graphs of this kind, the cost of Retrieve operations is less or equal to 2.

Graphs of this family are called 2^k- RQP graphs, with $K \in \mathbb{N}$. The construction method is described by induction on K and Figure 1 presents the trivial examples for K=0,1,2.

The reader will observe that, strictly speaking, graphs presented in this section are not N-RQP graphs because their vertices are not positive integers but pairs (i, j) of positive integers where $(i \leq j \wedge j \leq N)$. This is not an important problem, as pairs can be trivially encoded as positive integers 1 and an isomorphic N-RQP graph would be obtained. Authors pursue elegance in the presentation so vertices as pairs (i, j) are maintained.

The main characteristic of the construction of N-RQP solutions is that our graphs have the following property:

$$1 \leq \left|R_{ij}\right| \wedge \left|R_{ij}\right| \leq 2$$

Intuitively, the 2^{K+1}-RQP graph can be built by cloning twice the 2^K-RQP graph and then adding new vertices and edges that maintain the above mentioned property. To achieve this aim, after cloning the 2^K-RQP graph, new vertices and edges will be added taking into account that the property on Rij holds if j $\leq 2^K$ or i> 2^K.

Let us show an example; the 8-RQP graph in Figure 2 is the result of cloning the 2^2-RQP graph in Figure 1. In the 8-RQP design (Z, U, R) in terms of that graph, the values of all Ui and those Rij such that |Rij| > 2 are:

$U_1 = \{(1, 1), (1, 2)\}$ $U_2 = \{(2, 2), (1, 2)\}$ $U_3 = \{(3, 3), (3, 4)\}$ $U_4 = \{(4, 4), (3, 4)\}$
$U_5 = \{(5, 5), (5, 6)\}$ $U_6 = \{(6, 6), (5, 6)\}$ $U_7 = \{(7, 7), (7, 8)\}$ $U_8 = \{(8, 8), (7, 8)\}$

$R_{15} = \{(1, 2), (3, 4), (5, 5)\}$
$R_{16} = \{(1, 2), (3, 4), (5, 6)\}$
$R_{17} = \{(1, 2), (3, 4), (5, 6), (7, 7)\}$
$R_{18} = \{(1, 2), (3, 4), (5, 6), (7, 8)\}$
$R_{25} = \{(2, 2), (3, 4), (5, 5)\}$
$R_{26} = \{(2, 2), (3, 4), (5, 6)\}$
$R_{27} = \{(2, 2), (3, 4), (5, 6), (7, 7)\}$
$R_{28} = \{(2, 2), (3, 4), (5, 6), (7, 8)\}$
$R_{37} = \{(3, 4), (5, 6), (7, 7)\}$
$R_{38} = \{(3, 4), (5, 6), (7, 8)\}$
$R_{47} = \{(4, 4), (5, 6), (7, 7)\}$
$R_{48} = \{(4, 4), (5, 6), (7, 8)\}$

The idea is that new vertices and edges have to be added in order to decrease the cardinal of R_{i4} and R_{5j} to 1. R_{ij} is then obtained as the union of R_{i4} and R_{5j} with a resulting cardinal of 2. In the example $|R_{14}|=|R_{24}|=2$ so a pair of vertices are added representing $R_{14}=R_{12}\cup R_{34}$ and $R_{24}=R_{22}\cup R_{34}$. Reasoning symmetrically with R_{57} and R_{58} we get the 2^3-RQP graph in Figure 3. The application of the idea is shown in the left half of the 16-RQP graph (after clowning the 2^3-RQP) in Figure 4.

Fig. 2. A 8-RQP graph resulting after cloning the 2^2-RQP graph

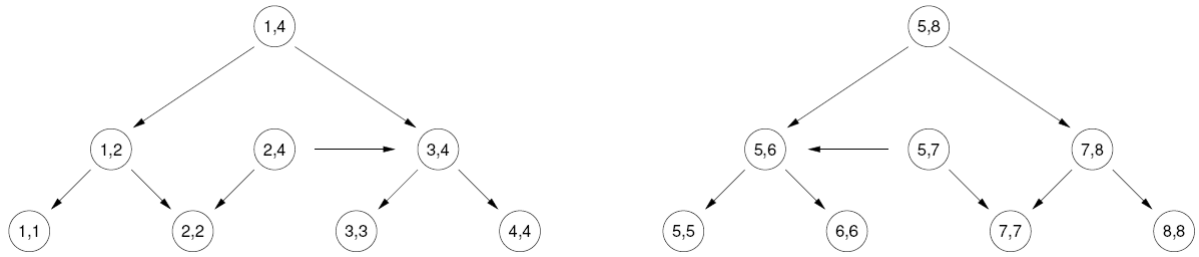


Fig. 3. The 2^3-RQP graph

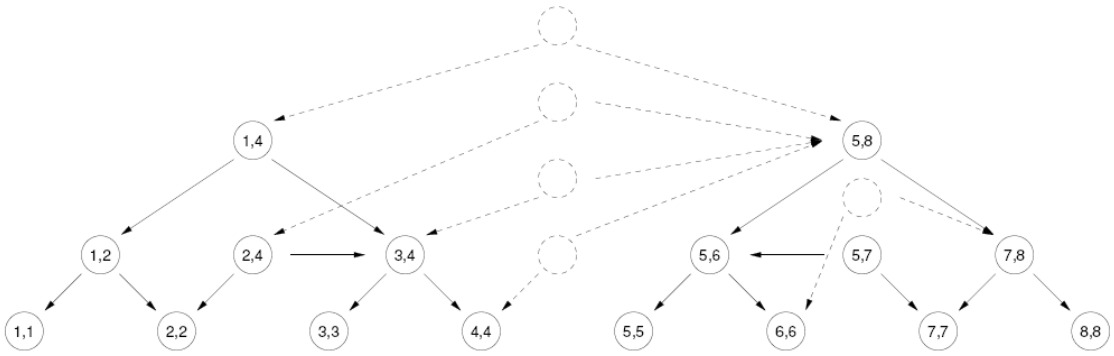


Fig. 4. Left half of the 2^4-RQP graph

Next we present now the formalization of 2^k-RQP graphs.

Definition 16

Let K be a natural number. A 2^K -RQP graph GK is defined inductively:

$$(1) \quad if \quad K = 0 \quad then \quad G^K = (\{1,1)\},0)$$

$$(2) \quad if \quad K > 0 \quad then \quad G^K = Duplicate(G^{K-1})$$

where function Duplicate is defined as

```
function Duplicate (G^K = (V^K,E^K) : Digraph) return Digraph is
  N : constant := 2^K
  M : constant := |V^K|
  V : {(i, j) ∈ 1..N × 1..N • i ≤ j} := ∅;
  E : P(V×V) := 0;
  i, j : 1..(2N);
```

```
begin
  -- The ''cloning'' loops
  for (i, j) in V^K loop
    V := V ∪{(i, j), (i + N, j + N)};
  end loop;
  for (i, j) → (i', j') in E^K loop
    E := E ∪ {(i, j) → (i', j'), (i + N, j + N) → (i' + N, j' + N)};
  end loop;
  -- (V,E) is a graph with two subgraphs which are just like G
  -- but with different node numbering

  for i in 1..(N - 1) loop -- The ''left half'' loop
    j := i + 1;
    while (i,N) ∉ V ∧ j ≤ N loop
      if (i, j) ∈ V ∧ (j, N) ∈ V then
        V := V ∪ {(i, N)};
        E := E ∪ {(i, N) → (i, j), (i,N) → (j,N)};
      else
        j := j + 1;
      end if;
    end loop;
  end loop;

  for j in (N + 2)..(2N) loop -- The ''left half'' loop
    i := j - 1;
    while (N + 1, j) ∉ V ∧ i ≤ 2N loop
      if (N, i) ∈ V ∧ (i, j) ∈ V then
        V := V ∪ {(N, j)};
        E := E ∪ {(N, j) → (N, i), (N, j) → (i, j)};
      else
        i := i + 1;
      end if;
    end loop;
  end loop;
  return (V,E);
end Duplicate;
```

Bibliography

[1] D.J. Volper, M.L. Fredman, *Query Time Versus Redundancy Trade-offs for Range Queries*, Journal of Computer and System Sciences 23, (1981) pp.355--365.

[2] W.A. Burkhard, M.L. Fredman, D.J.Kleitman, *Inherent complexity trade-offs for range query problems*, Theoretical Computer science, North Holland Publishing Company 16, (1981) pp.279--290.

[3] M.L. Fredman, *The Complexity of Maintaining an Array and Computing its Partial Sums*, J.ACM, Vol.29, No.1 (1982) pp.250--260.

[4] A. Toni, *Lower Bounds on Zero-one Matrices*, Linear Algebra and its Applications, 376 (2004) 275--282.

[5] A. Toni, Complejidad y Estructuras de Datos para el *problema de los rangos variables*, Doctoral Thesis, Facultad de Informática, Universidad Politécnica de Madrid, 2003.

[6] A. Toni, *Matricial Model for the Range Query Problem and Lower Bounds on Complexity*, submitted.

Authors' Information

Ángel Herranz Nieva – *Assistant Professor; Departamento de Lenguajes y Sistemas Informáticos; Facultad de Informática; Universidad Politécnica de Madrid; e-mail: aherranz@fi.upm.es*

Adriana Toni – *Facultad de Informática, Universidad Politécnica de Madrid, Spain; e-mail: atoni@fi.upm.es*

ON STRUCTURAL RESOURCE OF MONOTONE RECOGNITION[1]

Hasmik Sahakyan, Levon Aslanyan

Abstract: *Algorithmic resources are considered for elaboration and identification of monotone functions and some alternate structures are brought, which are more explicit in sense of structure and quantities and which can serve as elements of practical identification algorithms. General monotone recognition is considered on multi-dimensional grid structure. Particular reconstructing problem is reduced to the monotone recognition through the multi-dimensional grid partitioning into the set of binary cubes.*

1. Introduction

Monotone Boolean functions have an important role in research area since they arise in various application models, such as design of schemes, pattern recognition, etc.

Monotone Boolean functions are studied in different viewpoint and are known as objects of high complexity. First results, obtained by Mickeev [M, 1959] and Korobkov [K, 1965], characterize Sperner families in unit cube. After enormous investigations and overcoming difficulties, Korshunov [K, 1981] obtained the asymptotical estimate of the number of Monotone Boolean functions. It is characteristic that analytical formulas are not known at this point.

Another cluster of research work solves problems of algorithmic identification of monotone Boolean functions. Hansel [H, 1966] constructed the best algorithm in sense of Shannon criterion, then Tonoyan [T, 1979] constructed a similar algorithm with minimal use of memory. Later on there obtained some generalizations for multi-valued cube. Alekseev [A, 1976] generalized Hansel's result, Katerinochkina [K, 1978] gave precise description of structure of Sperner families.

It is typical that for multi-valued cube there is no explicit formula not only for the number of monotone functions, but also for the cardinality of middle layer. It makes difficult choice of algorithms for a concrete problem and estimation of their complexity.

Below in this paper some algorithmic resources are considered for elaboration and identification of grid defined monotone functions and some alternate structures are brought, which are more explicit in sense of structure and quantities and which can serve as elements of practical identification algorithms.

2. Learning monotone functions on multi-valued cube

Let Ξ^n_{m+1} denotes the grid of vertices of n dimensional, $m+1$ valued cube, i.e. the set of all integer-valued vectors $S=(s_1,s_2,\cdots,s_n)$ with $0\le s_i\le m$, $i=1,\cdots,n$. For any two vertices $S'=(s'_1,s'_2,\cdots,s'_n)$ and $S''=(s''_1,s''_2,\cdots,s''_n)$ of Ξ^n_{m+1} we say that S' is greater than S'', $S'\ge S''$ if $s'_i\ge s''_i$, $i=1,\cdots,n$. We call pair of vectors S' , S'' comparable if $S'\ge S''$ or $S'\le S''$, otherwise these vectors are incomparable. Set of pair wise incomparable vectors composes a Sperner family.

Usually vertices of Ξ^n_{m+1} are placed schematically among the $m\cdot n+1$ layers of Ξ^n_{m+1} according to their weights, – sums of all coordinates. Vector $\tilde{0}=(0,\cdots,0)$ is located on the 0-th layer; then the i -th layer consists of all vectors, with the weight i . An element of i -th layer might be greater than some vector from the $i-1$ -th layer, exactly by one component and exactly by one unit of value (such vector pairs are called neighbors and are connected by an edge). The vector $\tilde{m}=(m,\cdots,m)$ is located on the $m\cdot n$ -th layer.

Consider a binary function f on Ξ^n_{m+1}, $f:\Xi^n_{m+1}\to\{0,1\}$. We say that f is monotone if for any two vertices S' , S'' notion $S'\ge S''$ implies $f(S')\ge f(S'')$. The vector $S^1\in\Xi^n_{m+1}$ is a lower unit of monotone function

[1] The research is supported partly by INTAS: 04-77-7173 project, http://www.intas.be

f if $f(S^1)=1$ and for arbitrary $S \in \Xi_{m+1}^n$, such that $S < S^1$ it is true that $f(S)=0$. The vector $S^0 \in \Xi_{m+1}^n$ is an upper zero of monotone function f if $f(S^0)=0$ and for arbitrary $S \in \Xi_{m+1}^n$, such that $S > S^0$ it is true that $f(S)=1$.

Above defined monotone functions are known also as increasing monotone in contrast with a decreasing monotone function. A function f is decreasing monotone if for any two vertices S', S'', $S' \geq S''$ implies $f(S') \leq f(S'')$. For f decreasing monotone, the vector $S^1 \in \Xi_{m+1}^n$ is an upper unit if $f(S^1)=1$ and for any $S \in \Xi_{m+1}^n$, such that $S > S^1$ we get $f(S)=0$. The vector $S^0 \in \Xi_{m+1}^n$ is a lower zero of function f if $f(S^0)=0$ and for any $S \in \Xi_{m+1}^n$, such that $S < S^0$ we get $f(S)=1$.

In case when $m=1$ the definitions above lead to ordinary monotone Boolean functions defined on binary cube E^n.

Let a monotone function f be defined with the help of an oracle which, receiving any vector $S \in \Xi_{m+1}^n$, gives the value $f(S)$. The problem is in identification of arbitrary monotone function f by as far as possible small number of accesses to the oracle. Similar problems are interested in finding all or the maximal/minimal upper zeros or alternatively the minimal/maximal lower 1's of the given Boolean function. Consider an example. Let, it is given a set of n linear inequalities. A consistent subset of inequalities is coded by a vertex of E^n, where we define f as 0. The problem of finding the maximal consistent subset of inequalities is a known hard problem and the use of oracle reduces the problem to solving several subsystems of inequalities, which is just an alternative way of solving the main problem. The monotone binary function recognition on Ξ_{m+1}^n is the weighted inequalities version of the above given example model.

In [A, 1976] an algorithm of complexity $\leq |M| + \lfloor log_2 m \rfloor \cdot |N|$ is constructed to learn the binary monotone functions above the multi-valued discrete grid, which generalizes the Hansel's method ([H,]) for the case of monotone Boolean functions, here M and N denote the sets of vertices of middle layers of multi-valued grid/cube, i.e. layers which contain vectors with sums of coordinates equal to $\lfloor (m \cdot n)/2 \rfloor$ and $\lfloor (m \cdot n)/2 \rfloor + 1$ respectively. It is also proven that the complexity of the algorithm is approximately $\sqrt{n}$ time less than the whole number of vertices of the grid.

3. Ξ_{m+1}^n partitioning through binary cubes

In this section an alternate approach to traditional means is considered for identification of monotone functions defined on Ξ_{m+1}^n. First Ξ_{m+1}^n is partitioned into binary cube like structures and then Hansel's method is applied for identification of monotone Boolean functions. This approach may serve as a separate element of practical identification algorithms.

In Ξ_{m+1}^n we distinguish several classes of vectors.

Upper and lower homogeneous vectors. A vector of Ξ_{m+1}^n is called an upper h-vector (upper homogeneous) if the values of all its coordinates are at least $m/2$ for even m, and are at least $(m+1)/2$ for odd m. Similarly, a vector of Ξ_{m+1}^n is called a lower h-vector (lower homogeneous) if the values of all its coordinates are at most $m/2$ for even m, and are at most $(m-1)/2$ for odd m.

We denote by $\hat{H}$ the set of all upper h-vectors and by $\check{H}$ the set of all lower h-vectors. The cardinalities of sets $\hat{H}$ and $\check{H}$ are equal to $((m+1)/2)^n$ for odd m and to $(m/2+1)^n$ - for m even.

Middle vectors $\widetilde{m}_{mid+}$ and $\widetilde{m}_{mid-}$

$\widetilde{m}_{mid+} = ((m+1)/2, \cdots, (m+1)/2)$ and $\widetilde{m}_{mid-} = ((m-1)/2, \cdots, (m-1)/2)$ for odd m and $\widetilde{m}_{mid+} = \widetilde{m}_{mid-} = (m/2, \cdots, m/2)$ for even m. $\widetilde{m}_{mid+}$ is located on the $n \cdot (m+1)/2$-th layer of Ξ^n_{m+1} (the lowest layer that contains vector from $\widehat{H}$) and $\widetilde{m}_{mid-}$ is located on the $n \cdot (m-1)/2$-th layer of Ξ^n_{m+1} (the highest layer that contains vector from $\breve{H}$) for odd m; for even m the vector $\widetilde{m}_{mid+} = \widetilde{m}_{mid-}$ is located on the layer $n \cdot m/2$ and this is the only common vector of $\widehat{H}$ and $\breve{H}$.

Vertical equivalent vectors. Let $S', S'' \in \Xi^n_{m+1}$. S' and S'' are called v-equivalent (vertically equivalent) if one of them is obtained from the other by inverting some coordinates (that is replacing some coordinates by their complements up to the m).

For a given vector S denote by $V(S)$ the class of all v-equivalent vectors to S and call it the v-equivalency class of S. This structure $V(S)$ is congruent to a cube E^k, where k is the number of coordinates of S not equal to $m/2$ (this is valid for even m). For odd m $k = n$. It is also evident, that $V(S') = V(S)$ for an arbitrary $S' \in V(S)$.

In $V(S)$ we distinguish two vectors $\widehat{S} = (\widehat{s}_1, \cdots, \widehat{s}_n)$ and $\breve{S} = (\breve{s}_1, \cdots, \breve{s}_n)$ - upper and lower vectors, which coordinates are defined as follows:

$$\widehat{s}_i = \begin{cases} s_i, & s_i \geq m - s_i \\ m - s_i, & s_i < m - s_i \end{cases} \text{ and } \breve{s}_i = \begin{cases} m - s_i, & s_i \geq m - s_i \\ s_i, & s_i < m - s_i \end{cases}, \; i \in \overline{1,n}.$$

These are the only vectors of $V(S)$ that belong to sets $\widehat{H}$ and $\breve{H}$ respectively.

Consequently, for any S the class of its v-equivalency can be constructed by the upper vector $\widehat{S}$ and/or by the lower vector $\breve{S}$ of that class by coordinate inversions. v-equivalency classes of different upper homogeneous vectors are none intersecting.

This proves partitioning of the whole structure Ξ^n_{m+1} through binary cube like vertical extensions of elements of $\widehat{H}$ or $\breve{H}$. The following formula shows the picture of factorization of structure of Ξ^n_{m+1} through these cubical elements: $(m+1)^n = \sum_{k=0}^{n} \left(C_n^k \cdot 2^k \cdot (m/2)^k \right) = \sum_{k=0}^{n} \left(C_n^k \cdot m^k \right)$ for even m and $(m+1)^n = ((m+1)/2)^n \cdot 2^n$ for odd m.

Thus, we get $\left|\widehat{H}\right|$ disjoint subsets, congruent to binary cubes, which cover Ξ^n_{m+1}. Notice that if we construct the corresponding binary cubes, then a pair of vertices, comparable in a binary cube, is comparable also in Ξ^n_{m+1}. Therefore monotonicity in Ξ^n_{m+1} implies monotonicity in all received binary cubes and starting by a monotone function in Ξ^n_{m+1} and reconstructing the implied functions on cubes the initial function will be reconstructed in a unique way.

We recall now the problem of identification of monotone binary functions defined on Ξ^n_{m+1}.

By Hansel's result [H, 1966] an arbitrary monotone Boolean function with k variables can be identified by $C_k^{\lfloor k/2 \rfloor} + C_k^{\lfloor k/2 \rfloor + 1}$ accesses to the oracle.

Hence an arbitrary monotone function defined on Ξ^n_{m+1} can be identified by $\sum_{k=0}^{n} \left(C_n^k \cdot (m/2)^k \cdot \left(C_k^{\lfloor k/2 \rfloor} + C_k^{\lfloor k/2 \rfloor + 1} \right) \right)$ accesses for even m and by $((m+1)/2)^n \cdot \left(C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1} \right)$ - for odd m.

4. Characteristic vectors of subsets partitions of E^n and identification of monotone functions in $\widehat{H}$

For a given m, $0 \le m \le 2^n$ let ψ_m denote the set of all **characteristic vectors of partitions** of m-subsets of E^n. A nonnegative integer-valued vector $S = (s_1, s_2, \cdots, s_n)$ is called characteristic vector of partitions of a vertex subset M, $M \subseteq E^n$ if its coordinates are the sizes of partition-subsets of M by coordinates $x_1, x_2, \cdots, x_n$, which are the Boolean variables composing E^n. s_i is the size of one of partition-subsets of M in the i-th direction and $m - s_i$ is the complementary part of partition. For simplicity we will later assume that s_i is the size of the partition with $x_i = 1$.

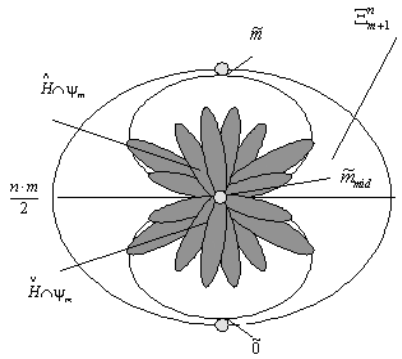
If $m \neq 0$ then ψ_m is not empty. It is also evident that $\psi_m \subseteq \Xi^n_{m+1}$. As other exceptions distinguish between the 2 boundary cases: if $m = 1$ then $\psi_m = \Xi^n_{m+1}$ and so $|\psi_m| = |\Xi^n_{m+1}| = 2^n$; if $m = 2^n$ then $|\psi_m| = 1$ and the vector with all coordinates 2^{n-1} indeed belongs to Ξ^n_{m+1}.

In [S, 2006] the entire description of ψ_m is given in terms of Ξ^n_{m+1} geometry. It is particularly proven that the main problem of describing characteristic vectors can be moved from the Ξ^n_{m+1} to the area of $\widehat{H}$ ($\breve{H}$), where the vector set ψ_m has monotonous structure, – it corresponds to the units of some monotone decreasing binary function defined on $\widehat{H}$ (monotone increasing binary function defined on $\breve{H}$).

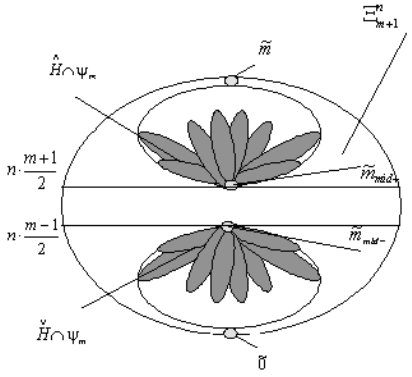
Figure 1 illustrates the sets $\widehat{H} \cap \psi_m$ and $\breve{H} \cap \psi_m$ for even and odd m values, correspondingly.

Thus for entire description of ψ_m it is sufficient to consider all monotone functions defined on $\widehat{H}$ or $\breve{H}$. We shall restrict ourselves to the consideration of decreasing monotone functions defined on $\widehat{H}$. Let $\widehat{\psi}_m$ be the subset of $\widehat{H} \cap \psi_m$ consisting of all upper units of corresponding monotone function.

In [AS, 2001] additional resource is introduced: L_{min} and L_{max}, - minimal and maximal numbers of layers of $\widehat{H}$, - are calculated, such that all vectors of $\widehat{\psi}_m$ are located between them. It importantly follows that the entire description of ψ_m is reduced to the identification of monotone functions with upper units between the layers L_{min} and L_{max}.



a) even m values



b) odd m values

Figure 1

Summarizing all the above consideration we come to the conclusion:

1) Algorithmic resource of learning monotone binary functions defined on Ξ^n_{m+1} includes structures such as:

- generalized Hansel's method and constructions, for monotone binary functions defined on multi-valued cube,
- Ξ^n_{m+1} partitioning through binary cube like vertical extensions of the elements of $\widehat{H}$ together with applying Hansel's result for monotone Boolean functions defined on that cubes,

2) For the entire description of ψ_m we reduce the problem to $\widehat{H}$ becoming able to possess with additional resources:

- learning monotone binary functions defined on $\widehat{H}$ by means of generalized Hansel's method,

- partitioning $\widehat{H}$ into binary cube like vertical extensions of its upper homogeneous elements and applying Hansel's method for them,
- identifying monotone functions defined on $\widehat{H}$ with use of additional information on location of their upper units through L_{min} and L_{max}.

The choice of concrete resource set depends on requirements of certain applications.

Conclusion

Algorithmic resources are considered for elaboration and identification of monotone functions. Current research proposes two new components - partitioning the multi-valued cube through binary cube like vertical extensions of its upper homogeneous elements; and learning upper homogeneous area through the analogous partitioning. The choice of concrete resource depends on requirements of certain application.

Bibliography

[M, 1959] V. Mickeev. On sets, containing maximal number of pair wise incomparable Boolean vectors, Prob.Cyb, 2, 1959.

[K, 1965] V. Korobkov. On monotone functions of algebra of logic, Prob.Cyb, 13, 1965.

[K, 1981] A. Korshunov. On the number of monotone Boolean functions, Prob.Cyb, 38, 1981.

[H, 1966] G. Hansel. Sur le nombre des fonctiones booleennes monotones de n variables, C.R.Acad.Sci., Paris, 262, N.20, 1966.

[T, 1979] G. Tonoyan. Cain partitioning of n-cube vertces and deciphering of monotone Boolean functions, Journal of Computational Mathematics and Math. Physics, 1979, vol. 19, N.6.

[A, 1976] V. Alexeev. On deciphering of some classes of monotone many valued functions, Journal of Computational Mathematics and Math. Physics, 1976, vol. 16, N.1.

[K, 1978] Katerinochkina N., On sets, containing maximal number of pair wise incomparable n-dimensional k-valued vectors, Mathematical notes, v. 24, no, 3, Sept. 1978.

[AS, 2001] L. Aslanyan, H. Sahakyan. On the boundary cases of partitioning of subsets of the n-dimensional unit cube, "Computer Science & Information Technologies" Conference, Yerevan, September 17-20, 2001

[S, 2006] H. Sahakyan. Numerical characterization on n-cube subset partitioning, submitted to Discrete Applied Mathematics.

Authors' Information

Hasmik Sahakyan – *e-mail: hasmik@ipia.sci.am*

Levon Aslanyan – *e-mail: lasl@sci.am*

Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia

DYNAMIC DISTRIBUTION SIMULATION MODEL OBJECTS BASED ON KNOWLEDGE

Alexander Mikov, Elena Zamyatina, Konstantin Osmehin

Abstract: *This paper presents the process of load balancing in simulation system Triad.Net, the architecture of load balancing subsystem. The main features of static and dynamic load balancing are discussed and new approach, controlled dynamic load balancing, needed for regular mapping of simulation model on the network of computers is proposed. The paper considers linguistic constructions of Triad language for different load balancing algorithms description.*

Keywords*: Distributed calculations, distributed simulation, static load balancing, dynamic load balancing, expert systems*

ACM Classification Keywords*: I.6 Simulation and Modeling I.6.8 Types of Simulation - Distributed: I.2 Artificial Intelligence I.2.5 Programming Languages and Software - Expert system tools and techniques*

Introduction

The complicate problem solutions with employment of high performance computers are very vital now. This is an actual problem for the simulation too. It is imposed by inevitability of increasing computation capacity and simulation running optimization on the one hand [1,2,3]. The simulation model objects are distributed across the calculation nodes in cluster or in local area network (LAN) or in wide area network (WAN) and interact by passing messages from one to another. From the other hand, it is necessary to use distributed simulation in order to combine already designed simulation models or because of inevitability to organize the conjoint work of some investigators participating in a common simulation experiment.

The computing environment heterogeneity and simulation model heterogeneity can be the reason of load balancing violation during simulation run. The heterogeneity of calculating environment is associated with varied capacity of calculating nodes and diverse capacity of communication lines. The simulation model heterogeneity is based on the fact that some of objects wait some event occasion but the other are in the state of calculating almost all the time initiating appropriate events. Besides, some objects may communicate with high intensity whereas the communication between the other of them is seldom the case. So unbalance would negate the benefit of distributed calculation.

That is why a demand arose in algorithms designing and programming tools development needed for load balancing preservation. It was correctly reasoned for distributed simulation and for distributed calculation as a whole [2,3].

The load balancing algorithm must be optimal for any model with any structure and any mechanism of time advancement. But it is not the easy problem. A considerable number of investigators attempted to design load balancing algorithms [2, 3]. But these algorithms were designed only for specific problems or they changed the original program code [2]. From the above discussion it appears that the new approach to load balancing problem is needed. Authors [5] suggest this approach: controlled load balancing algorithm based on knowledge. The paper considers the target setting, the summary review load balancing algorithms (and the load balancing algorithms used in distributed simulation) and the architecture of load balancing programming tools. Furthermore the authors propose the architecture of a load balancing subsystem for distributed simulation system Triad designed by them and suggest linguistic constructions for load balancing algorithms description.

Load balancing of computing nodes during simulation run

Most often the components of system being simulated are presented in PDES (PDES-Parallel Discreet Event Simulation) as logical concurrent processes ($LP_i, i=1 \div n$). Logical processes are allocated on calculating nodes (nodes of cluster or network) and interact by passing messages from one to another. One can find out the conflict of good balanced logical processes allocation and low speed of message interchange (communication lines have low capacity or these lines are overloaded) during simulation run. One can observe another situation: the time which is needed for simulation objects communication almost equal to zero, and in the same moment some computers (processors) are waiting the work and the other ones are overloaded. On the other hand good balanced system may demand a great amount of time for communication. So we can conclude that the strategy of load balancing system supposes regular workload of processors and not overloaded communication.

One must differ static and dynamic load balancing. Static load balancing must be fulfilled before the simulation experiment using the prior runtime data (in SPEEDES[2], for example) or structural characteristics of simulation model (Triad.Net). Triad.Net simulation system proposes to allocate a simulation object and substructure of this object (the structure of Triad simulation system is hierarchical) in one calculating node. Moreover Triad.Net load balancing subsystem finds out cliques and allocates them in one computing node too in order to reduce communication time.

But preliminary allocation of simulation objects is not effective always.

This is explicable on the basis of follow facts:

- A simulation model can be changed in the course of simulation run because of scheduling new events, new processes appearance, terminating some processes;
- A calculating environment can be changed because one or several processors (or computers) are failed;

– A processor (or computer) is used to calculate not only a simulation model but it may carry out another calculating works and portion of these works may increase with the time.

In any case, the benefit of distributing logical processes between calculating nodes before the simulation experiment very often cannot be found out. A dynamic load balancing supposes distribution of logical processes during simulation experiment. One can single out some stages in a dynamic load balancing process: an assessment of workloads, an initiation of load balancing, decision making of expediency of load balancing, migration of workloads from one node to another.

Automatic load balancing process makes a decision on workload (object of simulation model) migration from one node to another relying on data collected during simulation experiment. These data are saved in the data base and include data of two types:

– a data about simulation model (the frequency of exchange between simulation model objects, the number of objects allocated in the same calculating node, duration of some logical processes);
– a data about computing environment (computing load of calculating node, the load of communicating lines, calculating nodes lay-up and so on).

Besides, it is very important to have information about communication of processes (the topology of exchanges). One can assess the workload of a processor analytically basing on knowledge about simulation model behavior. Another way to assess workload is to make metrology during the simulation run. The most of modern computers contain time counters needed to evaluate the execution time for every application. There are special programming tools for data collection.

After that it is necessary to find out the load balancing violation and to make a decision about migration of objects from one calculating node to another because the frequent migration may reduce a benefit from load balancing.

User can find out unbalance using one of two ways:

– *Synchronously:* all the calculating nodes interrupt their run in some definite moments of synchronization, and load unbalance can be defined by comparison of workload of diverse computing nodes.
– *Asynchronous:* every calculating node has the history of its workload. So there is no single moment of synchronization needed for unbalance determination. Background process functioning across simulation model during simulation experiment evaluates workload.

Load balancing subsystem makes a decision about simulation nodes migration:

– In centralized manner which means that some special computer carries out data collection (data includes information about computing environment state) and makes a decision about the migration of the simulation objects.
– In distributed manner which means that each calculating node carries out its own load balancing algorithm. According to this algorithm simulation model objects could be transferred only to adjoining nodes.

The last is the stage of object migration and it is necessary to provide object integrity.

So a great amount of algorithms and strategies of load balancing exists. Nevertheless there are common features in architectures of various subsystems and common stages in various load balancing algorithms. Let us list load balancing subsystem components: first of all it is a component providing the distributed simulation model state assessment and estimation of computing environment characteristics, the control program which chooses logical processes and makes a decision about the moment of migration, the program tools providing the migration of the objects from one calculating node to another, data base with information about simulation model objects and computing environment and at last subsystem of visualization. This subsystem must show the simulation model objects mapping on computing environment, the scheme of communication, simulation model changes and computing environment transformation.

And now let us consider simulation model and load balancing subsystem in Triad.Net more precisely (Triad.Net is new distributed simulation system, it is advanced version of CAD and simulation system Triad).

Simulation Model in Triad

Simulation model in Triad.Net is represented by several objects functioning according to some scenario and interacting with one another by sending messages. Simulation model [4] is μ={STR,ROUT,MES} and it consists of three layers, where STR is a layer of structures, ROUT – a layer of routines and MES – a layer of messages appropriately. The layer of structure is dedicated to describe the physical units and their interconnections, but the layer of routines presents their behavior. Each physical unit can send a signal (or a message) to another object. So, each object has the input and output poles (P_{in} – input poles are used to send the messages, P_{out} – output poles serve to receive the messages). A message of simple structure can be described in the layer of routines, but a message of the complex one – only in the layer of messages. Many objects to be simulated have a hierarchical structure. So their description has a hierarchical structure too. One level of the structure is presented by graph P = {U, V, W}. P-graph is named as graph with poles. V is a set of nodes presenting the physical units of object to be designed, W – a set of connections between them, U – a set of external poles. The internal poles are used for information exchange within the same structure level; in contrast, the set of external poles serves to send signals (or more complex information) to the objects situated on higher or underlying levels of description. Special statement *out* (out <message> through <name of pole>) is used for message sending. A set of routines is named ROUT- routine layer.

Special algorithms – routines – define the behavior of a physical unit and are associated with particular node of graph P = {U, V, W}. Each routine is specified by the set of events (E-set), the linearly ordered set of time moments (T-set), the set of states {Q-set}. State is specified by the local variable values. Local variables are defined in routine. The state is changed if only an event occurs. One event schedules another event. So simulation system Triad.Net is a discrete-event one. Routine (as an object) has input and output poles (Pr_{in} and Pr_{out}). An input pole serves to receive messages, output – to send them. One can pick out input event e_{in}. All the input poles are processed by an input event, an output poles – by the other (usual) event.

Simulation system Triad.Net is advanced Triad system, but it is the distributed/parallel one. Conservative and optimistic algorithms were designed in Triad.Net.

Special subsystem called analysis subsystem includes special objects of two types. Some of them are named "information procedures". Information procedures examine simulation results and inspect the simulation run. Information procedures are "connected" to nodes or, more precisely, to routines, which describe the behavior of particular nodes during simulation run. Information procedures act as monitors of a simulation model. Each information procedure can watch several simulation model objects in same time and in any time of simulation run. The second type of analysis subsystem objects is "*conditions of simulation*". "*Conditions of simulation"* is a special linguistic construction defining the algorithm of investigation (it includes a list of information procedures dedicated to examine a simulation model during concrete simulation run). In order to examine the simulation model component from another point of view one can use other conditions of simulations with the new list of information procedure names. Besides, conditions of simulation define the condition of simulation run termination.

It must be remembered that the simulation model in Triad.Net is not static. There is a special type of variable – type "model" in Triad language and several operations with the variable type "model". These operations are defined for the model in general and for each layer. For example, one may add or delete a node, add or delete an edge (arc), poles, union or intersection of graphs. Routine layer permits to add or delete any event, layer of messages – to add or delete types or selectors. Besides, one or another routine (routine layer) using some rules can be assigned to the node (structure layer). The behavior of the object associated with this node would be changed. Besides, one does not need to retranslate the model. But we see that the structure of simulation model may be changed, and so we have an additional reason of dynamic load balancing procedure existence.

Load balancing subsystem in Triad.Net

Load balancing subsystem is dedicated to optimal distribution simulation model across calculating nodes (in multiprocessor computer or in network) and consequently to enhance the performance of these computers. Load balancing it is a problem of non isomorphic vertex-connected graphs mapping B: TM $\rightarrow$ NG, where TM – a set of graphs of simulation models, NG – a set of graphs – computer network configurations. Graph G $\in$ NG, G = {C, Ed}, can be defined by a set of calculating nodes C and a set of edges Ed (edges ED are associated `with

communication lines). One can consider NG as a super graph, containing all eventual (admissible) graphs G as sub graphs. Graph $M \in TM$, $M = \{U, V, W\}$ represents simulation model.

Let us consider three kinds of load balancing [5]: static B_s, dynamic (automatic) B_a and dynamic (controlled) B_c.

This is a generally recognized result – the existence of sub graph $G \subset NG$ which is isomorphic to simulation model M. But usually this graph does not exist, so it is valuable to find closely corresponding graph.

In regards to dynamic balancing B_a graphs G and M are considered to be loaded. The nodes of the first graph have a parameter – performance, edges – data rate. The characteristics of nodes in the second graph – time complexity, the characteristics of edges – the intensity of a traffic flow.

The weights of nodes and edges in graph NG (and consequently, any subgraph of this graph) are considered to be known. The corresponding graph M parameters must be defined during the simulation run. The "bottle neck" of simulation model and computer system is determined in accordance with some algorithm, and migration of the simulation model objects without interruption of simulation run is fulfilled. The automatic load balancing algorithm may be described in Triad language.

One can define better distribution simulation model objects across the nodes of graph G for the following simulation experiments using data collected by information procedures during previous simulation runs and genetic algorithms. Automatic dynamic balancing algorithm uses antecedents of the simulation run in order to plan the future simulation run. But the actual simulation model behavior may not correspond to this prediction.

Really it is well known that there is no a considerable gain in performance when employing automatic load balancing algorithm. Only the designer of simulation model knows the behavior of simulation model in specific situation: for example, the simulation model designer knows that the intensity of a flow of requests after 600 units of simulation time increases. Another example: the exchange intensity between two nodes becomes very high after 300 units of simulation time. Load balancing subsystem must map these two nodes of graph M in the same calculating node (graph G) or in two adjoining nodes of graph G with powerful communication line. So it becomes clear, that the efficiency of load balancing may increase if we develop some special tools to control the load balancing process.

There is a new approach of implementation of controlled dynamic load balancing based on knowledge in Triad.Net. Controlled dynamic load balancing subsystem includes expert component and information procedures developed by a model designer (non standard information procedures in other words). Expert component consists of optimization rules defined by the author of the given model (or of class of models). Non standard information procedures are intended to estimate the events (or conditions) of rule applications.

Load balancing subsystem architecture

So, knowledge-based load balancing subsystem includes:

- Expert system with knowledge base, rules editor, inference engine and module of explanation. Knowledge base consists of rules for optimal distribution of simulation model objects across the calculating nodes.
- Simulation model and computation environment analysis subsystem. Analysis subsystem consists of information procedures to collect data on simulation model objects behavior (the frequency of interchanges among the simulation model objects, the frequency of event occurrence and so on) and to collect data on computational environment (flow capacity of communication lines, workload of computers).
- Subsystem for simulation model and calculating environment visualization. Subsystem of visualization present diagrams and plots. User has an opportunity to choose information representation enjoying these or other information procedures. Besides subsystem must to represent simulation model mapping on calculating environment.
- Migration subsystem which carries out simulation model migration from one computing node to another.

As it was mentioned above special objects – information procedures observe simulation model objects, more precise, they observe local variables changing, events occurrences and fix the facts of messages sending and receiving. Information procedures can analyze and compare local variables of different routines in the same moment during simulation run. Subsystem of analysis includes standard information procedures. Besides user can design his own original one using Triad language.

The information procedures visualize characteristics of simulation model objects during simulation run. If the values of some characteristics are equal to some limit values the information procedures send a message to expert component. Expert component carries out some operations on graph G (this graph represents the structure of simulation model) mapped on graph M – graph of calculating environment.

The rules could be formulated as so:

- If there are more than 2 active objects in i-th computing node but j-th node is free then migrate one of simulation model object to j-th free;
- if there are two objects V_i and Vj in simulation model and they exchanges by messages and if there is communication line E_{dk} which communication capacity is rather high then it is advisable to map arc w_{ij} on E_{dk};
- if one can observe high intensity of two simulation model objects (V_i and V_j) exchanging by messages and there is free computing node D_i then it is advisable to migrate (V_i and V_j) in D_i

So rules imply operations on graph G. Rules are productions such as «if then else…» and could be described by Triad language. One can describe the automatic load balancing algorithm by Triad.Net too.

Language constructions in Triad for load balancing description

Let us show a little fragment of Triad program describing automatic load balancing algorithm. It is an algorithm from distributed simulation system SPPEEDES [3]. Following this algorithm one must choose the most loaded computation node in the case of unbalance occurrence. Further it is necessary to choose one or more simulation model objects which are situated on this overloaded node and to carry out the migration of chosen objects. The choice of objects is random.

Simulation model description begins with key word model (Triad language) and the end of Triad-program is marked with key word *endmod*. The model description includes a description of the layer of structure (*structure …endstr*), a description of layer of routines (*routine … endrout* (the behavior of simulation model objects)) and a layer of messages *message…endmsg* (messages with rather complicate structure). Let us suppose that the structure of simulation model G is represented a graph with nodes A,B,C,D,E,F (a complete graph with nodes A,B,C,D and two nodes E and F adjoining node D. This nodes a connected with edge.

model Mod1;

var graph G; *structure* S *def*…G1:=compl(A,B,C,D)+node(E)+Node(F)+edge(E<>F)+edge(D<>E)+ edge(D<>F) …endstr

…G:=S; (* we described the structure of simulation model, compl – graph constant «a complete graph»*)

It is necessary to use structure layer of Triad language to describe calculating environment.

….*Var graph* M; (* Calculating environment description, calculating environment has a star topology, network consists of three computers with computer named O in the center of star*)

structure S1 def … G2:=*star*(3)(O,P,R,S) …*endstr*; M :=S1: …(*структура ВС*).

Now we shall consider the description of a static load balancing. Let us use a procedure which carries out the preliminary mapping of simulation model objects. This procedure must be executed before a simulation run. So it is advisable to allocate the call statement of this procedure in language construction "conditions of simulation". This part of program code executes before other one. So load balancing procedure may be represented as follow:

procedure Static_Load_Balancing (*in ref node* G1, *ref node* V,W,X,Y,Z) *def* X:=G1; Y:=V; Z:=W;*endproc*

The algorithm of automated load balancing is presented as procedure *Automated_Load_Balancing.* Following this algorithm a simulation model object in one of computing nodes is selected by accident. The name of computing nodes is one of a parameters of procedure mentioned above. Selected simulation object is migrated to another computing node. The name of this node is an input parameter of procedure too. The names of nodes are defined as a most loaded and least loaded by standard information procedure. Let it be M.P and M.R.

procedure Automated_Load_Balancing (in ref node X,Y) def

set U *of node* (G.A,G.B,G.C,G.D); *ref node* Q; Q:= *random*(U); Migrate(Q) *from* (X) *to* (Y)

endproc

The linguistic construction "conditions of simulation" may include call statement of procedure Automated_Load_Balancing too (in main part after the call statements of information procedures).

Conditions of simulation Running …

Initial … Static_Load_Balancing(compl(G.A,G.B,G.C,G.D,G),G.E,G.F,M.P,M.R,M.S) …*endi*

….(*call statements of standard information procedures which are used for data collection*)…

(*call statement of automated load balancing procedure*)

Automated_Load_Balancing (M.P,M.R); *processing* ….*endproc*;

Endcond

Conclusions

So this paper describes the load balancing subsystem architecture and linguistic construction for their description in Triad language. These linguistic constructions are rather convenient for original load balancing algorithms description. This algorithms may be designed by investigators themselves and it may be more adequate for the specific simulation model behavior during simulation run. Load balancing subsystem in Triad.Net includes not only programming tools supporting static and automatic dynamic load balancing but programming tools for implementation of controlled dynamic load balancing based on knowledge too. The rules must be defined by investigator knowing the behavior of specific simulation model (or already known rules for similar simulation model may be modified).

Aknowledgements

This paper is supported by RFFI (RF) under project 07-07-412-a

Bybliography

1. [Fujimoto, 2003] Fujimoto R.M. Distributed Simulation Systems. In Proceedings of the 2003 Winter Simulation Conference S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds. pp. 124-134
2. [Wilson, 1998] Wilson L.F. and Wei Shen. Experiments In Load Migration And Dynamic Load Balancing In Speedes. Proceedings of the 1998 Winter Simulation Conference.D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds, pp.590-596
3. [Zheng, 2005] Zheng G. Achieving High Performance on Extremely Large Parallel Machines: Performance Prediction and Load Balancing; in Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005,165p. Доступно на сайте: http://charm.cs.uiuc.edu/
4. [Mikov, 1995] Mikov A.I. Simulation and Design of Hardware and Software with Triad// Proc.2nd Intl.Conf. on Electronic Hardware Description Languages, Las Vegas, USA, 1995. pp. 15-20.
5. [Миков, 2005] Миков А.И.., Замятина Е.Б., Осмехин К.А. Метод динамической балансировки процессов имитационного моделирования. В кн. «Материалы Всероссийской научно-технической конференции «Методы и средства обработки информации МСО-2005». М.: Изд-во МГУ, 2005, стр.472-478.

Authors Information

Alexander Mikov *– The Institute of Computing, the director, Full professor, RF, Krasnodar, Aksayskaya, 40/1-28; e-mail: alexander_mikov@mail.ru*

Elena Zamyatina *– Perm State University, associate professor, Computer Science Department, RF, Perm, 614017, Turgeneva, 33,. 40; e-mail: e_zamyatina@mail.ru*

Konstantin Osmehin *– Perm State University, post graduator, Computer Science Department, RF, Perm, 614017, ул. Gashkova, 28-12; e-mail: kosmehin@lukoilperm.ru*

GROUND OF MODEL FOR THE GENERALIZED CRITERION FORMING AT DIFFERENTIAL DIAGNOSTICS OF DERMATOLOGICAL DISEASES

Anatoly Bykh, Elena Visotska, Olga Kozina, Anna Tikhonova, Andrey Porvan, Alexander Zhook

Abstract: *The basic methods of decisions making in multi-criterion conditions are considered, from which the method of the weighed total for calculation of diagnostic indexes significance in differential diagnostics of dermatological diseases is chosen.*

Keywords: *dermatology, differential diagnostics, method of the weighed total, multi-criterion task.*

ACM Classification Keywords*: J.3 Life and medical scinces: Medical information systems*

Introduction

The external cover of human body – skin – is the difficult arranged organ which executes protective and physiological functions. A skin is constantly exposed to influence of various physical, chemical, biological factors of external environment that is why pathological processes are developing in its. Quite often rise of dermatopathology is related to influence of endogenous factors or with rise of allergic reactions, or in other words with the increased sensitiveness of organism in relation to some matters [Беренбейн, 2000]. All this, and also wide prevalence of skin's diseases among the population is cause to actuality of differential diagnostics problem of dermatological diseases.

Urgency of the problem

Differential diagnostics of skin illnesses is based on complex estimation of anamnesis data, features of morphological changes of skin and mucous membranes, common state of patient and his subjective feelings. In everyday clinical practice the dermatologists quite often run into a situation, when discerning of reliable clinical diagnosis is difficult. It is explained by as by plenty of possible clinical displays of skin's pathology, by variety of the conducted laboratory researches, by existence rarely meeting symptoms and pathologies, so that many symptoms are characterizing diseases with different etiology and pathogenesis. A different importance of the symptom is not equal for different diseases, that still more complicates differentiation of diagnoses [Ананьев, 2005]. So in order to define a correct diagnosis it is necessary to take into account plenty of different diagnostic information, or in other words it is necessary to decide a multicriterion task.

Determination of significance of diagnostic indexes

Normative (formal) methods, which assume that an expert know the definite rational method of correct decision choice, are basic methods of decisions making in multicriterion conditions [Брахман, 1984]. Depending on the role of an expert in forming and grounding of importance of alternative diagnostic decision all formal methods divide by axiomatic, lines, methods of compensation, methods of thresholds of incomparableness and man-machine methods. From these methods for task of differential diagnostics of dermatological diseases direct methods are most useful. Due to such group of direct methods an expert can formulate of resulting significance of the given symptom for each disease as dependence on its estimations on private criteria without the theoretical grounds, and the parameters of this dependence (weight and expressed of symptom) are formed directly by the method of expert estimations [Кац, 2004]. For every disease there is the set of diagnostic indexes which fully describe one. A primary task is arrangement of criterions due to its importance.

The method of lexicographic arrangement of criteria is most simple. Thus the diagnosis which has more high estimation on the most essential criterion is most probably existing, that is far not always is true. Also this method does not allow take into account equivalent of some criterions.

The generalized importance as quantitative estimation of diagnostic index preference exists as most reliable. To find this value it is necessary to ground a deciding rule, due to which importance of index in space of private criteria $f(x_i)$ is formed on, i.e. to decide the tasks both of structural and parameters identification of function of importance:

$$U(x) = G\left[f_i(x)\right], \tag{1}$$

where G is operator determining the type of dependence,

$f_i(x)$ are private criterions characterizing expressed of one or another diagnostic index for patient,

$i = \overline{1,n}$,

n is number of diagnostic indexes.

Structural identification of any mathematical model, including functions of importance, supposes the necessity of decision of two associate tasks:

– selection of meaningful factors which influence on output data of these models;

– determination of structure or type of operator establishing a connection between input and output model data.

Parameters identification consists of determination of concrete quantitative parameters of model.

The decision of task of structural identification of model is related to formulation and verification of some hypothesis. In the case of differentiation of skin's diseases the type of function of generalized importance of diagnostic index $x \in X$ is determined by the values of private criteria $f_i(x), i = \overline{1,n}$, which characterize the decision, and in common case these descriptions are not equivalent, i.e. they have a different weight for an expert-physician. It does not eliminate that in special case weight of some criteria can be identical. It is also necessary to take into payment that diagnostic indexes are heterogeneous. Every diagnostic index has the dimension, intervals and scales of measuring, i.e. indexes are not comparable. Then a formula (1) can be written as:

$$U(x) = G\left[k_i, f_i(x)\right], \tag{2}$$

where,

k_i are isomorphic parameters i.e. parameters with mutual univocal correspondence, bringing heterogeneous private criterions $f_i(x)$ into the single metrics and taking into account their weight.

The isomorphic parameters are set directly by an expert-physician in a numeral view by the expert estimations method.

Determination of structure – in other words type of operator establishing a connection between input (for example, weight and expressed of symptom) and output (for example, importance) data is a next step. For authentication of operator's type additive and multiplicative forms are most widely used [Петров, 2003].

Because the isomorphic parameters are constants multiplicative form of G operator authentication

$$U(x) = \prod_{i=1}^{n} k_i f_i(x) \tag{3}$$

is possible to represent in view

$$U(x) = \prod_{i=1}^{n} k_i \prod_{i=1}^{n} f_i(x). \tag{4}$$

$\prod_{i=1}^{n} k_j$ is a permanent scale multiplier and consequently all criterions become equally important, that is not true.

Thus a multiplicative form does not allow taking into consideration information about private criterions preference.

Inadmissible of complete indemnification of one parameters of other is another feature of multiplicative form. If even one of multipliers is equal 0, all multiplicative function takes on a zero value. And in the case of differential diagnostics of dermatological diseases the unexpressed at the patient of one or another diagnostic index which usually are characteristics for this pathology does not eliminate this pathology from consideration. It shows that

the multiplicative form of G operator identification is inapplicable at determination of diagnostic indexes importance in dermapathology.

In contrary of multiplicative form, additive form of operator dependence index's importance from its weight and expression for a patient

$$U(x) = \sum_{i=1}^{n} k_i f_i(x) \tag{5}$$

does not have the afore-named demerits. However formula (5) is correct only in case when k_i are considering both as weight of i-th diagnostic index and the coefficients of isomorphism. In common case determination of such coefficients is complex difficult task. This trouble can be overcame if to represent an additive function in a form

$$U(x) = \sum_{i=1}^{n} \delta_i f_i^H(x), \tag{6}$$

where δ_i are dimensionless coefficients characterizing weight of i-th diagnostic index for concrete pathology for which limitations are executed for:

$$0 \le \delta_i \le 1, \tag{7}$$

$$\sum_{i=1}^{n} \delta_i = 1; \tag{8}$$

$f_i^H(x)$ are normalized, i.e. the private criterions represented in isomorphic view [Ларичев, 1996]. In the case of dermatological differential diagnostics, estimations of emphasis of one or another diagnostic index at a patient can be considered as criterions $f_i^H(x)$. Isomorphism of private criterions means that they have an identical dimension and interval of possible values. Cause on determination δ_i are dimensionless coefficients, a dimension $f_i^H(x)$ must coincide with a dimension of $U(x)$, i.e. $f_i^H(x)$ characterizes local importance of i-th diagnostic index x for concrete pathology:

$$f_i^H(x) = p_i(x). \tag{9}$$

In accordance with (9) a function (6) will take view:

$$U(x) = \sum_{i=1}^{n} \delta_i p_i(x). \tag{10}$$

Method of private criterions importance determination based on application of dependence operator in additive form named the method of the weighed sum.

As well as for all direct methods, for the method of the weighed total rigorisms to the experts are characteristic, especially on the initial stages of work at forming of expert estimations.

The model of evaluation (10) is just only in case when the weight coefficients δ_i of private criterions $f_i^H(x)$ are set by the exact quantitative values. As it was already marked, experts-physicians are such data carriers, and it means that some procedures of its receipt are needed, i.e. parametrical identification of model is required. In order to decide this task principle of associative, method of communications between symptoms and syndromes, method of causal communications of symptoms with syndromes on contiguity in time is applied. For the decision this problem the most effective is complex application of different methods of analysis and account of knowledge of experts-physicians.

If on some reasons receipt of exact quantitative information about diagnostic index it is not possible, in general case the evaluation has to be produced in the conditions of greater or less degree of uncertainly. It is possible in the case of differential diagnostics in the group of rare insufficiently known diseases of skin. In this case the general model of determination of generalized importance of index $x \in X$ has the view:

$$U(x) = G\left[J(\alpha_i), p_i(x)\right], \tag{11}$$

where,

$J(\alpha_i)$ - information about mutual importance of private criterions.

Conclusion

It is possible to do a conclusion, that a plenty of methods of decision of the multicriterion tasks oriented to the concrete problem situations is presently developed. Application of one or another method is conditioned by the expert's role in forming and ground of alternatives value.

For estimation of significance of diagnostic indexes at differential diagnostics of dermatological diseases the additive weighed total method is recommended to use.

Adduction of all indexes to isomorphic view is the feature of this method.

Application of this method will allow to and to improve quality of differential diagnosis discerning and facilitate working tasks of doctor-dermatologist.

Bibliography

[Беренбейн, 2000] Б.А. Беренбейн, А.А. Студницин. Дифференциальная диагностика кожных болезней. Руководство для врачей. М.: Медицина, 2000. 712с.

[Ананьев, 2005] О. Л. Ананьев, Е. В. Анисимова, Н. В. Иваничкина и др. Кожно-венерические заболевания. Полный справочник. М.: Эксмо, 2005. 672с.

[Брахман, 1984] Т.Р. Брахман. Многокритериальность и выбор альтернативы в технике – М.: Радио и связь, 1984. 287с

[Кац, 2004] М.Д. Кац. Использование искусственного интеллекта для разработки методов дифференциальной диагностики внутри групп трудно различимых заболеваний. Клиническая информатика и телемедицина. 2004. №1. С.86 – 89.

[Петров, 2003.] Э.Г. Петров, М.В. Новожилова, И.В. Гребенник, Н.А. Соколова. Методы и средства принятия решений в социально-экономических и технических системах. Херсон: ОЛДІ-плюс. 2003.380с.

[Ларичев, 1996] О.И. Ларичев, Е.М. Мошкович, Ю.П. Иванилов. Количественные методы принятия решений. – М.: Физмалит, 1996. 208с.

Authors' Information

Anatoly Bykh *– Doctor of Physics and Mathematics, professor, Head of Biomedical Electronics Dept. of KNURE*

Elena Visotska *– PhD, lecturer of Biomedical Electronic Devices and Systems Dept. of KNURE*

Olga Kozina *– PhD, lecturer of Computers and Programing Department of National Technical University 'KPI'*

Andrey Porvan *– engineer of Biomedical Electronic Devices and Systems Dept. of KNURE*

Anna Tikhonova *– engineer of Biomedical Electronic Devices and Systems Dept. of KNURE*

Alexander Zhook *– student of Biomedical Electronic Devices and Systems Dept. of KNURE*

Kharkov National University of Radio Electronics, Ukraine, 61166, Lenin Avenue, 14, Biomedical Electronic Devices and Systems Department, e-mail: diagnost@kture.kharkov.ua

TABLE OF CONTENTS OF VOLUME 15, NUMBER 3