

**TECHNICAL
TRANSACTIONS**

**AUTOMATIC
CONTROL**

**ISSUE
3-AC (11)**

**YEAR
2013 (110)**

**CZASOPISMO
TECHNICZNE**

AUTOMATYKA

**ZESZYT
3-AC (11)**

**ROK
2013 (110)**



**WYDAWNICTWO
POLITECHNIKI
KRAKOWSKIEJ**

TECHNICAL TRANSACTIONS

AUTOMATIC CONTROL

ISSUE 3-AC (11)
YEAR 2013 (110)

CZASOPISMO TECHNICZNE

AUTOMATYKA

ZESZYT 3-AC (11)
ROK 2013 (110)

Chairman of the Cracow
University of Technology Press
Editorial Board

Chairman of the Editorial Board

Jan Kazior

Józef Gawlik

Przewodniczący Kolegium
Redakcyjnego Wydawnictwa
Politechniki Krakowskiej

Przewodniczący Kolegium
Redakcyjnego Wydawnictw
Naukowych

Scientific Council

Jan Błachut
Tadeusz Burczyński
Leszek Demkowicz
Joseph El Hayek
Zbigniew Florjańczyk
Józef Gawlik
Marian Giżejowski
Sławomir Gzell
Allan N. Hayhurst
Maria Kusnierova
Krzysztof Magnucki
Herbert Mang
Arthur E. McGarity
Antonio Monestiroli
Günter Wozny
Roman Zarzycki

Rada Naukowa

Automatic Control Series Editor

Piotr Kulczycki

Redaktor Serii Automatyka

Section Editor
Native speaker
Cover Design

Dorota Sapek
Tim Churcher
Michał Graffstein

Sekretarz Sekcji
Weryfikacja językowa
Projekt okładki

Pierwotną wersją każdego Czasopisma Technicznego jest wersja on-line
www.czasopismotechniczne.pl www.technicaltransactions.com

Technical Transactions
series *Automatic Control*

vol. 3-AC/2013

Information Processing for Automatic Control

Eds. Volodymyr Samoty, Mieczysław Zajac, Piotr Kulczycki

Editorial Board

Editor-in-Chief:

Piotr Kulczycki, Poland

Editorial Board:

Krassimir Atanassov, Bulgaria
Valentina Emilia Bălaș, Romania
Tadeusz Burczyński, Poland
Antoni Leon Dawidowicz, Poland
Guy De Tré, Belgium
Józef Gawlik, Poland
Alexander Gegov, Great Britain
Abdellali El Aroudi, Spain
Jerzy Józefczyk, Poland
Janusz Kacprzyk, Poland
Jerzy Klamka, Poland
Erich Peter Klement, Austria
László T. Kóczy, Hungary
Heikki Koivo, Finland
Józef Korbicz, Poland
Zdzisław Kowalczyk, Poland
Krzysztof Kozłowski, Poland
Krzysztof Malinowski, Poland
Radko Mesiar, Slovakia
Andrian Nakonechnyy, Ukraine
Witold Pedrycz, Canada
Leszek Rutkowski, Poland
Dominik Sankowski, Poland
Václav Snášel, Czech Republic
Ryszard Tadeusiewicz, Poland
Lipo Wang, Singapore
Rafael Wisniewski, Denmark

Executive Editors:

Dominika Gołuńska, Poland
Piotr A. Kowalski, Poland
Szymon Łukasik, Poland

MIECZYŚLAW ZAJĄC*

CONTROL OF ELECTRIC DRIVE BY MEANS OF INVERSE DYNAMICS

STEROWANIE NAPĘDU ELEKTRYCZNEGO METODĄ DYNAMIKI ODWROTNEJ

Abstract

This paper presents a method for positioning an electric drive with an elastic mechanical part by applying the inverse problem of dynamics. The presented assumptions take into account technological requirements and limitations of dynamic variables. The desired trajectory of the mechanical part of the electromechanical system has also been determined. On this basis, an algorithm for determining the control voltage waveform is proposed.

Keywords: electromechanical systems, control, inverse dynamics

Streszczenie

W artykule przedstawiono metodę pozycjonowania napędu elektrycznego z elementem elastycznym z zastosowaniem zagadnienia odwrotnego dynamiki. Sformułowano założenia i uwzględniając wymagania technologiczne oraz ograniczenia wielkości dynamicznych, określono przebieg pożądanej trajektorii układu elektromechanicznego. Na tej podstawie opracowano algorytm obliczeniowy w celu określenia przebiegu napięcia sterującego.

Słowa kluczowe: układy elektromechaniczne, sterowanie, odwrotne zagadnienie dynamiki

* D.Sc. Ph.D. Mieczysław Zajęc, e-mail: mzaj@pk.edu.pl, Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

Nomenclature

x	–	position of the winding engine mass [m],
$x^{(1)}$	–	velocity of the winding engine mass [m/s],
$x^{(2)}$	–	acceleration of the winding engine mass [m/s ²],
x_m	–	position of the point on the circumference of the drive wheel [m],
u_s	–	control signal calculated by the controller [V],
ε	–	correction signal (output of the correction block) [V],
ω	–	angular velocity of the motor [rad/s].

1. Introduction

Fast dynamic algorithms are a prerequisite in the positioning of real-time control systems of flexible manipulators and winding machines [2]. Analysis of elastic vibrations in such systems is usually carried out by investigating the drive transmission system. In general, the drive transmission system consists of a set of masses with elastic and damping connections.

In a multi-mass system, the most significant oscillation frequencies should be selected. If a system has only one significant natural frequency, its dynamic properties may be reproduced with a two-mass system with a fair degree of accuracy. Flexibility in mechanical joints results in a dynamic torque variable component which makes it difficult to obtain the desired dynamic characteristics [1].

As a result, the system can only imperfectly reproduce a predetermined trajectory in the state space. These imperfections can be partly compensated for by classical methods, i.e. by adaptation of the gain factor feedback. [13]. An example of applying nonlinear feedback methods to reducing overshoots in the positioning system when the actuator reaches the desired position is shown in [9]. Another approach is presented in [8], where the authors discuss the inverse dynamics velocity control of a direct drive manipulator based on the sliding mode compensation technique. The compensated inverse dynamic velocity control scheme is robust against the detrimental effects of model uncertainties and exhibits robust tracking performance of the desired velocity trajectories. In [7], a discrete-time sliding mode control strategy combined with an inverse dynamics approach to motion control of robot manipulators is proposed.

When positioning such systems, three key issues must be addressed – determination of the control system structure, calculation of parameter values and determination of control variable waveforms ensuring the desired trajectory.

Electrical drives with linear or angular position control require the use of specialized digital systems. Such systems often implement complex control algorithms [3]. The aim of this paper is to present a simple example of a drive system loaded with elastic and damping elements. The drive control system works by solving the inverse dynamics problem [4], which requires relatively little computational effort.

The determination of input values (such as control voltages and electromagnetic moments) from the given kinematic elements of motion or from the given properties of motion is one of the main problems associated with the dynamics of electromechanical systems [13]. The inverse dynamics problem has attracted the attention of engineers due to its wide scope of potential applications and general solvability [14]. Computer-aided modeling of various systems by applying the of inverse problems of dynamics has led to a considerable broadening of the concept of the inverse problem itself [11]. If the given properties of motion of the electromechanical system can be represented analytically as first integrals of the appropriate equations of motion, then, in general, the solution of the inverse problem of dynamics is reduced to the construction of an adequate system of differential equations. The coefficients of these differential equations can be computed on the basis of technological requirements, limitations imposed on state variables and other properties known. Solving these differential equations yields information regarding the forces and moments which act upon the system [15].

2. Structure of the control system

It is assumed that the general structure of the control system comprises the following elements:

- a controller which implements the control task
- an object, i.e. a DC motor drive with a permanent magnet in a closed-loop control system with speed and current controllers, containing a transistor-based amplifier,
- mechanical load, involving a flexible damping element and the mass of the mechanism itself,
- a correction block which limits the impact of changes in system conditions and disturbances.

It is assumed that the structure of the motor speed control system is a cascade consisting of two PI regulators whose settings have been optimized. The speed controller acts as the outer loop controller which controls velocity, while the current controller acts as the inner loop controller which treats the output of the outer loop controller as its setpoint and controls the current of the motor.

The structure of this electromechanical drive system is depicted in Fig. 1 with two additional labels: u – corrected control signal; z – disturbance signal.

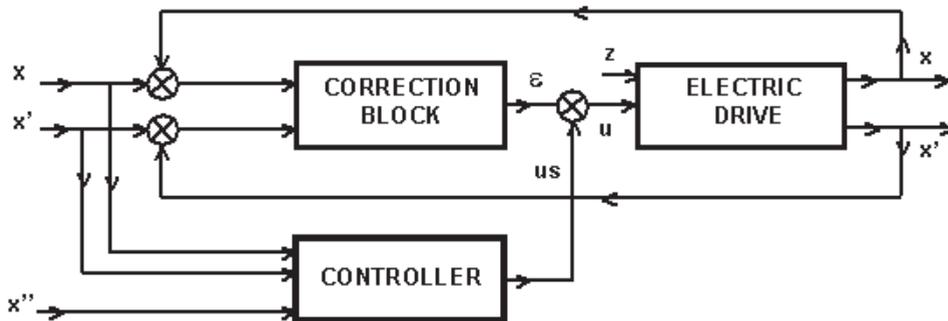


Fig. 1. The control system structure of a DC drive control of an elastic element

3. Electromechanical model and its motion control

The purpose of control is to carry the winding engine mass from the initial state $(x_0, x_0^{(1)}, x_0^{(2)})$ to the final state $(x_k, x_k^{(1)}, x_k^{(2)})$ in minimum time t_k while satisfying limitations imposed on selected state variables such as $x^{(1)}, x^{(2)}$.

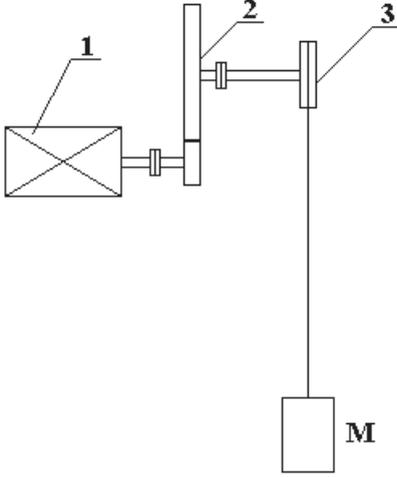


Fig. 2. The general scheme of the electromechanical system, where: 1 – motor, 2 – gear, 3 – drive wheel, M – winding engine mass on elastic rope

The paper presents the results of simulating this process in an electromechanical system. The corresponding block diagram is shown in Fig. 2.

The problem can be reduced to computing the optimal waveform of the engine control variable, which limits transient oscillations of the winding engine mass. The desired trajectory of the winding engine mass can be calculated on the basis of limits imposed on state variables describing the dynamics of the motor and the winding engine mass while minimizing the duration of the process.

The dynamics of the presented electromechanical system is described by the following relationship:

$$x(t)^{(1)} = f[x(t), u(t), t] \quad (1)$$

where:

- $x(t) = \{x_1(t), \dots, x_n(t)\}^T$ – vector of state variables,
- $u(t) = \{u_1(t), \dots, u_m(t)\}^T$ – vector of control variables,
- m, n – dimensions of each vectors,
- T – transpose of a vector,
- $f[x(t), u(t), t]$ – predetermined vector function (generally nonlinear),
- $x(t)^{(1)}$ – first derivative of the state vector.

In this paper, the following assumptions were made:

- the desired trajectory of the winding engine mass in the state space was selected in such a way as to satisfy the following technological limitations:

$$x_z(t) = \xi_z(t); \quad t_0 < t < t_0 + t_k; \quad 0 < t_k < \infty \quad (2)$$

where:

- t_0 – initial time,
- t_k – duration of the transient state;

- the right side of the equation (1) can be expressed as the following sum:

$$x(t)^{(1)} = g[x(t), t] + h[u(t), t] \quad (3)$$

where the input vector function may be represented as:

$$h[u(t), t] = \xi(t)^{(1)} - g[\xi(t), t] \quad (4)$$

4. Specification of the control task

The control task can be expressed as follows: we need to find a vector control function $u(t)$ for which the trajectory of the winding engine mass corresponds with the desired trajectory $\xi(t)$ with the assumed accuracy [5].

In practice, the j -th component of the vector function can typically be expressed as:

$$h_j[u(t), t] = \sum_{i=1}^m \alpha_{ji}(t) u_i(t) \quad \text{for } j = 1, \dots, n \quad (5)$$

where the $\alpha_{ji}(t)$ coefficients follow from time-dependent parameters of the system and are not difficult to calculate.

An approximation of the difference $\xi(t)^{(1)} - g[\xi(t), t]$ can be computed using linear combinations $\sum_{i=1}^m \alpha_{ji}(t) u_i(t)$ for $j = 1, \dots, n$.

By specifying for each state variable ($j = 1, \dots, n$), the error of the approximation as:

$$\varepsilon_j(t) = g_j[\xi(t), t] - \xi_j^{(1)}(t) + \sum_{i=1}^m \alpha_{ji}(t) u_i(t) \quad (6)$$

and minimizing the functional:

$$I = \int_{t_0}^{t_0+t_k} \varepsilon(t)^T \varepsilon(t) dt \quad (7)$$

in each step of the procedure we can determine the coefficients $\alpha_{ji}(t)$ and the control functions $u_i(t)$.

Taking into account the technological limitations of the components of the state vector (e.g. regulator output signals, armature current, speed and acceleration of the winding engine mass), additional conditions may be obtained:

$$x_j(t)_{\min} < x_j(t) < x_j(t)_{\max} \quad j = 1, \dots, n \quad (8)$$

where:

$x_j(t)_{\min}, x_j(t)_{\max}$ – the lowest and highest admissible values of the j -th component of the state vector.

The limitations imposed on vector control signal components (e.g. speed setpoint signal voltage) can be expressed as follows:

$$u_j(t)_{\min} < u_j(t) < u_j(t)_{\max} \quad j = 1, \dots, m \quad (9)$$

where:

$u_j(t)_{\min}, u_j(t)_{\max}$ – the lowest and highest admissible values of the j -th component of the vector control signal.

Taking into account the above mentioned conditions, the components of the input vector function $h[u(t), t]$ may be defined as:

$$\begin{aligned} h_j [u(t), t] &= \sum_{i=1}^m \alpha_{ji}(t) u_i(t) = \xi_j(t)^{(1)} - g_j [\xi(t), t] = \\ &= h_j^*[t, t_0, t_k, u_{\max}, u_{\min}, x_{\min}, x_{\max}, \alpha(t)] \end{aligned} \quad (10)$$

where:

h^* – in general, a nonlinear vector function of parameters (e.g. limiting values of state variables such as start and end time), which defines a desired trajectory of the winding engine mass,

$\alpha(t)$ – a time-dependent parameter matrix.

The control task leads to the solution of equation (10) with respect to vector control components.

5. Simulation studies

Simulation studies were carried out for a DC motor drive with permanent magnets, manufactured by Wamel, series 5680, type DPM 56-DF4 K-7707, designed to feed the drives of numerically controlled machines [5]. Its basic parameters and the limiting values of the control system are as follows: armature resistance 1.75 Ω ; armature inductance 0.037 H, motor torque constant, 0.85 Nm/A; rated torque with the rotor stopped 7.4 Nm, peak torque with the rotor stopped 62.0 Nm, rated speed 1200 rpm, rated current 9.0 A; peak current 71 A; maximum supply voltage 107 V; reduced moment of inertia of the rotating masses 0.015 kgm²; velocity measurement constant $k_{ig} = 0.301$ Vs; current feedback gain 0.33 V/A; maximum control voltage $u_{\max} = 8.5$ V; gain rectifier built on power transistors 50 V/V. The signal delay introduced by the rectifier was seen as negligible.

The study used a cascade armature current control system with speed and current PI controllers. Settings have been optimized on the basis of the polynomial Ellert criterion, limiting the output signals of current and speed controllers to between -10.0 V and $+10$ V; gain of the speed controller: 4.084 V/V; time constant of the speed controller: 0.08 s; gain of the current controller: 0.698 V/V; time constant of the current controller: 0.021 s.

The parameters of the mechanical part of the system and limitations placed upon the state variables associated with it were as follows: mass of the winding engine $M = 20$ kg; elasticity modulus of the rope $k = 100$ N/m; substitute damping factor of the rope $c = 20$ Ns/m, radius of the drive wheel $r = 0.2$ m; speed limit of the winding engine mass $x_{\max}^{(1)} = 5.648$ m/s; mass acceleration limits: -1.25 m/s² $\leq x_{\max}^{(2)} \leq 1.25$ m/s²; derivative of mass acceleration limits: -5.0 m/s³ $\leq x_{\max}^{(3)} \leq 5.0$ m/s³; second derivative of mass acceleration limits: -25.0 m/s⁴ $\leq x_{\max}^{(4)} \leq 25.0$ m/s⁴.

The aim of the process control was to position the winding machine mass. This was achieved by appropriately shaping the engine speed waveform. The whole dynamic process lasted 20 s and consisted of three stages: startup (0–5 s); movement with maximum established speed (5–15 s); and braking (15–20 s).

To determine the trajectory of the winding engine mass $x(t)$, it is necessary to analyze the technological limitations and find the relationships between them. The maximum values of the speed and acceleration of the mass correspond to linear velocity and linear acceleration of a point on the circumference of the drive wheel. Peak values of the linear velocity $x_{m \max}^{(1)}$ and linear acceleration of a point on the circumference of the drive wheel $x_{m \max}^{(2)}$ can be expressed as follows:

$$x_{m \max}^{(1)} = \frac{u_{\max} r}{k_{ig}}; \quad x_{m \max}^{(2)} = \frac{i_{\max} k_m r}{J + M r^2} \quad (11)$$

where:

- r – radius of the drive wheel,
- k_m – motor torque constant,
- k_{ig} – velocity-feedback gain factor,
- u_{\max} – control voltage limit,
- i_{\max} – larmature current limit,
- $x_{m \max}^{(1)}$ – peak value of the linear position derivative of a point on the circumference of the drive wheel,
- J – moment of inertia of the rotating masses (reduced to the motor shaft).

The armature current limit in a transient state is indirectly related to the limit of the second derivative of the linear position of a point on the circumference of the drive wheel. The differential equation of the drive transmission system can be expressed as:

$$M x^{(2)}(t) + c x^{(1)}(t) + k x(t) = c x_m^{(1)}(t) + x_m(t) \quad (12)$$

where:

- $x^{(1)}(t), x^{(2)}(t)$ – first and second derivatives (respectively) of coordinates specifying the position of the winding engine mass.

The architecture of the electromechanical system described above is depicted in Fig. 3.

The top branch of the electromechanical control system comprises a trajectory controller which is used to solve the inverse problem of dynamics, as well as a DC drive system with PI

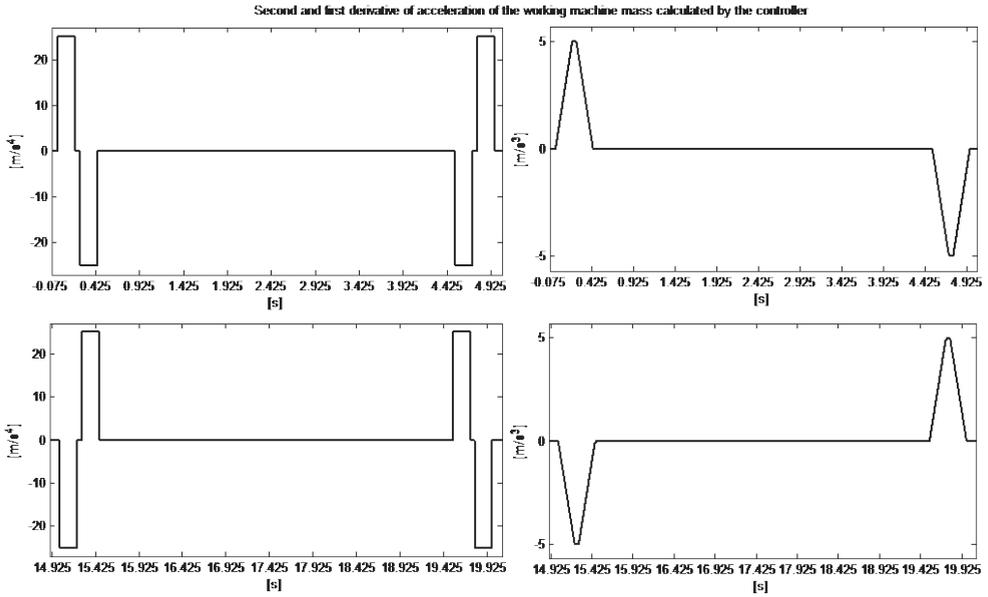


Fig. 4. The second and first derivative of the winding engine mass in the engine startup and braking stages, as calculated by the controller

Fig. 4 shows the first and second derivatives of the winding engine mass in the engine startup and braking stages. Calculations were performed using the trajectory controller, which implements the principle of inverse dynamics. In this way, we can obtain the desired waveforms of variables describing the motion of the winding engine, which define the components of the right side of differential equation (10) and allow us to proceed with its numerical integration. The waveform of the winding engine mass acceleration in the real system may deviate from the typical trapezoidal shape (dotted line) – the controller should therefore provide for a limitation of its second derivative (solid line), as can be seen in Fig. 5.

In order to calculate acceleration waveforms for the electromechanical transmission system, the controller implements the algorithm resulting from equation (12). The starting point for the calculation is provided by the computed acceleration of the winding engine mass and the initial condition imposed on the position of a point on the circumference of the drive wheel. The procedure consists of iterative numerical integration of equation (13), each time placing the integral calculated by the previous iteration on the right-hand side of the formula

$$x_m^{(1)}(t) = \frac{M}{c} x^{(2)}(t) + x^{(1)}(t) + \frac{k}{c} x(t) - \frac{k}{c} x_m(t) \quad (13)$$

Fig. 5 shows the waveform of the control voltage signal, which was calculated using the inverse problem of dynamics. On the basis of the mathematical model of the electromechanical transmission system, the controller calculates the linear velocity waveform of a point on the circumference of the drive wheel. Subsequently, by applying the mathematical model

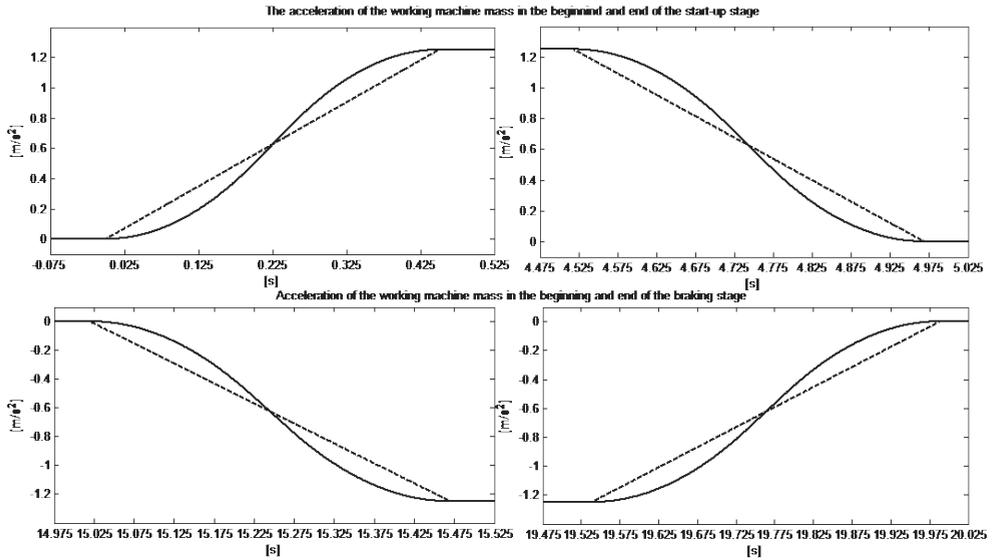


Fig. 5. Programmed waveform of the acceleration of the winding engine mass

of the DC motor control system, it calculates the control voltage signal. As can be shown, the controller modifies the motor speed waveform in stages corresponding to the assumed maximum allowed value of the second derivative of the winding engine acceleration coefficient:

- in the startup stage: 0.0–0.45 s and 4.55–5.0 s,
- in the braking stage: 15.0–15.45 s and 19.55–20.00 s.

Fig. 6 presents a comparison between the linear velocity waveforms of the winding engine mass and a point on the circumference of the drive wheel. The aim of the control process was to position the winding engine mass. This was achieved by appropriately shaping the engine speed waveform.

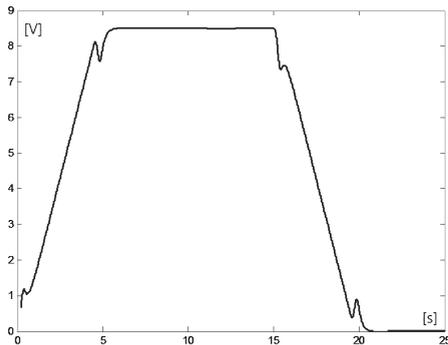


Fig. 6. Control voltage waveform calculated using the proposed algorithm

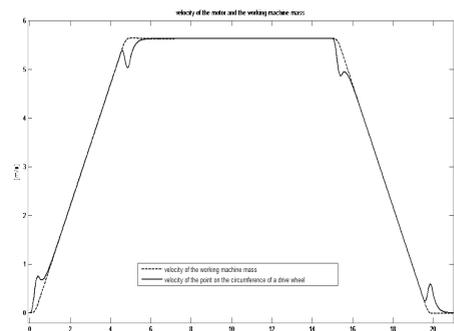


Fig. 7. Linear velocities of the winding engine mass and a point on the circumference of the drive wheel

Fig. 8 and Fig. 9 show the waveforms extending in the initial part of the startup stage (0.2–1.2 s) and in the final part of the startup stage (4.5–6.0 s). Fig. 10 and Fig. 11 show the waveforms extending in the early braking stage (14.8–16.2 s) and the late braking stage (15.9–20.7 s).

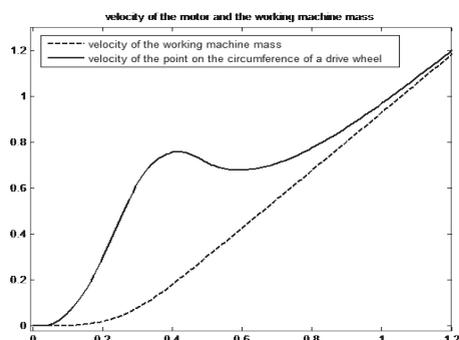


Fig. 8. Linear velocities of the winding engine mass and a point on the circumference of the drive wheel during the early startup stage

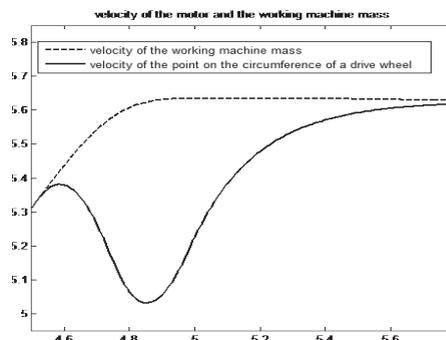


Fig. 9. Linear velocities of the winding engine mass and a point on the circumference of the drive wheel during the late startup stage

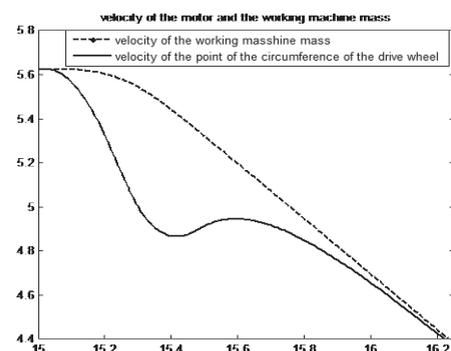


Fig. 10. Linear velocities in the early braking stage

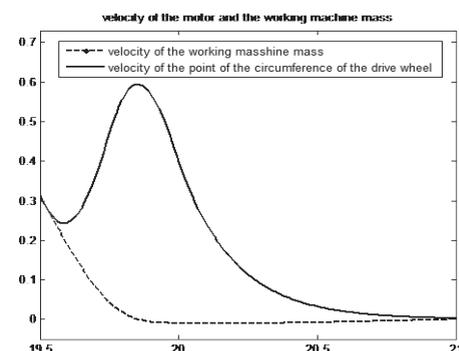


Fig. 11. Linear velocities in the late braking stage

Fig. 12 and Fig. 13 present a comparison between the linear acceleration of the winding engine mass and the linear acceleration of a point on the circumference of the drive wheel during the final stages of startup and braking. It can be shown that the values of the second derivative of the linear acceleration of the winding engine mass satisfy the limit conditions $-25.0 \text{ m/s}^4 \leq x_{M/\text{max}}^{(4)} \leq 25.0 \text{ m/s}^4$.

Fig. 14–Fig. 17 show a comparison between the waveforms of the electromagnetic moment of the motor and the dynamic load torque on the motor shaft in the working cycle.

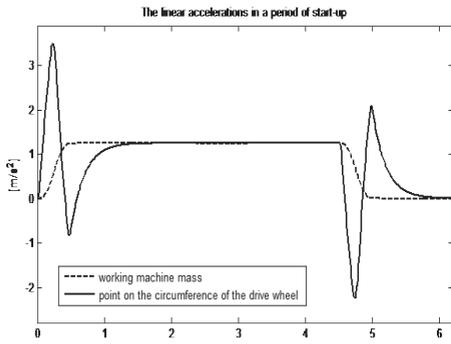


Fig. 12. Linear acceleration of the winding engine mass and a point on the circumference of the drive wheel during startup

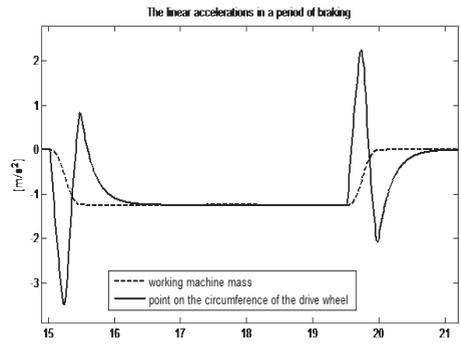


Fig. 13. Linear acceleration of the winding engine mass and a point on the circumference of the drive wheel during braking

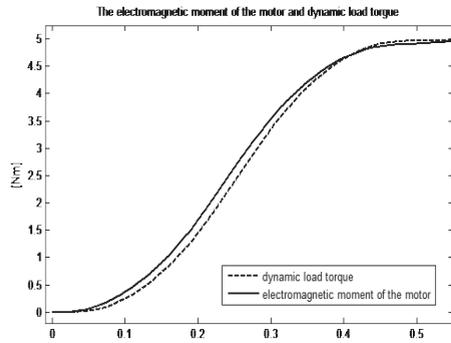


Fig. 14. Electromagnetic moment of the motor and dynamic load torque in the early startup stage

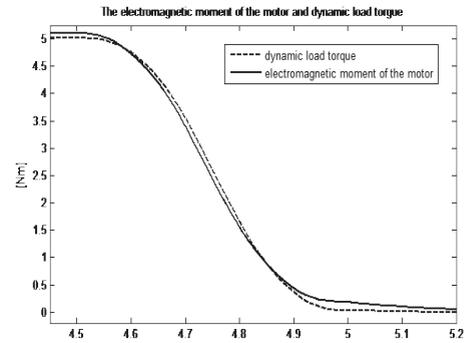


Fig. 15. Electromagnetic moment of the motor and dynamic load torque in the late startup stage

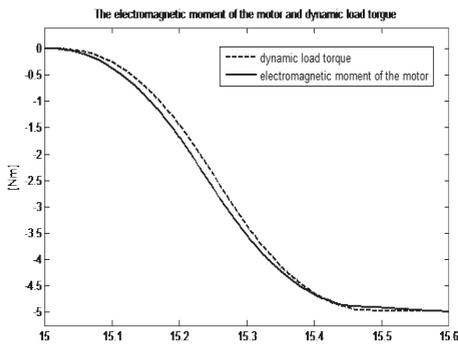


Fig. 16. Electromagnetic moment of the motor and dynamic load torque in the early braking stage

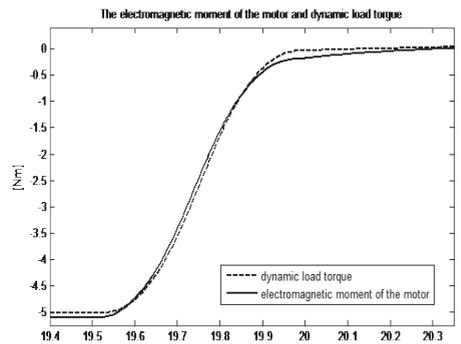


Fig. 17. Electromagnetic moment of the motor and dynamic load torque in the late braking stage

6. Conclusions

One of the major problems associated with the dynamics of electromechanical systems is deriving control variables from equations which describe the behavior of a physical system in terms of its motion as a function of time. The properties of motion of the electromechanical system can be computed in many different ways, one of which is by applying the inverse dynamics method [10, 11]. In this paper, the construction of the programmed motion of a winding engine is reduced to equations of motion under which the motion of the winding engine mass is characterized by minimum oscillations.

In order to solve the inverse problem of dynamics, the necessary and sufficient conditions for ensuring motion with specific properties were formulated. The motion properties of an electromechanical system whose state is defined by a vector of generalized coordinates and a vector of generalized velocities were given as a system of inequalities in compliance with technological requirements. The following interdependences were found in the scope of technological limitations. In a transient state, the motor velocity and winding engine mass limits depend on control voltage limits, while the motor and winding engine acceleration limits depend on armature current limits. It is assumed that the transient state should be as short as possible.

The trajectory of the programmed motion was calculated by assuming that the variables should reach their limits in the shortest time. From the equations of motion constructed on the basis of inverse dynamics, controlling variables and corresponding parameters were determined.

The inverse dynamics method is suitable for direct control over the motion of the winding engine mass in order to avoid negative dynamic effects. Nevertheless, disturbances in a real electromechanical system may result in the loss of control quality. If load torque disturbances are directly transmitted to the motor shaft, the presented engine control system compensates them well. When disturbances interact with the winding engine mass instead, and momentary values describing the dynamic state of this mass can be measured in real time, the correction block shown in Fig. 1 should be applied. Solving such problems is important in the context of hoisting machine control systems in the mining industry [6].

Similar results can be obtained by formulating the issue as a problem of finding a control law for a given system assuming a certain optimality criterion with a square indicator of quality. The task of the control system is then to find a control variable waveform $u_s(t)$ which will minimize energy losses during transients (e. g. startup or braking) at time t_s . The work performed by the electromagnetic torque of the motor during startup is equal to the sum of the final kinetic energy of the motor and the actuator, and the action of dissipative forces.

The concept of minimizing energy losses during startup may be replaced by minimizing losses associated with the work of dissipative forces. This leads to minimization of the following functional:

$$W_1 = \int_0^{t_s} c \left(x_m^{(1)} - x^{(1)} \right)^2 dt \quad (14)$$

where:

- c – is damping factor,
- T – is the startup duration,
- $x_m^{(1)}$ – is the linear velocity of a point on the circumference of the drive wheel.

References

- [1] Dwivedy S.K., Eberhard P., *Dynamic analysis of flexible manipulators, a literature review*, Mechanism and Machine Theory, Vol. 41, Issue 7, July, 2006, 749-777.
- [2] Ider S.K., *Inverse dynamics of parallel manipulators in the presence of drive singularities*, Mechanism and Machine Theory, Vol. 40, Issue 2, January, 2005, 33-44.
- [3] Jarzębowska E., *Model-Based Tracking Control of Nonlinear Systems*, CRC Series: Modern Mechanics & Mathematics, CRC Press, Taylor & Francis Group, 2012.
- [4] Galiullin A.S., *Inverse Problems of Dynamics*, Mir Publishers, Moscow 1984.
- [5] Zajac M., *Wybrane zagadnienia syntezy sterowania programowego układów elektromechanicznych*, Rozwój zautomatyzowanych układów napędu elektrycznego, Komitet Elektrotechniki PAN, PWN, 1992, 129-139.
- [6] Zajac M., *New Method of Mine Winder Control in Deep Shafts*, Prace Instytutu Elektrotechniki, Vol. 151, 1988, 81-93.
- [7] Calanca A., Capisani L.A., Ferrara A. et al., *An Inverse Dynamics-Based Discrete-Time Sliding Mode Controller for Robot Manipulators*, Robot Motion and Control, 2007, 137-146.
- [8] Barai R.K., Dey A., *Sliding mode compensation of inverse dynamics velocity control*, 12th Int. Workshop on Variable Structure Systems (VSS), 2012, 155-160.
- [9] Morales B., Carelli R., *Robot control with inverse dynamics and non-linear gains*, Latin American Applied Research, Vol. 33, No. 4, Oct./Dec. 2003, 393-397.
- [10] Kuo A.D., *A Least-squares Estimation Approach to Improving the Precision of Inverse Dynamics Computations*, Transactions of the ASME, Journal of Biomechanical Engineering, Vol. 120, Feb. 1999, 148-159.
- [11] Kato O., Sugiura I., *An interpretation of airplane general motion and control as inverse problem*, Journal of Guidance, Control, and Dynamics, Vol. 9, No. 2, 1986, 198-204.
- [12] Venture G., Kojima T., Tagawa Y., *Fast Motion Control of Robotics Systems Using Inverse Dynamics via 'Simulation of Feedback Control Systems' (IDCS)*, The 1st Joint International Conference on Multibody System Dynamics, May 25–27, 2010, Lappeenranta, Finland.
- [13] Wach P., *Dynamics and control of Electrical Drives*, Springer-Verlag, 2011.
- [14] Крутько П.Д., Лакота Н.А., Попов Е.П., *Конструирование алгоритмов управления приводных систем по методу обратных задач динамики*, Доклады Академии Наук СССР, Кибернетика и теория регулирования, 1982, 1086-1089.
- [15] Крутько П.Д., *Обратные задачи динамики управляемых систем. Линейные модели*, Наука 1987.

VOLODYMYR MAKSYMOVYCH*, OLEH HARASYMCHUK**, YURIY KOSTIV*

POISSON PULSE SEQUENCE GENERATORS BASED UPON MODIFIED GEFFE GENERATORS

GENERATORY CIĄGÓW IMPULSÓW POISSONA OPARTE NA ZMODYFIKOWANYCH GENERATORACH GEFFEGO

Abstract

The article presents principles of optimizing the parameters of structural elements Geffe generator. The quality of this optimization is confirmed by statistical tests package NIST STS. The article provides methodology for research into the settings of the output signals of the Poisson pulse sequence generators using Pearson's chi-squared test.

Keywords: Geffe generator, pseudo-random sequence, tests NIST STS

Streszczenie

W artykule zawarto zasady optymalizacji parametrów elementów strukturalnych generatora Geffego. Jakość optymalizacji została potwierdzona za pomocą pakietu testów statystycznych NIST STS. Przedstawiono metodykę badania parametrów wyjściowych sygnałów generatorów kolejności impulsów Poissona z zastosowaniem zmodyfikowanego kryterium Pearsona.

Słowa kluczowe: generator Geffego, ciąg pseudolosowy, testy NIST STS

* Prof. D.Sc. Ph.D. Volodymyr Maksymovych, M.Sc. Yuriy Kostiv, Chair of Information Technology Security, Lviv Polytechnic National University.

** Ph.D. Oleh Harasymchuk, e-mail: garasymchuk@ukr.net, Chair of Data Protection, Lviv Polytechnic National University.

1. Introduction

As information technology undergoes rapid development, random and pseudorandom sequence generators (PRSGs) substantially widen the scope of their application.

As it stands at present, multiple methodologies and principles exist supporting the generation of pseudorandom sequence, each of the said methodologies and principles having their respective advantages and disadvantages [1–5]. Among the aforementioned pseudorandom sequence generators, the merits of the Geffe generator stand out [6–10] – however, its qualitative characteristics have so far been insufficiently researched. Thus, a necessity arises to improve the characteristics of the Geffe generator in order to ensure that the output sequences that it generates could be used directly and/or indirectly, to face and approach challenges that arise in the sphere of data protection.

In order to come to a certain conclusion as of whether it is or is not possible to use a certain pseudorandom sequence generator for the purpose of solving specific tasks, one should assess its quality and operation reliability. Conducting tests on generators, especially those being used in data protection systems (cryptographic applications in particular) is a pressing and practically important task. As of today, a number of graphic and evaluating tests are used to evaluate pseudorandom sequences. Additionally, several software products have been developed which contain packages of tests and these packages are used to verify a variety of statistical properties pertaining to pseudorandom generators, the most well-known of these being the package of NIST STS statistical tests [11, 12].

The purpose of this paper is to use the NIST STS package of statistical tests to ascertain which settings are optimal for the structural elements of a Geffe generator, by way of modification of the structural principles of its basic generators. It is the intention to develop new and means methods of building Poisson pulse generators (PPG), to conduct a comparative analysis providing insight into their characteristics, and to develop methodologies allowing their quality to be assessed.

2. Assessment of Geffe generator's statistical characteristics

A Geffe generator enables the mixing of two sequences (x_1 and x_2) from the outputs of two M-sequence generators the latter also being known as generators based upon linear feedback shift registers (LFSR) by way of controlling the LFSR output sequence 3. Such mixing takes place pursuant to the following function:

$$F(x_1, x_2, x_3) = x_1 \bar{x}_3 + x_2 x_3 = x_3 \oplus x_1 x_2 \oplus x_2 x_3 \quad (1)$$

which can be performed using the multiplexer $2 \rightarrow 1$ – see Fig. 1 [1].

M-sequence generators being fundamental to a Geffe generator can be created in a variety of ways, pursuant to the following equation:

$$Q(t+1) = T^r Q(t) \quad (2)$$

where $Q(t)$ and $Q(t + 1)$ – represent the generator’s register state in the t i $t + 1$ instants of time respectively (i.e. prior to and following the arrival of a synchronising pulse); T stands for square matrix of order N , whereas r stands for a degree of a primitive polynomial.

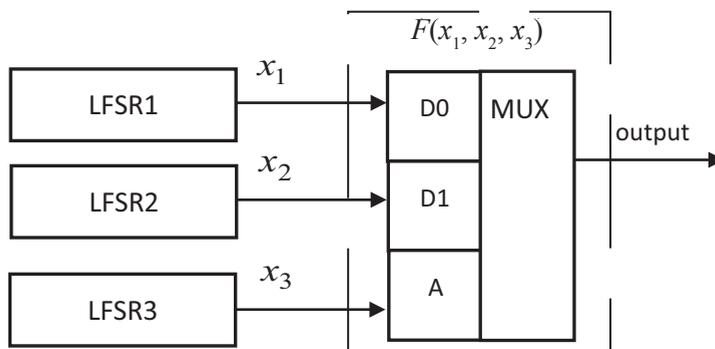


Fig. 1. A Geffe generator

We have taken a decision to use statistical tests and a change in the r degree (and thus a change in the structure of the generator itself) to ascertain whether such actions will or will not influence the quality of the Geffe generator’s output pseudorandom sequence.

In order to assess the quality, we have not only opted for basic M-sequence generators with varying degrees of generative polynomial but also modified the degree of r to which the T matrix is elevated.

Assessment of output sequences from the generator was conducted using a package of NIST STS statistical tests. As of the present moment, no similar assessments of the Geffe generator can be found in research literature. In order to receive sequences from such a generator, we have turned to Delphi language choosing it to be our medium to develop simulation models of the said generator which allowed us to produce output sequences depending upon how the settings are changed.

A package of NIST STS tests includes 15 statistical tests developed to verify the hypothesis of randomness of binary sequences of arbitrary length are generated by PRSGs [11].

If the result of the P test falls within the 0.98–1.00 range, such a test result is pronounced successful. If the probability of P is below 0.98, the test is considered unsuccessful. Using the results procured in such a manner, we proceed to develop a statistical portrait of generators which is comprised of a matrix with a size of $m \times q$, where m stands for the number of binary sequences which are being verified and q stands for the number of statistical tests which are being performed to test each sequence. The ultimate resolution as to whether the sequence has or has not turned out to be random is taken following the obtaining of a cumulative result for all tests [12].

The testing was conducted at the significance level $\alpha = 0.01$, as recommended by the developers of NIST STS. Statistical portraits which can be seen upon observing Fig. 2 and Fig. 3 have the appearance of a size 1000×188 matrix which contains 188 000 values of respective probabilities. All of the figures display the confidence range in red lines.

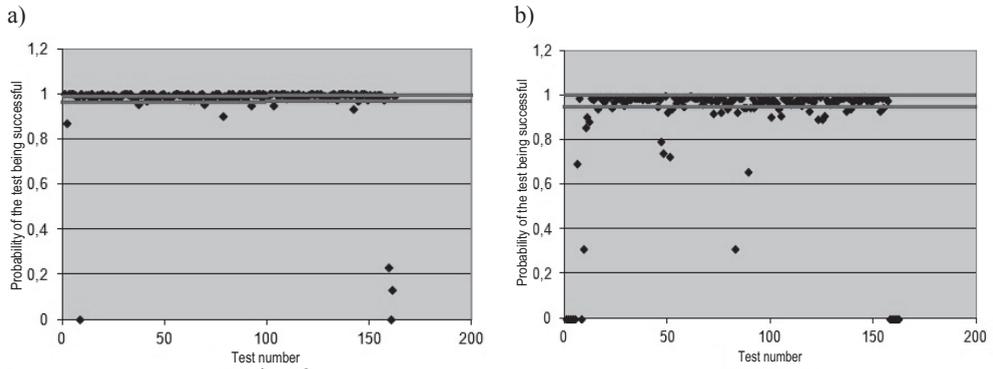


Fig. 2. A statistical portrait of Geffe generator No. 1: a) $r = 1$, b) $r = 5$

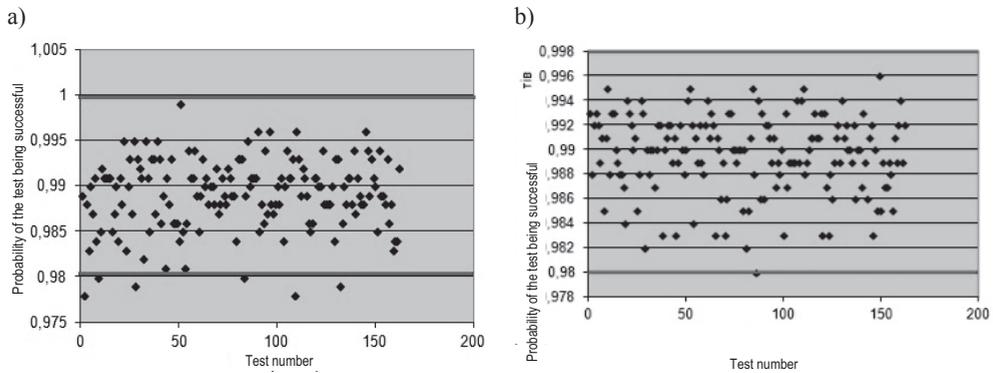


Fig. 3. A statistical portrait of Geffe generator No. 4: a) $r = 1$, b) $r = 5$

We have researched a large number of generators, but the optimisation of working settings is best shown using several of the following combinations:

1. *Geffe generator No. 1*: LFSR 1 and LFSR 2 based upon polynomial $\Phi(x) = 1 \oplus x^{12} + x^{17}$; LFSR 3 – $\Phi(x) = 1 \oplus x^6 + x^7$;
2. *Geffe generator No. 2*: LFSR 1 based upon polynomial $\Phi(x) = 1 \oplus x^{12} + x^{17}$ and LFSR 2 based upon polynomial $\Phi(x) = 1 \oplus x^{18} + x^{25}$; LFSR 3 – $\Phi(x) = 1 \oplus x^6 + x^7$;
3. *Geffe generator No. 3*: LFSR 1 and LFSR 2 based upon polynomial $\Phi(x) = 1 \oplus x^{18} + x^{25}$; LFSR 3 – $\Phi(x) = 1 \oplus x^6 + x^7$;
4. *Geffe generator No. 4*: LFSR 1 and LFSR 2 based upon polynomial $\Phi(x) = 1 \oplus x^{18} + x^{31}$; LFSR 3 – $\Phi(x) = 1 \oplus x^6 + x^7$.

A detailed report on the assessment of Geffe generators for each test is provided in Table 1.

Results of the research provided in Fig. 2 and Fig. 3, as well as in Table 1 testify to the fact that an increase of the degree r of basic M-sequence generators, the quality of a Geffe generator improves, since a number of failed tests decreases.

Results of the testing of Geffe generators

No.	Statistical Test	Generators number							
		1		2		3		4	
		$r=1$	$r=5$	$r=1$	$r=5$	$r=1$	$r=5$	$r=1$	$r=5$
1	Frequency (Monobit) Test	-	-	-	+	-	+	+	+
2	Frequency Test within a Block	-	-	-	+	-	+	-	+
3	Cumulative Sums (Cusum) Test	-	-	-	-	-	+	+	+
4	Runs Test	-	-	+	+	+	+	+	+
5	Test for the Longest Run of Ones in a Block	-	-	+	+	+	+	+	+
6	Binary Matrix Rank Test	+	+	+	+	+	+	+	+
7	Discrete Fourier Transform (Spectral) Test	-	-	-	-	-	-	+	+
8	Non-Overlapping Template Matching Test	-	-	+	+	+	+	-	+
9	Overlapping Template Matching Test	-	-	+	+	+	+	+	+
10	Maurer's 'Universal Statistical' Test	+	-	+	+	+	+	+	+
11	Approximate Entropy Test	-	-	+	+	+	+	+	+
12	Serial Test	-	-	+	+	+	+	+	+
13	Linear Complexity Test	+	-	+	+	+	+	+	+
14	Random Excursions Test	-	+	+	+	+	+	+	+
15	Random Excursions Variant Test	-	+	+	+	+	+	+	+

3. Poisson Pulse Sequence Generators created on the basis of modified Geffe generators

Poisson pulse sequence generators (PPSGs) can be created on the basis of pseudorandom sequence generators PRSGs [13] using both software and hardware. The primary advantage of the latter is its high performance. Another paper [13] offered PPCG structures based upon linear congruential pseudorandom sequence generators. Partial research into statistical characteristics of their output signals has shown that they mainly satisfy the requirements stipulated for devices meant to be used in computing technology. However, the fact that the algorithm of their work includes multiplication and division operators and thus also multiplication and division circuits, this causes substantial performance losses, thereby reducing the principal advantage that hardware has. Therefore, a problem arises of how to find efficient and fast methods and means to perform PPCG and find methods to assess their quality.

The main requirements stipulated for the attention of developers in the process of building a PPCG are stated below:

- a PPCG must have satisfactory statistical characteristics;
- a PPCG must have a long repetition period;

- high performance (for both hardware and software);
- a wide frequency range for output signal;
- ability to control the average frequency of the output signal;
- maximum possible simplicity of the build (for both hardware and software).

One of the most successful structures to form a pulse sequence with Poisson law of distribution which allows for the prompt control of the average frequency of PPSG output pulses is presented on Fig. 4. It is comprised of a PRSG, a comparing element (CE) and a logical element AND. PRSG, in turn, is comprised of generators based on generators built upon shift registers with linear feedback and a multiplexer $2 \rightarrow 1$.

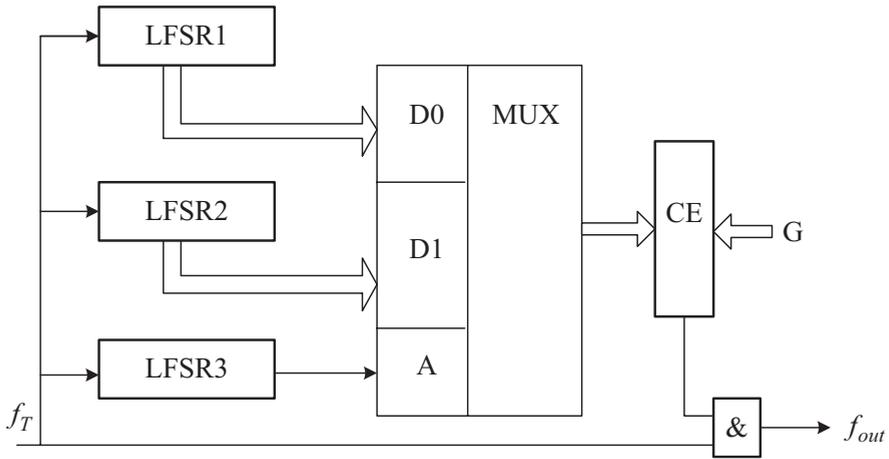


Fig. 4. PPSG based upon modified Geffe generator

Timed pulses arrive at the output of PPSG if the X number at the output of PPSG is smaller than the Control Code G. The average frequency of generator's input pulses is determined via the following equation:

$$f_{out} = \frac{G}{X_{max}} f_T \quad (3)$$

where:

- X_{max} – stands for the maximum value of X,
- f_T – stands for frequency of repetition of timed pulses,
- G – stands for control code.

One of the possible options for building a PPSG is to build it on the basis of a Geffe generator which has the multiplication operator removed from its algorithm. Thus, a prospective direction in which the PPSG's characteristics may be improved is if it has such generators used in its structure, and if modifications can be performed if necessary, and if the settings of output pulse sequence can be controlled.

Since we have now researched the optimal settings of Geffe generators, we have attempted to create a model of PPSG on the basis of Geffe generators and assess the quality of output sequence.

4. Methodology for assessment of PPSG on the basis of Geffe generators

At the present time, there exist a great number of tests (graphic and evaluative) as well as packages of tests for the assessment of the quality of PPSG and PSG. We have applied one such test to assess the quality of the Geffe generator.

However, if we want to assess the quality of sequences obtained from generators with distributions diverging from a uniform distribution, we shall find far fewer of such tests. In particular, if one wants to assess the quality of a PPSG, one will also find few applicable tests for that purpose, and those that do exist are mostly graphic, whereby the conclusion of whether a certain sequence is or isn't applicable, may be quite subjective. It is far better to conduct such a research using assessment tests.

We, in order to research statistical characteristics, have offered the following methodology.

A sequence of PPSG input pulses is distributed into n equal groups, each of which contains i_{\max} pulses (see Fig. 5). The maximum allowable number of groups is n_{\max} . Groups of input pulses correspond to groups of output pulses with the number of pulses $k_1, k_2, \dots, k_{n_{\max}}$.

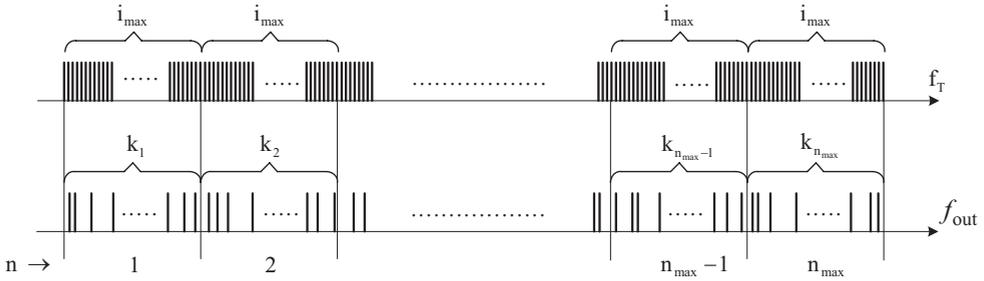


Fig. 5. Breakdown of input and output pulses into groups

The methodology offered by us is based upon classical methodology employed to verify the hypothesis of parent distribution pursuant to the Poisson law using Pearson's chi-squared test (the χ^2 test) [14]. That said, considering the specific characteristics of PPSG's architecture, the following supplements have been offered:

- for the $k_1, k_2, \dots, k_{n_{\max}} - k_c$ numbers, a nominal (theoretical) average value is set, regardless of what the value of the G control code is,
- the i_{\max} is variable, depends upon what the value of G is, and is determined using the following equation:

$$i_{\max} = \frac{x_{\max}}{G} k_c \quad (4)$$

Further verification of the mentioned hypothesis proceeds as follows:

1. Using the empiric distribution obtained as a result of PPSG simulation, we determine the average value of $k_1, k_2, \dots, k_{n_{\max}} - k_B$.
2. For the λ parameter of Poisson's distribution, a random average $\lambda = k_B$ is taken.
3. Using the Poisson's formula:

$$P_j = \lambda^j \frac{e^{-\lambda}}{j!} = k_B^j \frac{e^{-k_B}}{j!} \quad (5)$$

we determine the probability of the appearance of exactly j pulses (within the i_{\max} range) in n_{\max} tests ($j = 0, 1, 2, \dots$).

4. Theoretic frequencies are obtained as follows:

$$Q_j = P_j \cdot n_{\max} \quad (6)$$

5. In the process of simulation, empiric frequencies – N_j – are determined.
6. For each value of j , Pearson's chi-squared test is used to determine the following:

$$S_j = \frac{(N_j - Q_j)^2}{Q_j} \quad (7)$$

$$\chi_c^2 = \sum_{j=0}^{j_{\max}} S_j \quad (8)$$

7. If necessary, the N_j, P_j and Q_j values corresponding to the unlikely probabilities of P_j may be summarised into one or two groups; in this case, the (7) and (8) calculations are performed with that fact having been taken into due consideration.
8. The number of degrees of freedom is determined as follows:

$$r = d - 2 \quad (9)$$

where d stands for the number of groups remaining after possible summarisation.

9. Using the tables of χ^2 critical distribution points [14]; using a selected level of significance α (usually, α is given one of the following three variants of value: 0.1; 0.05; or 0.01) and using the number of degrees of freedom, r , the critical χ_{kp}^2 value is determined. If $\chi_c^2 < \chi_{kp}^2$ – then there is no grounds to reject the hypothesis stating that the pulse sequence is in compliance with Poisson's law of distribution.

The proposed methodology was applied primarily in the process of examination of PPSG parameters implemented in software whereby the Random function of the Delphi programming language (environment) was used.

The Fig. 6 displays the results of a research into ГИИИ on the basis of a modified Geffe generator whereby the following values of parameters of its structural elements were taken: polynomial $F(x) = 1 + 18x + 25x^2$, matrix T1, LFSR 1 – $r = 10$; LFSR 2 – $r = 5$; LFSR 3 – $r = 3$.

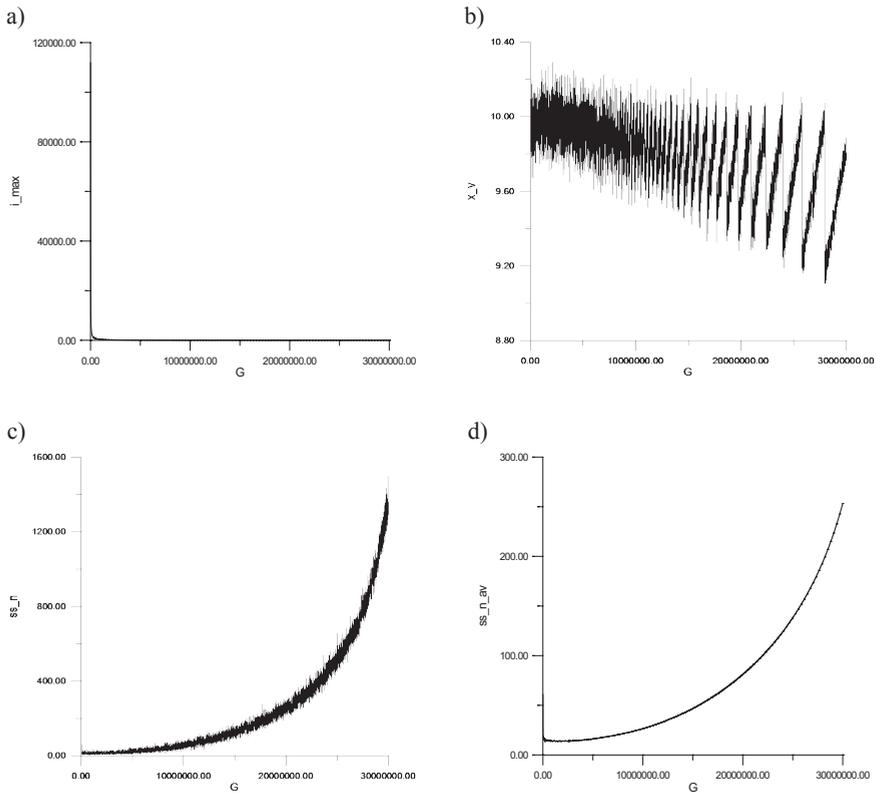


Fig. 6. Research results of PPSG on the basis of a Geffe generator whereby G changes its value within the 0–30000 range

Here one can see how the following values are dependent upon the control code G:

- i_{\max} – number of input pulses in the group i_{\max} (Fig. 6a);
- x_v – statistical (selected) average number of output pulses in the group k_b (Fig. 6b);
- ss_n – value of the Pearson's chi-squared test χ_c^2 (Fig. 6c);
- ss_n_{av} – current average value of Pearson's chi-squared test χ_{ccp}^2 (Fig. 6d).

The results of the application of the proposed methodology for research into statistical characteristics of the input signals of PPSG for some variants of PPSG architecture on the basis of Geffe generators are provided in Fig. 7–8.

The provided results allow us to come to the conclusion that the statistical characteristics of the output signal of PPSG substantially depend upon the architecture of PSG. The PPSG is built in compliance with the offered structure (Fig. 4) which not only allows for the improvement of statistical characteristics but also for extending/increasing the period of repetition.

Statistical characteristics of the output signal of PPSG substantially depend upon the degree of the r matrix which sets the structure of LFSR linear feedbacks.

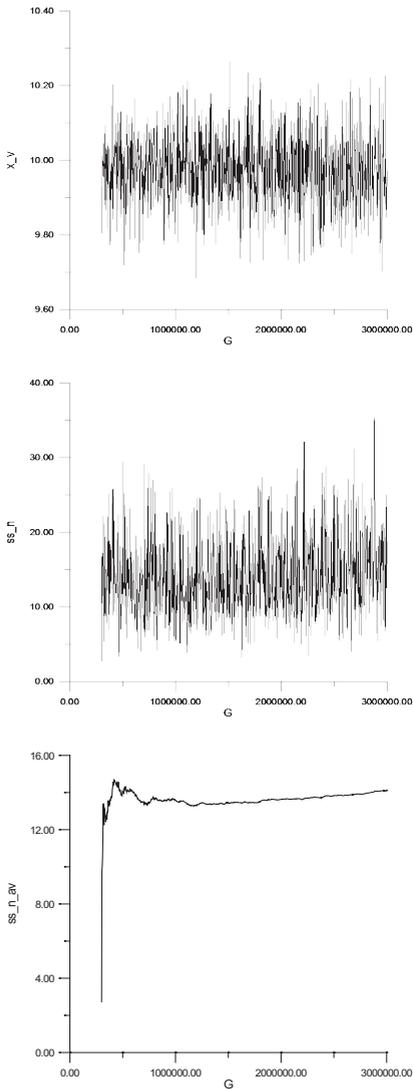


Fig. 7. Research results of PPSG on the basis of a Geffe generator whereby G changes its value within the 300–3000 range

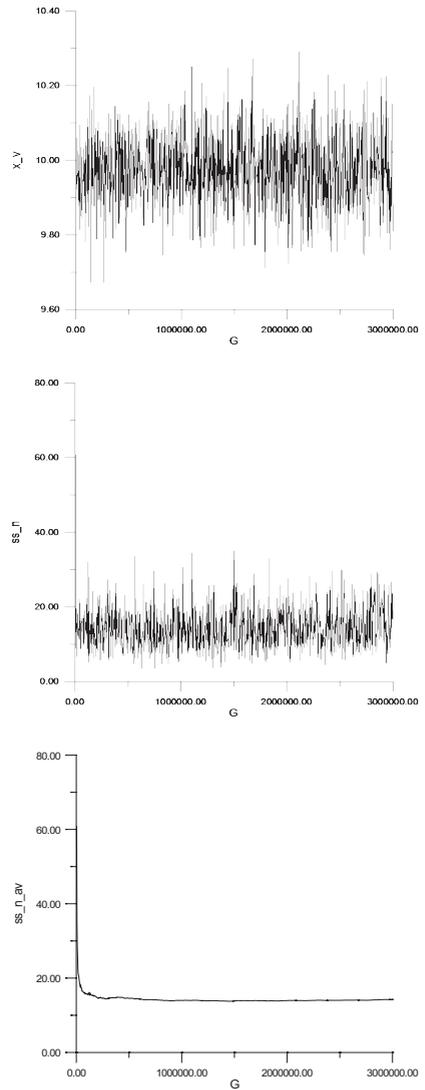


Fig. 8. Research results of PPSG on the basis of a Geffe generator whereby G changes its value within the 0–3000 range

5. Conclusions

By way of using simulation modelling and statistical testing, it has been ascertained that by modifying the structure of basic LFSR generators, specifically by increasing the value of the r degree and by choosing the generative polynomial with a larger degree, one may

substantially improve the quality of Geffe generator's output pulse sequence. Research has shown that $r = 5$ is the optimal value here. Further augmentation of r for large degrees of the generative polynomial does not cause substantial improvement in the quality of the output sequence; however, in this case the creation of such a generator requires higher equipment expenditures.

The paper proposes to build a PPSG on the basis of a modified Geffe generator which, in order to create pseudorandom numbers, resorts to the multiplexing of PSG bits on the basis of LFSR with the help of output of one of the bits of the control LFSR which allowed for expanding the range of values of the control code enough to be able to obtain satisfactory statistical characteristics of the output signal.

The statistical characteristics of PPSG are satisfactory enough for it to be used, in particular, to test radiation dosage metres. The proposed methodology for testing can be efficiently applied to test PPSGs built on the basis of other basic generators.

References

- [1] Ivanov M.A., *Cryptographic methods for data protection in computer systems and networks*, Moscow 2001, 368.
- [2] Ivanov M.A., *Theory, application, and quality assessment of pseudorandom sequence generators*, Moscow 2003, 240.
- [3] Harasymchuk O.I., *Pseudorandom number generators, their application, classification, principal methods of construction and assessment of quality*, Kyiv 2002, 7.
- [4] Harasymchuk O.I., *Poisson Pulse Sequence Generators based on m-sequence generators*, Herald of Lviv Polytechnic National University, Computer Sciences and Information technologies, No. 521, 2004, 17-23.
- [5] Rock A., *Pseudorandom Number Generators for Cryptographic Applications*, Salzburg 2005, 57-65.
- [6] Rosenthal J., *Detection and Exploitation of Small Correlations in Stream Ciphers*, Institute of Mathematics, University of Zurich, 2008.
- [7] Qi D., *Modified Geffe Test Pattern Generator for Built-in Self-test*, IEEE Pacific Rim Conference, 22–24 Aug. 2007, 210-213.
- [8] Oujezský V., *Cryptographic Sequence Generators for Stream Cipher and Their Behavioral Description*, International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, Vol. 4, Issue 3, March 2014.
- [9] Khamees H.Th., *Encryptoin and decryption of data by Using Geffe Algorithm*, International Journal of Modern Engineering Research (IJMER), Vol. 2, Issue 3, May–June 2012, 1354-1359.
- [10] Wei S., *On Generalization of Geffe's Generator*, IJCSNS International Journal of Computer Science and Network Security, Vol. 6, No. 8A, August 2006, 161-165.
- [11] *NIST statistical tests*, available on: <http://csrc.nist.gov/groups/ST/toolkit/rng/documents/nissc-paper.pdf>
- [12] NIST SP 800-22. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, available on: <http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf>

- [13] Harasymchuk O.I., *Test pulse sequence generators for radiation dosage metres*, Herald of Lviv Polytechnic National University, Thermal engineering. Environmental engineering. Automatisation, No. 06, 2004, 186-192.
- [14] *Pearson's chi-squared test*, available on: http://en.wikipedia.org/wiki/Pearson%27s_chi-squared_test.

ZBIGNIEW KOKOSIŃSKI*

A PARALLEL DYNAMIC PROGRAMMING ALGORITHM FOR UNRANKING SET PARTITIONS

RÓWNOLEGLY ALGORYTM PROGRAMOWANIA DYNAMICZNEGO DO KONWERSJI LICZB PORZĄDKOWYCH W PODZIAŁY ZBIORU

Abstract

In this paper, an $O(n)$ parallel algorithm is presented for unranking set partitions in Hutchinson's representation. A simple sequential algorithm is derived on the basis of a dynamic programming paradigm. In the parallel algorithm, processing is performed in a dedicated parallel architecture combining certain systolic and associative features. The algorithm consists of two phases. In the first phase, a coefficient table is created by systolic computations. Then, n subsequent elements of a partition codeword are computed, in $O(1)$ time each, through associative search operations.

Keywords: set partition, unranking algorithm, associative algorithm, parallel dynamic programming

Streszczenie

W artykule przedstawiono równoległy algorytm o złożoności $O(n)$ dla wyznaczania podziału zbioru $\{1, \dots, n\}$ w reprezentacji Hutchinsona na podstawie jego liczby porządkowej. Prosty algorytm sekwencyjny opiera się na paradygmacie programowania dynamicznego. Algorytm równoległy łączy w sobie cechy programowania systolicznego i asocjacyjnego. Algorytm składa się z dwóch kroków. W pierwszej kolejności, za pomocą obliczeń systolicznych, wyznaczana jest tablica współczynników, zwanych liczbami Williamsona. Następnie, przez asocjacyjne wyznaczanie maksimum zbioru liczb, obliczanych jest n kolejnych elementów reprezentujących podział, każdy w czasie $O(1)$.

Słowa kluczowe: podział zbioru, algorytm konwersji liczby porządkowej w obiekt, algorytm asocjacyjny, równoległe programowanie dynamiczne

* Ph.D. Zbigniew Kokosiński, email: zk@pk.edu.pl, Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

1. Introduction

Listing, ranking and unranking of combinatorial objects is an important problem in computer science and engineering in many application areas [12, 13].

The most common application of unranking algorithms is ‘translation’ of randomly generated of integer ranks for a class of combinatorial objects into random objects, that are used in generation of instances used in software testing, Monte Carlo algorithms etc. They can also be applied for the generation of ordered sequences of combinatorial objects, in adaptive parallel generation algorithms, and in task partitioning in optimization algorithms constructed on the basis of exhaustive search. Another application of ranking and unranking algorithms is enumerative coding in which selected classes of equiprobable messages are translated into classes of combinatorial objects, providing unique object’s indexing within the class. The assigned index and the code of the enumerative class represent the compressed data for the message [27]. Unranking algorithms can also be used for enumeration [3], crossover operations on chromosomes in genetic algorithms [28], metaheuristics etc.

Partitions are widely used to solve optimization problems in bioinformatics, forensic science, and scheduling [7]. In computational molecular biology partitions play an important role in understanding the role of genes in determining global characteristics of species. In a multi-state distribution system, the overall quality of service can be maintained by quick enumeration partitions of the variables used in decision diagrams that model the system [3].

The first known algorithm for generating (n, m) -partitions, $1 \leq m \leq n$, i.e partitions of n -element set into at most m nonempty blocks is due to Hutchinson [8]. Over the following years, a number of sequential algorithms was developed [4, 6, 11, 22]. An increasing interest in parallel computing also resulted in the development of many parallel algorithms for the partition generation problem in various models of computations [5, 16, 19, 21, 25]. The structure of the set of partitions was investigated and new ranking/unranking techniques were developed satisfying some particular requirements [15, 29]. A fast hardware rank to partition converter has been proposed in [3].

Parallelization of the generation algorithm on the set of objects level is possible by means of an adaptive scheme. The linearly ordered sequence of objects is divided into k subsequences of equal size, where k is the number of available processors. An unranking algorithm is used for determining the first object of each subsequence.

In the present paper, we propose a new parallel algorithm for unranking set partitions. The dynamic programming technique is used which was successfully applied in many application areas. So far a dynamic programming paradigm was used in a number of sequential algorithms for unranking combinations [17], partitions [15, 29], t-ary trees [20, 22] and some other combinatorial objects [9, 10]. Although, in general, unranking problems are inherently sequential, a portion of computations can be parallelized. However, many sequential algorithms are not suitable for parallelization. Until now, parallel algorithms were proposed for unranking combinations and t-ary trees [18, 20, 22].

The rest of the paper is organized as follows. The next section introduces a representation of set partitions. Then, a sequential algorithm for unranking of set partitions in Hutchinson’s representation is presented in section 3. Section 4 describes an associative algorithm developed on the basis of a parallel dynamic programming method.

2. Representation of set partitions

Let us introduce the basic notions used throughout this paper.

Let $\langle A_i \rangle_{i \in I}$ denote an *indexed family of sets* $A_i = A$, where: $A = \{1, \dots, m\}$, $I = \{1, \dots, n\}$, $1 \leq m, n$. Any mapping f , which ‘chooses’ one element from each set A_1, \dots, A_n is called a *choice function* of the family $\langle A_i \rangle_{i \in I}$ [23]. With additional restrictions, we can model by choice functions various classes of combinatorial objects [9].

Below, we define choice functions ρ corresponding to partition codewords known from literature [8, 29] (see Table 1).

If additional conditions: 1. $a_1 = 1$ and 2. $a_i \in \{1, \dots, \max[a_1, \dots, a_{i-1}] + 1\}$, for $2 \leq i \leq n$, and $i \in I$, are satisfied, then any choice function $\rho = \langle a_i \rangle_{i \in I}$ that belongs to the indexed family $\langle A_i \rangle_{i \in I}$ is called the *partitioning choice function* of this family (r -sequence). Set of all partitioning choice functions contains representations of all m -block partitions of the set A . In Hutchinson’s representation of partitions we deal in fact with indexed sets $R_i = \{1, \dots, i\} \subseteq A_i$.

Table 1

Sequences of all (m, n) -partitions ($1 \leq m \leq n = 4$, $B(n) = 15$)

Rank	r -sequence	Rank	r -sequence	Rank	r -sequence
1	1 1 1 1	6	1 2 1 1	11	1 2 2 3
2	1 1 1 2	7	1 2 1 2	12	1 2 3 1
3	1 1 2 1	8	1 2 1 3	13	1 2 3 2
4	1 1 2 2	9	1 2 2 1	14	1 2 3 3
5	1 1 2 3	10	1 2 2 2	15	1 2 3 4

The number of different m -block partitions of n -element set for $m \geq 2$ is called the Stirling number of the second kind:

$$S(n, m) = S(n-1, m-1) + mS(n-1, m), \quad \text{for } 0 < m < n \quad (1)$$

where $S(n, n) = 1$, for $n \geq 0$ and $S(n, 0) = 0$, for $n > 0$. The above formula describes construction of the Stirling triangle.

Let us recall the concept of Williamson numbers [15, 29]. The number of different at most m -block partitions of the n -element set $\rho = \langle r_1, \dots, r_{n-\nu}, r_{n-\nu+1}, \dots, r_n \rangle \in \langle R_i \rangle_{i \in I}$ with constant $\langle r_1, \dots, r_{n-\nu} \rangle$ and $\max\{r_1, \dots, r_{n-\nu}\} = \mu$ is called Williamson number:

$$W_n^m(\nu, \mu) = \mu W_n^m(\nu-1, \mu) + W_n^m(\nu-1, \mu+1) \quad (2)$$

where $\mu, \nu \geq 1$, $W_n^m(1, \mu) = 1$, for $1 \leq \mu \leq m$, and $W_n^m(1, \mu) = 0$ for $\mu > m$.

The above recursive formula describes the construction of the Williamson triangle for a given value of m . In particular, when $n = v$ and $\mu = 1$, the following equation holds:

$$W_n^m(n, 1) = \sum_{l=0}^m S(n, l) \tag{3}$$

Table WT containing a part of the Williamson triangle, for $n = m = 6$, is shown in Table 2. Williamson numbers $W_6^6(v, \mu)$ are stored in elements $WT[v, \mu]$ of the table WT. If $n = m$, then the number of all partitions of n -element set with at most n blocks is called Bell number B_n :

$$B_n = W_n^n(n, 1) = \sum_{l=0}^m S(n, l) \tag{4}$$

The above formulas, (1)–(4) come from the literature. The terms ‘Williamson number’ and ‘Williamson triangle’ were first proposed by the author in [15]. Two-dimensional tables containing parts of Williamson triangle are used in dynamic programming unranking algorithms for set partitions.

Table 2

**Construction of the table WT for (m, n) -partitions
($1 \leq m \leq n = 6, B(n) = 203$)**

$j =$	1	2	3	4	5	6
$i =$						
1	1	1	1	1	1	1
2	2	3	4	5	6	
3	5	10	17	26		
4	15	37	77			
5	52	151				
6	203					

Let us now introduce the lexicographic order of the set of all choice functions of the family $\langle A_i \rangle_{i \in I}$.

For the given choice functions $\rho = \langle d_1, \dots, d_k \rangle$ and $\gamma = \langle g_1, \dots, g_k \rangle$, we say that δ is less than γ according to the *increasing lexicographic order*, if and only if there exists $i \in \{1, \dots, k\}$, satisfying $d_i < g_i$, and $d_j = g_j$, for every $j < i$.

For given choice functions $\delta = \langle d_1, \dots, d_k \rangle$ and $\gamma = \langle g_1, \dots, g_k \rangle$, we say that δ is less than γ according to the *decreasing lexicographic order*, if and only if there exists $i \in \{1, \dots, k\}$ satisfying $d_i > g_i$ and $d_j = g_j$, for every $j < i$.

3. Sequential dynamic programming algorithm

We will start from the construction of the sequential algorithm given in [15], which is a modification of the Williamson's algorithm [29]. In the algorithm UnrankPart, the table WT is used, which includes a part of a modified Williamson Triangle (see Table 2).

In the unranking algorithm given below, the table WT is already given. The following restrictions on pairs (i, j) , $k \leq n$ should be taken into account during construction of this table (2 and 3 optionally):

- 1) $n \leq n_{\max}$, where: n_{\max} is any natural number (size of the triangle),
- 2) $1 \leq k \leq k_{\max} \leq n_{\max}$ (columns from 1 to k_{\max}),
- 3) $0 \leq n - k = r \leq n_{\max} - k_{\min}$ (rows from 1 to $r + 2$).

Algorithm UnrankPart

Input: n – number power of the set, Index – rank of the choice function ρ representing set partition ($1 \leq \text{Index} \leq B(n)$, Table WT with elements $WT[i, j]$ containing Williamson numbers $W_n^n(i, j)$, for $1 \leq j \leq n$).

Output: Table R with choice function ρ .

Method: Computations proceed with partitions ranks in increasing lexicographic order. In each step 6.1, the element $mWT[i, j]$ with maximum m satisfying the given inequality is selected and next value $R[n - i + 1]$ is obtained in step 6.1.2. After n iterations we obtain the partition ρ with given index.

1-5 initialization phase

1. Index=Index-1;
2. **for** i=1 **to** n **do** R[i]=1;
3. i=n-1;
4. j=1;
5. m=j;
6. **while** Index > 0 **do**
 - 6.1. **if** $mWT[i, j] \leq \text{Index}$
 - then**
 - 6.1.1. Index=Index- $mWT[i, j]$;
 - 6.1.2. $R[n-i+1]=R[n-i+1]+m$;
 - 6.1.3. **if** $j < R[n-i+1]$ **then** $j=R[n-i+1]$;
 - 6.1.4. $m=j$;
 - 6.1.5. $i=i-1$;
 - else**
 - 6.1.6. $m=m-1$;
 7. **return** R.

Example 1

For the input data given below compute table R using the algorithm UnrankPart.

Input:

$n=4$, Index(R)=10.

Solution:

Index=9, R=[1,1,1,1], i=3, j=1, m=1.
 Index=9 > 0, 1*WT[3,1]=5 ≤ 9, Index=4, R[2]=2, j=2, m=2, i=2.
 Index=4 > 0, 2*WT[2,2]=6 > 4, m=1.
 Index=4 > 0, 1*WT[2,2]=3 ≤ 4, Index=1, R[3]=2, j=2, m=2, i=1.
 Index=1 > 0, 2*WT[1,2]=2 > 1, m=1.
 Index=1 > 0, 1*WT[1,2]=1 ≤ 1, Index=0, R[4]=2, j=2, m=2, i=0.

Output: R=[1,2,2,2].

Theorem 1

Algorithm UrankPart is correct and its asymptotic computational complexity is $O(n^2)$.

Proof

The set of all $B(n)$ partitions can be depicted in the form of a rooted ordered tree of height n ($n + 1$, when root is included). The structure of this tree reflects the recursive structure of the partition set (see Fig. 1). Each node of the tree is labeled by a Williamson number that is equal to the number of leaves of the subtree rooted in this node (labels 1 for leaves are omitted in Fig. 1). Nodes with isomorphic subtrees are labeled with the same Williamson numbers.

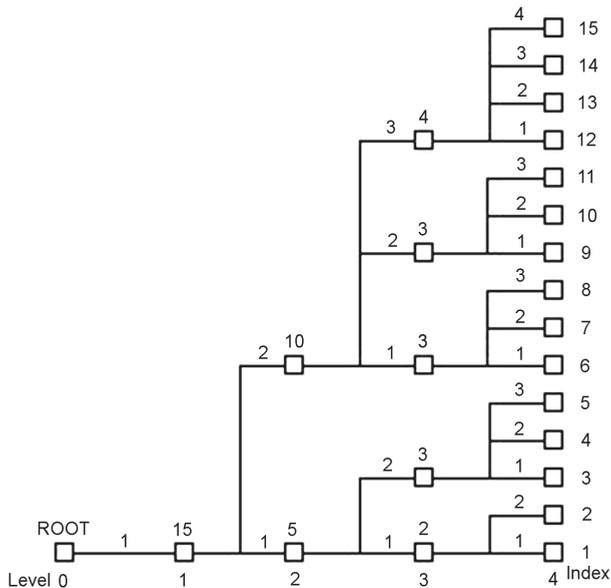


Fig. 1. The rooted ordered tree of all $B(4) = 15$ partitions with edge and node labels

Edges connecting ancestors with their descendants are also labeled. The single edge coming from the root is labeled 1. The number of descendants of a given node exceeds by one the highest edge label on the path leading from the root to this node. Edges connecting a parental node with its children receive successive integers as their labels. A sequence of edge labels on a path leading from the root to a leaf, represents a set partition. Traversing the tree in

preorder and listing all paths from the root to subsequent leaves – by sequences of edge labels – is equivalent to generation (enumeration) of all $B(n)$ partitions in increasing lexicographic order. Let us assign to all such paths their ranks in lexicographic order. Unranking the object with the given rank is equivalent to finding in the tree the path corresponding to $Index$, $1 \leq Index \leq B(n)$. We determine the path with rank $Index$ by examining sizes of ordered subtrees on the consecutive levels starting from the root. In order to do this, the current relative $Index$ of the choice function ρ is compared with $mW_n^n(i, j)$, where $W_n^n(i, j)$ are node labels and $1 \leq m \leq j$. Maximum m satisfying inequality in step 6.1 of the algorithm UnrankPart is determined, $Index$ is updated and the next item of the required object is obtained. In each level i , no more than j comparisons are made. Single iteration with complexity $O(n)$ is repeated $O(n)$ times. Hence, the total complexity of the algorithm is $O(n^2)$.

4. Parallel dynamic programming algorithm

The original contribution of the paper is parallelization of the sequential algorithm. In the first phase, the table WT is precomputed by systolic computations according to formula (2). Thus, the table WT contains coefficients used in our parallel dynamic algorithm suitable for a hardware acceleration in an associative computational structure.

A simple parallel unranking algorithm implementing associative memory search operations (*no greater than* and *maximum*) may be sketched as follows:

Algorithm UnrankPart-Par

Input:

n – number power of the set, $Index$ – rank of the choice function ρ representing set partition ($1 \leq Index \leq B(n)$), Table WT with elements $WT[i, j]$ containing Williamson numbers $W_n^n(i, j)$, for $1 \leq j \leq n$.

Output: Table R with choice function ρ .

Method: Computations proceed with partitions ranks in increasing lexicographic order. In order to determine elements of R , an associative search is used. In each step 5.1 all elements $WT[i, m]$, $1 \leq m \leq j$, satisfying the given inequality are selected. Then, element with maximum m coordinate is selected and next value $R[n-i+1]$ is obtained in step 5.4. After n iterations we obtain the partition ρ with given index.

1-4 initialization phase

1. $Index = Index - 1$;
2. $R[1] = 1$;
3. $i = n - 1$;
4. $j = 1$;
5. **while** $Index > 0$ **do**
 - 5.1. **search in parallel for** all $m = j$ **downto** 0: $mWT[i, j] \leq Index$;
 - 5.2. **select maximum** m ;
 - 5.3. $Index = Index - mWT[i, j]$;
 - 5.4. $R[n-i+1] = R[n-i+1] + m + 1$;

5.5. **if** $j < R[n-i+1]$ **then** $j=R[n-i+1]$;
 5.6. $i=i-1$;
 6. **return** R.

Example 2

For the input data given below compute table R using the algorithm UnrankPart-Par.

Input:

$n=4$, $\text{Index}(R)=10$.

Solution:

$\text{Index}=9$, $R[1]=1$, $i=3$, $j=1$.

$\text{Index}=9 > 0$, $1 * \text{WT}[3,1]=5 \leq 9$, $m=1$, $\text{Index}=4$, $R[2]=2$, $j=2$, $i=2$.

$\text{Index}=4 > 0$, $2 * \text{WT}[2,2]=6 > 4$, $1 * \text{WT}[2,2]=3 \leq 4$, $m=1$, $\text{Index}=1$, $R[3]=2$, $j=2$, $i=1$.

$\text{Index}=1 > 0$, $2 * \text{WT}[1,2]=2 > 1$, $1 * \text{WT}[1,2]=1 \leq 1$, $m=1$, $\text{Index}=0$, $R[4]=2$, $j=2$, $i=0$.

Output: $R=[1,2,2,2]$.

Theorem 2

Algorithm UnrankPart-Par is correct and its asymptotic computational complexity is $O(n)$.

Proof

The unranking algorithm is a parallel version of the algorithm UnrankPart. Correctness of the method results from the proof of Theorem 1. Parallel search in step 5.1 is organized in an associative manner. Value Index, which is a pattern register is simultaneously compared with all values $mW_n^n(i,j)$, $0 \leq m \leq j$. This reduces the search time to $O(1)$. The maximum value m in step 5.2 is computed associatively in $O(1)$ time. Then, the next value $R[n-i+1]$ is determined. Each iteration in step 5 has the time complexity $O(1)$. Hence, the total complexity of the algorithm is $O(n)$.

5. Concluding remarks

The presented algorithm UnrankPart-Par is the next in the line of unranking algorithms for classes combinatorial objects, developed by the author on the basis of parallel dynamic scheme and oriented on acceleration of the required computations by fast associative operations. It can be implemented in all cases when the time of unranking of (n, m) -partitions is of critical importance. Both systolic and associative components of the parallel computations are scalable. The $O(n)$ parallel implementation improves the computational complexity of the sequential unranking algorithm for set partitions UnrankPart by factor n . The original contribution of the present paper are also Theorems 1 and 2 with proofs of algorithms' finiteness and exactness.

References

- [1] Akl S.G., *Parallel computation: models and methods*, Prentice Hall, 1997, 475-509.
- [2] Akl S.G., Stojmenović I., *Generating combinatorial objects on a linear array of processors*, [in:] Zomaya A.Y. (editor): *Parallel Computing: Paradigms and Applications*, Int. Thompson Comp. Press, 1996, 639-670.
- [3] Butler J.T., Sasao T., *Hardware index to set partition converter*, Proc. ARC'2013, Lecture Notes in Computer Science, Vol. 7806, 2013, 72-83.
- [4] Djokić B. et al., *A fast iterative algorithm for generating set partitions*, The Computer Journal, Vol. 32, 1989, 281-282.
- [5] Djokić B. et al., *Parallel algorithms for generating subset and set partitions*, Proc SIGAL Int. Symposium on Algorithms, Tokyo 1990.
- [6] Er M.C., *A fast algorithm for generating set partitions*, The Computer Journal, Vol. 31, 1988, 283-284.
- [7] Hankin R.K.S., West L.J., *Set partitions*, J. of Statistical Software, Vol. 23, Code Snippet 2 (December 2007), available at: <http://www.jstatsoft.org/>
- [8] Hutchinson G., *Partitioning algorithms for finite sets*, Comm. ACM, Vol. 6, 1963, 613-614.
- [9] Kapralski A., *New methods for the generation of permutations, combinations and other combinatorial objects in parallel*, Journal of Parallel and Distributed Computing, Vol. 17, 1993, 315-326.
- [10] Kapralski A., *Modelling arbitrary sets of combinatorial objects and their sequential and parallel generation*, monograph, Studia Informatica, Vol. 21, No. 2 (40), 2000.
- [11] Kaye R., *A Gray code for set partitions*, Information Processing Letters, Vol. 5, 1976, 171-173.
- [12] Knuth D.E., *The art of computer programming. Fundamental algorithms*, Addison-Wesley, 3rd edition, 1997.
- [13] Knuth D.E., *The art of computer programming*, Pre-fascicle 3B. *Generating all partitions*, Addison-Wesley, 2004.
- [14] Kokosiński Z., *On generation of permutations through decomposition of symmetric groups into cosets*, BIT, Vol. 30, 1990, 583-591.
- [15] Kokosiński Z., *Circuits generating combinatorial objects for sequential and parallel computer systems*, Monografia 160, Politechnika Krakowska, Kraków 1993.
- [16] Kokosiński Z., *Mask and pattern generation for associative supercomputing*, Proc. 12th Int. Conf. Applied Informatics AI'94, Annecy 1994, 324-326.
- [17] Kokosiński Z., *Algorithms for unranking combinations and their application*, Proc. 7th Int. Conf. Parallel and Distributed Computing and Systems PDCS'95, Washington D.C., USA, 1995, 216-224.
- [18] Kokosiński Z., *Unranking combinations in parallel*, Proc. Int. Conf. Parallel and Distributed Processing Techniques and Applications PDPTA'96, Sunnyvale, CA, USA, Vol. I, 1996, 79-82.
- [19] Kokosiński Z., *On parallel generation of set partitions in associative processor architectures*, Proc. Int. Conf. Parallel and Distributed Processing Techniques and Applications PDPTA'99, Las Vegas, USA, Vol. III, 1999, 1257-1262.
- [20] Kokosiński Z., *A parallel dynamic programming algorithm for unranking t-ary trees*, Proc. PPAM'2003, Lecture Notes in Computer Science, Vol. 3019, 2004, 255-260.

- [21] Kokosiński Z., *A new algorithm for generation of exactly m -block set partitions in associative model*, Proc. PPAM'2005, Lecture Notes in Computer Science, Vol. 3911, 2006, 67-74.
- [22] Kokosiński Z., *Parallel enumeration of t -ary trees in ASC SIMD model*, Int. J. Computer Science and Network Security, Vol. 11, No. 12, 2011, 38-49.
- [23] Mirsky L., *Transversal theory*, Academic Press, 1971.
- [24] Semba I., *An efficient algorithm for generating all partitions of the set $\{1, 2, \dots, n\}$* , Journal of Information Processing, Vol. 7, 1984, 41-42.
- [25] Stojmenovič I., *An optimal algorithm for generating equivalence relations on a linear array of processors*, BIT, Vol. 30, 1990, 424-436.
- [26] Stojmenovič I., *On random and adaptive parallel generation of combinatorial objects*, Int. Journal of Computer Mathematics, Vol. 42, 1992, 125-135.
- [27] Tomic R.V., *Quantized indexing: background information*, Technical Report TR05-0625, 1st Works Corporation, June 25, 2005, 39.
- [28] Üçoluk G., *A method for chromosome handling of r -permutations of n -element set in genetic algorithms*, Proc. 4th Int. Conf. on Evolutionary Computing ICEC'97, Indianapolis, USA, 55-58, 1997.
- [29] Williamson S.G., *Ranking algorithms for lists of partitions*, SIAM J. of Computing, Vol. 5, 1976, 602-617.

KRZYSZTOF SCHIFF*

ANT COLONY OPTIMIZATION ALGORITHM FOR THE 0-1 KNAPSACK PROBLEM

ALGORYTM MRÓWKOWY DLA 0-1 PROBLEMU PLECAKOWEGO

Abstract

This article describes a new ant colony optimisation algorithm for the discrete knapsack problem with a new heuristic pattern, based on the ratio of the square of the profit coefficient to the square of the weight coefficient of the original problem. This new heuristic is used in order to choose objects that should be packed into the knapsack. This pattern was compared with two used in ant algorithms and which have been presented in the literature on the subject of ant colony optimisation algorithms for the 0-1 Knapsack Problem. The two other patterns are based on the ratio of the profit coefficient to the weight coefficient multiplied respectively by the total and the current knapsack load capacity. Results of tests under a wide range of ant algorithm parameters such as the number of cycles, the number of ants, the evaporation rate, and the load knapsack capacity are shown and discussed.

Keywords: knapsack problem, ant colony optimisation, heuristic algorithm

Streszczenie

W artykule przedstawiono algorytm mrówkowy dla dyskretnego problemu plecakowego z nową heurystyką wyboru obiektów i został on porównany z dwoma innymi algorytmami spotkanymi w literaturze przedmiotu pod względem uzyskiwanego całkowitego zysku z załadowanych do plecaka przedmiotów. Nowa heurystyka wyboru została wyrażona poprzez stosunek kwadratu zysku do kwadratu wagi wybranego przedmiotu, gdy dwie znane już heurystyki to stosunek zysku do wagi odpowiednio pomnożony przez całkowitą i bieżącą ładowność plecaka. W artykule przedstawiono wyniki przeprowadzonych testów dla szerokiego zakresu parametrów algorytmów mrówkowych takich jak: współczynnik parowania, liczba cykli, liczba mrówek, ładowności plecaka jak i dla różnej liczby dostępnych przedmiotów do załadunku.

Słowa kluczowe: problem plecakowy, algorytm mrówkowy, heurystyka

* Ph.D. Krzysztof Schiff, e-mail: kschiiff@pk.edu.pl, Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

1. Introduction

Many optimisation problems in decision-making can be presented as the 0-1 Knapsack Problem (KP) [1]. The 0-1 Knapsack Problem consists of loading objects in to a knapsack in such a way that the obtained total profit of all objects included in the knapsack is maximum and the sum of the weights of all packed objects does not exceed the total knapsack load capacity. Each object can be loaded or not loaded into the knapsack; this is the 0-1 decision concerning object loading. There are also other versions of this problem such as the Multi-dimensional 0-1 Knapsack Problem [10, 13–15] or the Multiple 0-1 Knapsack Problem [7–9, 11, 12].

The 0-1 Knapsack Problem is an NP-difficult (NP: non-polynomial) problem [2]. The exact solution to an NP problem is not obtained in a short period of time, computer algorithms take a great deal of time to arrive at a solution. The 0-1 Knapsack Problem can be solved by using the exact methods [4–6]. In order to obtain the solution for the 0-1 KP in a short period of time, heuristic algorithms are used. Such algorithms, which are based on the behaviour of ants, are taken into consideration in this paper.

Ant algorithms are very suitable for NP-complete problems [17]. Ants construct solutions to the problem and the best solution from their work is remembered in each algorithm cycle. Ants construct their solution using a pheromone, which is a chemical signal. The quantity of the pheromone connected with the objects, which constitute a solution to the problem, varies during the algorithm action. The quantity of the pheromone decreases for all objects and is shown as evaporation. The quantity of the pheromone rises when an additional quantity is added to all objects, which constitutes the best solution. A greater quantity of the pheromone means a greater probability of the object being selected by ants during their search for the optimal solution to the problem. A decision on element selecting depends not only on the quantity of the pheromone, but also on heuristic information, which can be expressed with a different kind of pattern. The heuristic pattern is additional information for ants about the problem, which helps them to find a better solution in comparison to a situation when this heuristic information is not used during optimal construction of a solution by all ants.

A mathematical model of the 0-1 Knapsack Problem is presented in section 2, a general pseudo-code of the Ant Colony Optimisation algorithm is discussed, a proposed heuristic pattern and two other patterns which have been used in ant algorithms, are formulated in section 3. The results of the conducted tests are shown and discussed in section 4. These experiments compare the action of algorithms with all three heuristic patterns – the new version presented in this paper, the static version used in papers [7–9, 11] and the dynamic version used in papers [10, 12–16].

In the static heuristic pattern, the total load knapsack capacity does not change at all during the algorithm's operation, whereas in the dynamic heuristic pattern, the current load knapsack capacity changes constantly since objects are constantly added to the knapsack in each cycle of the ant algorithm.

2. Mathematical formalisation

The mathematical model of the 0-1 Knapsack Problem can be stated in the same way as in paper [3]:

$$\max \sum_{i=1}^n z_i x_i \quad (1)$$

with constraints:

$$\sum_{i=1}^n w_i x_i \leq C \quad (2)$$

where:

- C – is the total knapsack load capacity,
- z_i – is the profit on an object i ,
- w_i – is the weight of an object i ,
- C, z_i, w_i – are all integers and positive numbers,
- and $x_i = 0$ – when an object i has not been loaded into a knapsack,
- or $x_i = 1$ – when an object i has been loaded into a knapsack.

The knapsack has its own capacity C . Each object has its own weight w_i . The total weight of all objects which have been packed into the knapsack should not exceed the total knapsack load capacity. All objects should be selected to be packed so that the total profit Z of all objects which have been packed into the knapsack should be maximal. An object o_i has been loaded into a knapsack or it has not been loaded; thus a variable x_i corresponding to this object has two states or values $\{0, 1\}$. There are n objects to be loaded into the knapsack $\{o_1, o_2, \dots, o_n\}$. Each object o_i has its own profit z_i and its own weight w_i . The set $N = \{o_1, o_2, \dots, o_n\}$ is the set of all available objects which can be loaded into the knapsack.

When an object o_i goes into the knapsack, the latter's capacity is decreased by the weight of this loaded object o_i . This new value for the available knapsack capacity is called the current knapsack capacity V_c . If the weight of any object is greater than the current knapsack capacity V_c , then that object cannot be loaded into the knapsack and is removed from the list N . After an object o_i is loaded into the knapsack, a new list N_j has to be compiled. This new list N_j is obtained from the preceding list N_p , which was stated before the object o_i was loaded into the knapsack. The solution to the problem constitutes the objects loaded into the knapsack, i.e. objects which have been included in the set S .

The total weight of all packed objects in the knapsack should not be larger than the knapsack load capacity and the total profit of all packed objects in the knapsack should be maximal. At the beginning of the ants' work, the knapsack is empty; thus the set S is also empty ($S = \{\}$). Next, all objects from the set N are verified in terms of their weights and the knapsack load capacity. A new object o_i can be selected to be packed into the knapsack only from those objects whose weight is less than the current knapsack load capacity. After a selected object o_i has been packed into the knapsack, the current knapsack capacity is reduced. There is an object o_i inside the knapsack $S = \{o_{i1}\}$.

In some steps j , other objects are loaded into the knapsack, so the contents of the set S_j are $\{o_{i1}, o_{i2}, \dots, o_{ij}\}$. If the set $S_k = \{o_{i1}, o_{i2}, \dots, o_{ik}\}$ is the solution to the problem, then the set $S_j = \{o_{i1}, o_{i2}, \dots, o_{ij}\}$ is a partial solution to the problem or a solution under construction, $j < k$. The final contents of the set S_k are obtained as a result of the selection of objects from available objects N . When the next object o_{im} is selected, then state S_j changes to another state

$S_m = \{o_{i1}, o_{i2}, \dots, o_{ij}, o_{im}\}, j, m < k$. The total current profit of objects is Z_j and Z_m according to the states j and m . After an object o_j is packed into the knapsack, the current knapsack load capacity is less than it was before.

As a result of the reduced current knapsack load capacity, not all objects from the set N_i can now be loaded since their weights are too great in comparison to the current knapsack load capacity. Some objects from the set N_i are removed, since their weights are too great to be packed into the knapsack, and thus a new set N_j of available objects is obtained. The set N_j , from which a new object o_j can be selected in order to be packed into the knapsack, is called the neighbourhood of state S_i . Objects which constitute this neighbourhood N_j come from neighbourhood N_i ; these objects came in turn from neighbourhood N_p , with a weight less than or equal to the current knapsack load capacity V_c .

3. Structure of the ACO algorithm

In ant algorithms a colony of artificial ants is looking for a good solution to the investigated problem. The pseudo-code of the ACO algorithm is presented as procedure 1. Each artificial ant constructs an entire solution to the problem in a certain number of steps; at each step there is an intermediate solution, a partial solution or a state. In each step, each ant k goes from one state i to another state j and thus constructs a new intermediate solution. At the end, the entire solution will have been obtained in a certain number of steps. At each step, each ant k takes into consideration a set of feasible expansions to its current state and moves to one of these in probability. This set of feasible expansions is called a neighbourhood.

In the presented algorithm for the 0-1 Knapsack Problem, at each state i there is a partial solution S_i of the Knapsack Problem; each ant selects the next object o_i from the set N_i of available objects, goes to the next state j and adds this selected object to a partial solution S_j in order to construct, at the end of the algorithm operation, the entire solution S to the 0-1 Knapsack Problem. At the end of the algorithm operation, the set of objects S constitutes a solution to the 0-1 Knapsack Problem. Each ant k starts with an empty set S and successively adds to this set objects selected one after the other with probability p_j moving from one state i to another state j . At each state i there are certain objects in set S_i which constitute a partial solution. Each ant, in order to construct a solution, uses common information which is encoded in pheromone trails τ_j . Each ant also deposits pheromones on all objects included in the knapsack when a solution has been found. The quantity of the pheromones $\Delta\tau$ deposited depends on the quality of this solution Q . Each ant's move also depends on the so-called attractiveness of the move μ_j . In order to avoid a very rapid convergence to a locally optimal solution, the evaporation mechanism $\tau = \rho\tau$ is used. Over time, the pheromone trail evaporates, thus reducing its attractive strength. Each ant k moves from one state i to another state j according to a transition probability rule p_j :

$$p_j = \begin{cases} \frac{\tau_j^\alpha \mu_j^\beta}{\sum_{j \in N_i} \tau_j^\alpha \mu_j^\beta}, & \text{for } j \in N_i \\ 0, & \text{for } j \notin N_i \end{cases} \quad (3)$$

ACO procedure for the 0-1 Knapsack Problem

```

begin
  while (a cycle exists) do
    while (an ant  $k$ , which has not yet worked, exists) do
      while ( $V_c \geq 0$ ) do
        select a next object  $o_j$  from  $N_i$  with probability  $p_j = \begin{cases} \frac{\tau_j^\alpha \mu_j^\beta}{\sum_{j \in N_i} \tau_j^\alpha \mu_j^\beta}, & \text{for } j \in N_i \\ 0, & \text{for } j \notin N_i \end{cases}$ 

        add a selected object to a partial solution  $S = S + \{o_j\}$ 
        update the current knapsack load capacity  $V_c = V_c - w_j$ 
        update the profit  $Z = Z + z_j$ 
        update the neighbourhood of the current state  $N_i = \{o_i : w_i \leq V_c\}$ 
      end
      remember the best solution if a better solution has been found
    end
    remember a global best solution if a better solution has been found
    use an evaporation mechanism  $\tau = \rho\tau$ 
    update pheromone trails  $\tau = \tau + \Delta\tau$ 
  end
end.

```

using the pheromone trail τ_j and the attractiveness μ_j of the move. The pheromone trail τ_j is useful information, deposited by other ants, about their usage of object j in the past. The attractiveness μ_j is the desire to select an object j from the neighbourhood N_i of the current state. The attractiveness μ_j enables better selection of an object from all available objects which constitute the neighbourhood N_i of the current state and which can be added to the solution under construction. The neighbourhood N_i of state i is composed of objects which can be added to a constructed partial solution. At the start of the ants' work, all objects can be added to a partial solution of the problem, i.e. to a solution of a problem under construction. The number of these objects is reduced afterwards not only because of their inclusion in the partial solution S , but also because some of these objects cannot be added to a solution which is under construction, since these objects already fail to satisfy solution constraints. Only those objects which still satisfy solution constraints can be added to a constructed partial solution. The partial solution of the problem is a part of a solution or a solution under construction. The partial solution is a subset of the objects which constitute an entire solution to the problem. Parameters α and β , which are used in the transition probability rule p_j expressed by formula (3), indicate how important the pheromone trail τ_j and the attractiveness μ_j are during transitions from one state to another. After a solution has been found, each ant deposits some quantity $\Delta\tau$ of pheromones on all objects which constitute the solution S , in accordance with the pattern:

$$\tau = \tau + \Delta\tau \quad (4)$$

A quantity of deposited pheromones $\Delta\tau$ is expressed as:

$$\Delta\tau = f(Q) = \frac{1}{1 + \frac{z_{best} - z}{z_{best}}} \quad (5)$$

Thus those objects which were included into a solution have received an additional quantity of pheromones and can be selected afterwards with a higher probability than other objects.

An evaporation mechanism is incorporated into ant algorithms in order to avoid too rapid a convergence to a suboptimal solution. The intensity of evaporation is controlled by the parameter ρ . The quantity of the pheromone on each object is updated at the end of each cycle in accordance with the pattern:

$$\tau = \rho\tau, \quad \rho \in (0, 1] \quad (6)$$

Three ant colony optimisation algorithms were implemented. They are called:

1) AKA1, – with the attractiveness μ_j of the move expressed as:

$$\mu_j = \frac{z_j}{\frac{w_j}{V_c}} \quad (7)$$

2) AKA2, – with the attractiveness μ_j of the move expressed as:

$$\mu_j = \frac{z_j}{w_j^2} \quad (8)$$

3) AKA3, – with the attractiveness μ_j of the move expressed as:

$$\mu_j = \frac{z_j}{\frac{w_j}{C}} \quad (9)$$

where:

- C – is the total knapsack load capacity,
- V_c – is the current knapsack load capacity, $V_c = C - \sum_{g \in S_i} (w_g)$,
- S_i – is a partial solution,
- $\sum_{g \in S_i} (w_g)$ – is the weight of all objects which were included in the partial solution S_i ,
- w_j – is the weight of selected object j ,
- z_j – is the profit of selected object j ,
- μ_j – is the attractiveness of selecting an object j .

AKA1 and AKA3 are algorithms with static and dynamic heuristic patterns. The heuristic patterns for algorithms AKA1 and AKA3 are the ones used in ant algorithms for multi-knapsack systems and which have been adopted to the one-knapsack system in this paper. The heuristic pattern of the AKA2 algorithm is the one proposed in this paper. All tests were conducted on a computer with an Intel Celeron CPU, 1.7 GHz and 256 MB RAM.

4. Results of experiments

The first experiment was conducted for a different number of algorithm cycles $\{100, 200, 300, 400, 500, 600\}$ for 300 objects and for a knapsack load capacity equal to 3000, for an evaporation rate set at 0.95 and for a number of ants equal to 120. A profit z_i and a weight w_i were randomly generated for each object o_i in the range $\langle 1, 10 \rangle$ and $\langle 1, 100 \rangle$ adequately. Thus, it was possible to generate 1000 different objects (w_i, z_i) . Average values were obtained from 10 measurements for each number of cycles – these values are shown in Table 4.1 and in Fig. 4.1. The results show that the AKA2 algorithm yields a higher profit than the other two algorithms. These three values of profit converge when the number of cycles rises, but if the result must be obtained as fast as possible, the AKA2 algorithm is the most suitable. The AKA2 algorithm yields a higher profit than the other two algorithms when the number of cycles is equal to or less than 200.

Table 1

Profit in dependence on number of cycles

n. of cycles	100	200	300	400	500	600
AKA1	815.0	820.0	820.2	820.2	820.4	820.8
AKA2	819.9	820.9	821.1	821.2	820.9	820.9
AKA3	817.3	819.5	819.9	820.8	820.4	820.9

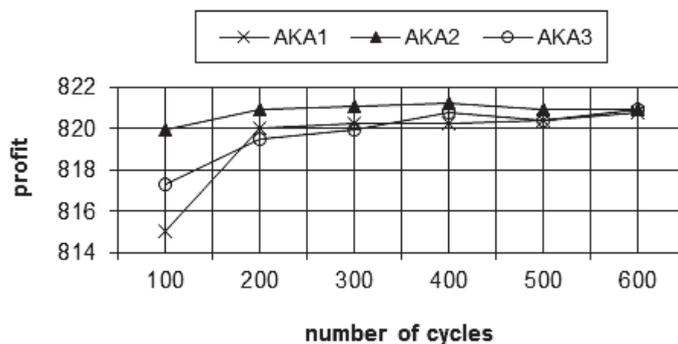


Fig. 1. Profit in dependence on number of cycles

Table 2

Profit in dependence on number of ants

n. of ants	20	40	60	80	100	120	140
AKA1	799.8	800.6	800.4	801.2	801.5	801.9	801.5
AKA2	801.2	801.6	802.1	801.8	802.1	802.3	802.2
AKA3	799.7	800.6	800.6	800.8	801.0	801.7	801.5

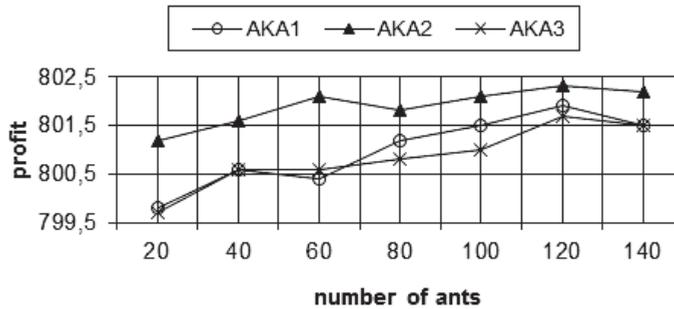


Fig. 2. Profit in dependence on number of ants

The second experiment was conducted for a constant number of cycles equal to 500 and for a different number of ants $\{20, 40, 60, 80, 100, 120, 140\}$, for a load knapsack capacity equal to 3000, an evaporation rate equal to 0.95 and a constant number of available objects set at 300. A profit z_i and a weight w_i were randomly generated for each object o_i in the range $\langle 1, 10 \rangle$ and $\langle 1, 100 \rangle$ adequately. Thus, it was possible to generate 1000 different objects (w_i, z_i) . Average values were obtained from 10 measurements for each different number of cycles; these values are shown in Table 2 and in Fig. 2. The results show that the AKA2 algorithm yields a higher profit than the other two algorithms for all numbers of ants. All profit values of three algorithms rise when the number of ants rises. There is also some degree of saturation – if the number of ants is higher than 60, there is no significant improvement in obtained profit.

Table 3

Profit in dependence of evaporation rate

ρ	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
AKA1	827.7	827.6	828.3	828.3	828.5	828.9	828.2	828.1	827.3
AKA2	829.4	829.5	829.8	829.5	829.7	829.5	829.5	829.6	828.9
AKA3	828.6	828.6	828.7	829.4	828.6	828.8	828.6	828.2	827.1

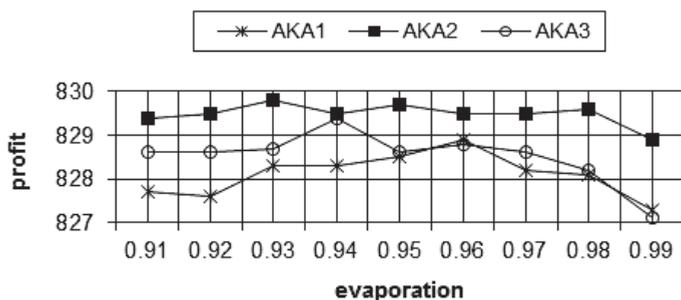


Fig. 3. Profit in dependence of evaporation rate

The third experiment was conducted for a constant number of cycles equal to 500 and for a constant number of ants equal to 100, for a constant load knapsack capacity set at 3000 and for a constant number of available objects set at 300, but for different evaporation rates (0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99). A profit z_i and a weight w_i were randomly generated for each object o_i in the range $\langle 1, 10 \rangle$ and $\langle 1, 100 \rangle$, respectively. Thus, it was possible to generate 1000 different objects (w_i, z_i). Average values were obtained from 10 measurements and shown in Table 3 and in Fig. 3. The results of the experiment show that the AKA2 algorithm yields a higher value of profit than the other two algorithms. The best range for the AKA2 algorithm is between 0.93 and 0.98.

Table 4

Profit in dependence on the number of ants for a number of cycles equal to 200

n. of ants	40	60	80	100	120
AKA1	833.8	833.8	834.3	834.0	835.0
AKA2	835.7	836.2	836.4	836.5	836.8
AKA3	833.9	833.4	833.9	834.7	834.5
p. of reference	837.2	837.2	837.2	837.2	837.2

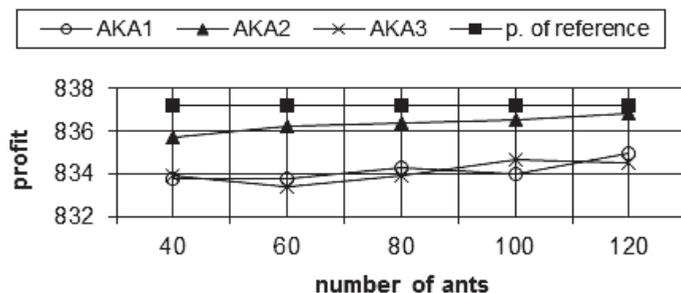


Fig. 4. Profit in dependence on the number of ants for a number of cycles equal to 200

The first experiment has shown that there is no need to set the number of cycles higher than 200 when the number of ants equals 120. Therefore, the number of cycles can be set lower without lowering the quality of the solution. In Tables 4 and 5 and in Fig. 4 and 5, the results for the numbers of cycles equal to 200 and 300 and for different numbers of ants have been shown. The point of reference is the profit obtained when the number of cycles and ants were set at 500 and 120, respectively. This point of reference is achieved when the number of cycles and the number of ants are set at 300 and 60, respectively. Thus, there is no need to set the number of cycles at 500 and the number of ants at 120 and wait longer in order to find a solution to the problem.

Table 5

Profit in dependence on number of ants for a number of cycles equal to 300

n. of ants	40	60	80	100	120
AKA1	835.0	834.8	834.9	835.0	835.7
AKA2	836.5	837.2	837.0	836.9	836.9
AKA3	834.6	835.2	835.7	834.8	835.7
p. of reference	837.2	837.2	837.2	837.2	837.2

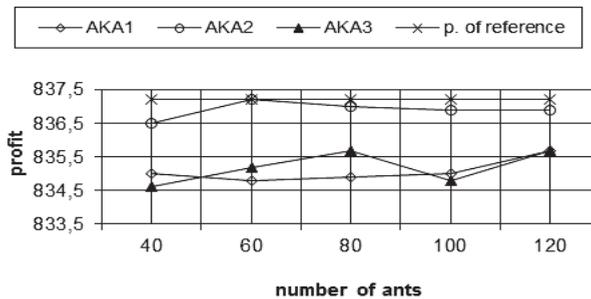


Fig. 5. Profit in dependence on number of ants for a number of cycles equal to 300

The next experiment concerns a profit in dependence on the knapsack load capacity. The number of cycles was set at 300, the number of ants to 80, the evaporation rate to 0.95, the number of objects to 300, the knapsack load capacity consecutively to (1000, 2000, 3000, 4000, 5000). A profit z_i and a weight w_i were randomly generated for each object o_i in the range $\langle 1, 10 \rangle$ and $\langle 1, 100 \rangle$, respectively. Thus it was possible to generate 1000 different objects (w_i, z_i) . Average values were obtained from 10 measurements and are shown in Table 6 and in Fig. 6. The AKA2 algorithm yields higher profits than the other two algorithms and its superiority is applicable for all knapsack load capacity values.

Table 6

Profit and differences in profit for variable load knapsack capacity

Load capacity	1000	2000	3000	4000	5000
AKA1	466.7	664.1	817.2	946.4	1058.3
AKA2	467.7	665.2	817.7	947.0	1059.7
AKA3	466.6	664.1	816.6	946.3	1058.4
AKA1-AKA3	0.1	0.0	0.6	0.1	-0.1
AKA2-AKA3	1.1	1.1	1.1	0.7	1.3
AKA3-AKA3	0.0	0.0	0.0	0.0	0.0

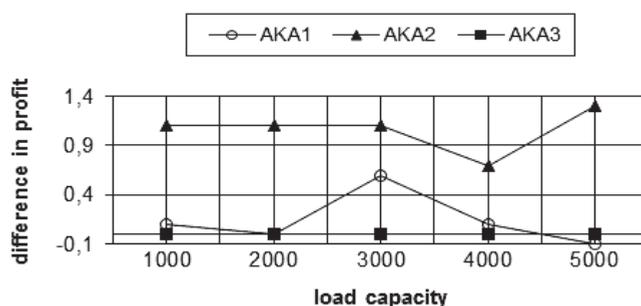


Fig. 6. Differences in profit for variable knapsack load capacity

The following experiment concerns profit and the number of objects. The number of cycles was set at 300, the number of ants at 80, the evaporation rate at 0.95, the knapsack load capacity at 3000 and the number of objects consecutively at (100, 200, 300, 400, 500). A profit z_i and the weight w_i were randomly generated for each object o_i in the range $\langle 1, 10 \rangle$ and $\langle 1, 100 \rangle$, respectively. It was possible to generate 1000 different objects (w_i, z_i). Average values were obtained from 10 measurements – these are shown in Table 7 and in Fig. 7. The AKA2 algorithm yields higher profits than the two other algorithms and rises when the number of objects rises.

Table 7

Profits for different numbers of objects

n. of objects	100	200	300	400	500
AKA1	482.5	672.8	817.2	936.3	1048.5
AKA2	482.5	673.0	817.7	938.6	1051.7
AKA3	482.6	672.6	816.6	937.1	1048.6
AKA1-AKA1	0.0	0.0	0.0	0.0	0.0
AKA2-AKA1	0.0	0.2	0.5	2.5	3.2
AKA1-AKA1	0.1	-0.2	-0.6	0.8	0.1

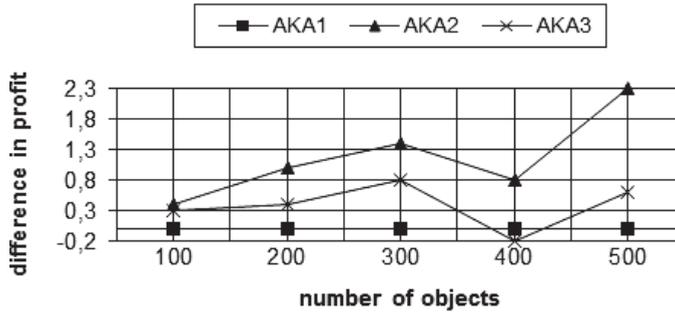


Fig. 7. Differences in profit for different entry numbers of objects

The last (but not least) experiment concerns profit and the number of objects. The number of cycles was set at 300, the number of ants at 80, the evaporation rate at 0.95, the knapsack load capacity at 3000 and the number of objects consecutively at (100, 200, 300, 400, 500). A profit z_i and the weight w_i were randomly generated for each object o_i in the range $\langle 1, 10 \rangle$ and $\langle 1, 10 \rangle$, respectively. Thus, it was possible to generate 100 different objects (w_i, z_i) , so that objects were generated with the same characteristics (w_i, z_i) . Average values were obtained from 10 measurements for each different number of cycles – these values are shown in Table 8 and in Fig. 8. The AKA2 algorithm yields higher profits than the other two algorithms and rises when the number of objects rises. Objects with repeated characteristics (w_i, z_i) do not influence the result of the experiment.

Table 8

Profits for different entry numbers of objects

n. of objects	100	200	300	400	500
AKA1	458.6	635.4	750.0	842.9	925.2
AKA2	459.0	636.4	751.4	843.7	927.5
AKA3	458.9	635.8	750.8	842.7	925.8
AKA1-AKA1	0.0	0.0	0.0	0.0	0.0
AKA2-AKA1	0.4	1.0	1.4	0.8	2.3
AKA3-AKA1	0.3	0.4	0.8	-0.2	0.6

The last experiment concerns a profit and a number of objects. The number of cycles was set at 300, the number of ants at 80, the evaporation rate at 0.95, the knapsack load capacity at 3000 and the number of objects consecutively at (100, 200, 300, 400, 500). A profit z_i and the weight w_i were randomly generated for each object o_i in the range $\langle 1, 100 \rangle$ and $\langle 1, 100 \rangle$ respectively. Thus, it was possible to generate 10,000 different objects (w_i, z_i) and there was almost no chance to generate objects with the same characteristic (w_i, z_i) . Average values were obtained from 10 measurements and are shown in Table 9 and in Fig. 9. The AKA2 algorithm yields higher profits than the other two algorithms and rises when the number of objects rises.

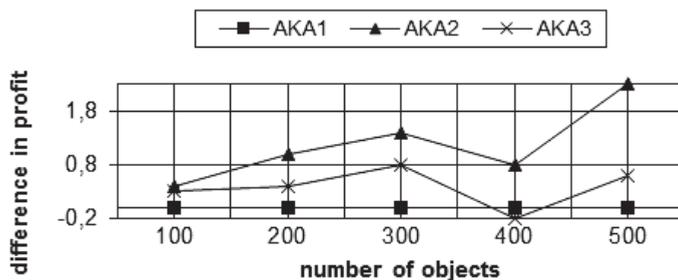


Fig. 8. Differences in profit for different entry numbers of objects

Table 9

Profit for the different entry numbers of objects

n. of objects	100	200	300	400	500
AKA1	4400.8	6465.7	7826.7	8921.9	9890.5
AKA2	4399.3	6466.3	7835.3	8943.5	9914.1
AKA3	4396.9	6465.7	7833.1	8922.9	9898.4
AKA1-AKA1	0.0	0.0	0.0	0.0	0.0
AKA2-AKA1	-1.5	0.6	8.6	21.6	23.6
AKA3-AKA1	-3.9	0.0	6.4	1.0	7.9

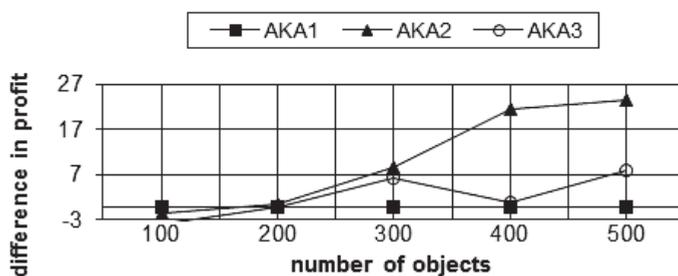


Fig. 9. Differences in profit for different entry numbers of objects

5. Conclusion

The experiments have shown that the AKA2 algorithm can find a solution with a higher total profit than the other two algorithms. Moreover, it can find this solution more rapidly, as it appears from the first experiment, in which the AKA2 algorithm yields a maximal profit after 200 cycles as compared to 600 cycles for the other two algorithms.

References

- [1] Kolthe P., Christensen H., *Planning in Logistics: A survey*, PerMIS'10 September 28–30, Baltimore 2010.
- [2] Garey M.R., Johnson D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco 1979.
- [3] Sysło M., Deo N., Kowalik J., *Algorytmy optymalizacji dyskretnej*, PWN, 1993.
- [4] Toth P., *Dynamic programming algorithms for the 0-1 knapsack problem*, Computing 25, 1980, 29-45.
- [5] Fayard D., Plateau G., *Algorithm 47: An algorithm for the solution of the 0-1 knapsack problem*, Computing 28, 1982, 269-287.
- [6] Kolesar P., *A branch and bound algorithm for the knapsack problem*, Management Science 13, 1967, 723-735.
- [7] Fidanova S., *Ant Colony Optimization for Multiple Knapsack Problem and Heuristic Model*, Kluwer Academic Publishers, 2004.
- [8] Fidanova S., *Ant Colony Optimization for Multiple Knapsack Problem and Model Bias*, [in:] Margenov S., Vulkov L.G., Wasniewski J. (Eds.), *Numerical Analysis and Its Applications*, LNCS, Vol. 3401, Springer, Berlin Heidelberg, 2005, 280-287.
- [9] Fidanova S., *Probabilistic Model of Ant Colony Optimization for Multiple Knapsack Problem*, In Lirkov I., Margenov S., Wasniewski J. (Eds.), LSSC 2007, LNCS 4818, Berlin 2008, 545-552.
- [10] Alaya I., Solnon C., Gheira K., *Ant algorithm for the multi-dimensional knapsack problem*, International Conference on Bioinspired Optimization Methods and their Applications, (BIOMA 2004), 2004, 63-72.
- [11] Fidanova S., *Heuristic for the multiple knapsack problem*, IADIS International Conference on Applied Computing, 2005, 255-260.
- [12] Boryczka U., *Ants and Multiple Knapsack Problem*, 6th International Conference on Computer Information Systems and Industrial Management Applications (CISIM'07), 2007, IEEE Computer Society, No. P2894, 2007, 149-154.
- [13] Ke L., Feng Z., Ren Z., Wei X., *An ant colony optimization approach for the multi-dimensional knapsack problem*, Journal of Heuristics, Vol. 16, No. 1, 2010, 65-83.
- [14] Shahrear I., Faizul B., Sohel R., *Solving the Multidimensional Multi-choice Knapsack Problem with the Help of Ants*, M. Dorigo et al. (Eds.), ANTS 2010, LNCS 6234, Berlin 2010, 312-323.
- [15] Ji J., Huang Z., Liu C., Liu X., Zhong N., *An Ant Colony Optimization Algorithm for Solving the Multidimensional Knapsack Problems*, [in:] Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IEEE Computer Society, Los Alamitos, 2007, 10-16.
- [16] Leguizamón G., Michalewicz Z., *A new version of ant system for subset problems*; [in:] Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, Vol. 2, 1999, 1458-1464.
- [17] Dorigo M., Caro D.G., Gambardella L.M., *Ant algorithms for discrete optimization*, Artificial Life, Vol. 5, No. 2, 1999, 137-172.

IRYNA VERBENKO, ROMAN TKACHENKO*

GANTRY AND BRIDGE CRANES NEURO-FUZZY CONTROL BY USING NEURAL-LIKE STRUCTURES OF GEOMETRIC TRANSFORMATIONS

NEURONOWO-ROZMYTE STEROWANIE SUWNICAMI BRAMOWYMI I POMOSTOWYMI Z UŻYCIEM STRUKTUR PRZEKSZTAŁCEŃ GEOMETRYCZNYCH O CHARAKTERZE NEURONOWYM

Abstract

Fuzzy logic is based on the use of natural language such as 'far or close', 'cold or hot' and etc. Its application range is very wide, from household appliances to the management of complex industrial processes. Many modern management tasks cannot be simply solved by classical methods because of the very great complexity of mathematical models. However, mathematical transformations are required for using the fuzzy logic theory on a computer and give a possibility to convert linguistic variables to their numerical value in the computer and vice versa. In this paper a gantry and bridge crane control system for managing carts swinging during transporting a load with high accuracy positioning during movement is presented. *T*-Controller fuzzy inference system as a base for crane management system is described and its main advantages in comparison with traditional systems are delineated. Schema of simplified crane model is introduced.

Keywords: gantry and bridge cranes, fuzzy inference systems, *T*-Controller, simplified model of the crane system

Streszczenie

Logika rozmyta bazuje na pojęciach języka naturalnego, takich jak „blisko lub daleko”, „zimny albo gorący” itp. Zakres zastosowania logiki rozmytej jest bardzo szeroki, począwszy od prostych urządzeń gospodarstwa domowego, a skończywszy na zarządzaniu złożonymi procesami przemysłowymi. Wiele współczesnych zadań planowania i sterowania nie da się rozwiązać za pomocą klasycznych metod, ze względu na zbyt dużą złożoność obliczeniową modelowanych procesów. Wprawdzie przekształcenia matematyczne stanowią wymóg podczas komputerowej realizacji tego typu zadań, jednak podejmująca ją logika rozmyta daje możliwość konwersji informacji zakodowanych w języku naturalnym na odpowiadające im wartości numeryczne. Przedmiotem niniejszego artykułu jest system sterowania suwnicami bramowymi i pomostowymi dla zarządzania wózkami obrotowymi podczas transportu ładunku o wysokiej dokładności pozycjonowania położenia. Jako podstawę systemu zarządzania suwnicą przedstawiono system wnioskowania rozmytego za pomocą *T*-regulatora rozmytego, podkreślając jego zalety w porównaniu z tradycyjnymi systemami wnioskowania rozmytego. W artykule zawarto również uproszczony schemat modelu suwnicy.

Słowa kluczowe: suwnice pomostowe i dźwigowe, system wnioskowania rozmytego, *T*-regulator, uproszony model mechanizmu suwnicy

* M.Sc. Iryna Verbenko, Doctoral Studies, e-mail:iryna.verbenko@outlook.com, Prof. D.Sc. Ph.D. Roman Tkachenko, Information Technologies of Publishing Business Department, Institute of Computer Science and Information Technologies, Lviv Polytechnic National University.

1. Introduction

The most impressive feature of human intelligence is the ability to make correct decisions in conditions of incomplete and fuzzy information. Construction of the models that reflect human thinking and their use in computer systems today is one of the most important problems in science.

Artificial intelligence that easily copes with the tasks of managing complex technical objects was helpless in such simple situations. For creating intelligent systems that can adequately communicate with a person needed a new mathematical tool to translate controversial statements in the language of strict mathematical formulas. Fuzzy logic is very useful in cases of high accuracy and repeatable operations when people might be stuck or even not able to perform such operation.

Let we have crane and we need to carry some object from point to point in the shortest time. The main problem in moving object via crane is the emergence of swinging it that leads to performance degradation, increasing transportation time and reliability of gear could cause an accidents.

One of the ways to improve efficiency of process of a control system is based on the using of human experience and the fuzzy modeling that used controller based on fuzzy logic and allows considering the adverse effects, caused by system nonlinearities.

Therefore, it is necessary to predict and to provide measures for the suppression of these swinging. Then based on fuzzy model predict power that will be applied to the crane depends on elapsed distance and swinging angle to eliminate swinging at the destination point.

In this paper the gantry and bridge cranes control system has been presented. The simplified model of the crane is used to simulate the crane motion. Two widely known Runge-Kutta and Euler methods of the integration of differential equations systems were applied to the equation of the crane motion.

Common knowledge of crane operator was employed to supervise the crane movement on it's path from start to finish. Consequently, the fuzzy logic was used to convert operator's knowledge into understandable for computer language. As a result, the fuzzy rules were created and described using the following crane values like angle, positiona and power.

T-Controller fuzzy inference system was chosen among other systems as a base for crane management system. The reason is that the *T*-Controller system is more accurate because it is based on geometrical transformation model and much faster than Mamdani and Takagi-Sugeno systems because required less time to setup and configure the fuzzy inference system.

2. Gantry and bridge cranes management process

Gantry and bridge cranes in most cases used to transport load from one point to another. They raise one cart with flexible cables that are attached to the 'head' of the crane. The 'head' of the crane moves along the horizontal track. If a crane is climbed and starts moving, the cart begins to swing.

To solve this problem, there are two ineffective ways. One of them is to reach the end point, and then wait for the crane swinging reaches an acceptable level. However, it is definitely required too much time, especially taking into consideration the environmental conditions. Cart should be loaded and swapped out in a minimum time for economic reasons.

Another option is such as to pick up the cart and move so slow that swing was minimal. However, this could be used not on a windy day and nonetheless, this method takes too much time. An alternative is to create a crane, which will be used additional cables to correct the position of the cart during transporting. Nevertheless, very few cranes use it due to the higher cost of the solution.

For these reasons, the majority of cranes are still using people-operators to control engine power. Operator compensates external influence and sets the power simultaneously, so that the load reached their destination as soon as possible. This is no easy task, but an experienced operator is able to get good results.

In this paper the crane management system based on common experience of crane operator will be presented. The rules which operator use during his or her work were modified to comprehensible for computer language. These rules were written in form of fuzzy rules and used in *T*-Controller fuzzy inference system. Such approach gives a possibility to apply human common sense, knowledge and experience for solving everyday tasks without adequate mathematical model, for instance, air conditioning, climate control systems for commercial buildings such as offices, and household appliances like washing machines that can tell how large a load is or refrigerators that use fuzzy logic to cool different sections correctly. Moreover, another example of applying of fuzzy logic technology is traffic management where there is difficult to create any models of traffic, but IF-THEN logic can deal with vague expectations and respond to situations. The gantry and bridge cranes management system is realized to help people in their everyday work and is based on the same acquired human principles.

3. The simplified model of crane movement simulation

The simplified model of the crane is used to simulate the crane movements. The crane system is realized as pendulum model. It is shown on the Fig. 1.

The following characteristics are taken into account in this model: x_c is a position of the cart relative to the start point, x_p and y_p are positions of the center of mass, M_c is the weight of the cart, M_p is the weight of the crane, α is an angle of the cart, F_c is an external force applied to the cart.

If the cart is pushed it will move in a positive direction to the right and an angle of the pendulum will take a positive value and will move counter clockwise. In addition, the zero angle corresponds to the vertical position of the lower pendulum [5].

This single pendulum can be represented as a system with one input parameter u (the power of a crane) and two output: a (angle of the cart) and x_c (cart position).

The mathematical equations of motion (1) and (2) can be calculated by using Lagrange equations with the total potential and the kinetic energies.

$$\ddot{x}_c = \frac{-(I_p + M_p J_p^2) B_{eq} \cdot \dot{x}_c + (M_p^2 J_p^3 + I_p M_p I_p) \sin(\alpha) \dot{\alpha}^2 + M_p I_p \cos(\alpha) B_p \dot{\alpha}}{(M_c + M_p) I_p + M_c M_p J_p^2 + M_p^2 J_p^2 \sin^2(\alpha(t))} + \frac{M_p^2 J_p^3 \cos(\alpha) \sin(\alpha) - (I_p + M_p J_p^2) \frac{\eta_g K_g^2 \eta_m K_t K_m \cdot \dot{x}_c}{R_m r_{mp}^2} + (I_p + M_p J_p^2) \frac{\eta_g K_g \eta_m K_t}{R_m r_{mp}} U_m}{(M_c + M_p) I_p + M_c M_p J_p^2 + M_p^2 J_p^2 \sin^2(\alpha)} \quad (1)$$

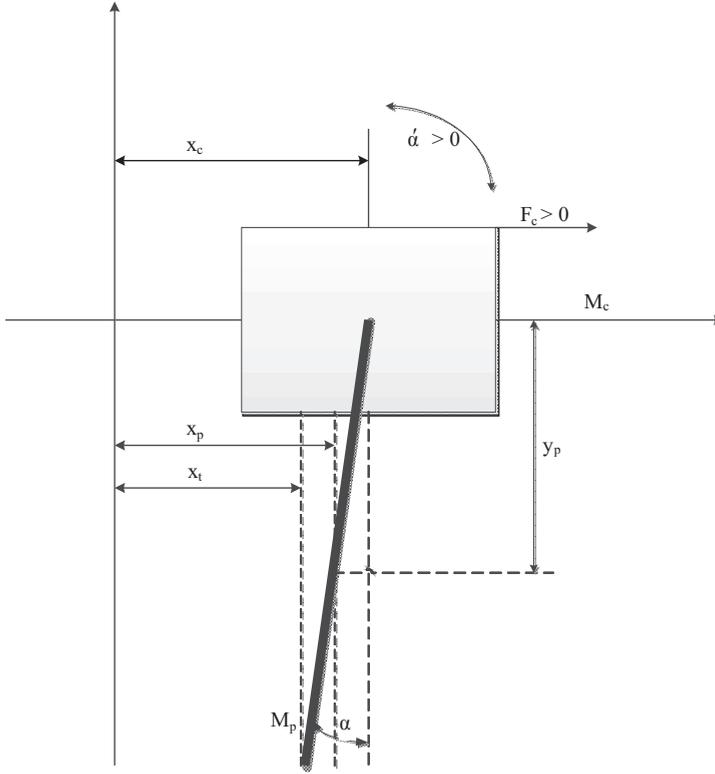


Fig. 1. The simplified model of the crane

$$\ddot{\alpha} = \frac{-(M_c + M_p)B_p \cdot \dot{\alpha} - M_p^2 l_p^2 \sin(\alpha) \cos(\alpha) \cdot \dot{\alpha} + M_p l_p \cos(\alpha) B_{eq} \cdot \dot{x}_c}{(M_c + M_p)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin^2(\alpha)} - \frac{(M_c + M_p)M_p g l_p \sin(\alpha) + (M_p l_p \cos(\alpha)) \frac{\eta_g K_g^2 \eta_m K_t K_m \cdot \dot{x}_c}{R_m r_{mp}^2} - M_p l_p \cos(\alpha) \frac{\eta_g K_g \eta_m K_t U_m}{R_m r_{mp}}}{(M_c + M_p)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin^2(\alpha)} \quad (2)$$

After the linearization of nonlinear single model of the crane is received the following system of differential equations that represented in (3) and (4).

$$\begin{bmatrix} \dot{x}_c(t) \\ \dot{\alpha}(t) \\ \ddot{x}_c(t) \\ \ddot{\alpha}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1.5216 & -11.6513 & 0.0049 \\ 0 & -26.1093 & 26.8458 & -0.0841 \end{bmatrix} \cdot \begin{bmatrix} x_c(t) \\ \alpha(t) \\ \dot{x}_c(t) \\ \dot{\alpha}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1.5304 \\ -3.5261 \end{bmatrix} \cdot U_m(t) \quad (3)$$

$$\begin{bmatrix} x_c \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_c(t) \\ \alpha(t) \\ \dot{x}_c(t) \\ \dot{\alpha}(t) \end{bmatrix} \quad (4)$$

Parameters of equations (1) and (2) are listed in the Table 1 with the values taken from [5, 6]. The linear equation of motion (3) and (4) are calculated after replacing in (1) and (2): $\cos(\alpha) = 1$ and $\sin(\alpha) = \alpha$, and using the parameters in the Table 1 [6].

Table 1

Parameters from (1) and (2) equations

Parameters	Description
$B_{eq} = 5.4$ [Nms/rad]	equivalent viscous damping coefficient as seen at the motor pinion
$B_p = 0.0024$ [Nms/rad]	viscous damping coefficient as seen at the pendulum axis
$\eta_g = 1$	planetary gearbox efficiency
$\eta_m = 1$	motor efficiency
$g = 9.81$ [m/s ²]	gravitational constant of earth
$I_p = 0.0078838$ [kgm ²]	pendulum moment of inertia
$J_m = 3.9001e-007$ [kgm ²]	rotor moment of inertia
$K_g = 3.71$	planetary gearbox gear ratio
$K_m = 0.0076776$	back electro-motive force (EMF) constant
$K_t = 0.007683$	motor torque constant
$l_p = 0.3302$ [m]	pendulum length from pivot to center of gravity
$M_c = 1.0731$ [kg]	lumped mass of the cart system, including the rotor inertia
$M_p = 0.23$ [kg]	pendulum mass
$R_m = 2.6$ [Ω]	motor armature resistance
$r_{mp} = 0.00635$ [m]	motor pinion radius

Such widely known methods of the integration of differential equations systems like numerical methods of Adams, Bashforth, Hamming, Runge-Kutta and Euler can be applied to the equation of the crane motion. The latter two methods, namely Euler and Runge-Kutta were used in crane management system to solve system of differential equations that represented in (3) and (4).

4. Traditional fuzzy inference systems and their advantages and disadvantages

Many modern management tasks cannot be simply solved by classical methods because of the very great complexity of mathematical models. However, mathematical transformations are required for using the fuzzy logic theory on a computer and give a possibility to convert linguistic variables to their numerical value in the computer and viceversa.

On the Fig. 2 the regions of most effective use of modern management techniques is shown. As can be seen, the classical control methods work well with fully deterministic control objects and deterministic environment. Fuzzy control methods are applied to systems with incomplete information and the high complexity of the control object.

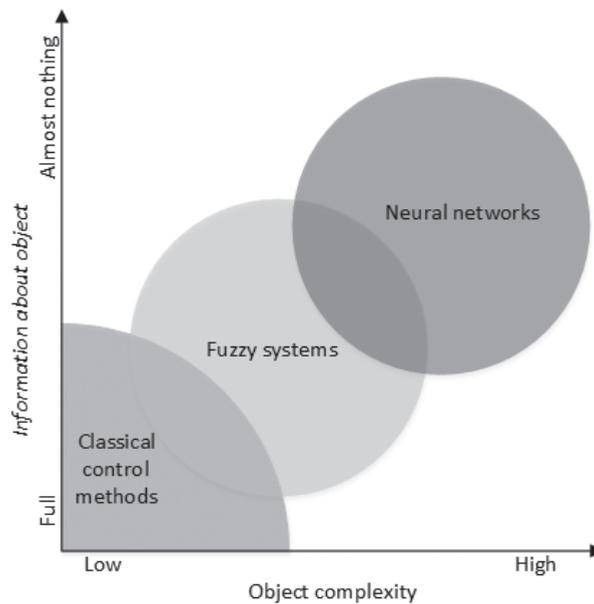


Fig. 2. Effective use of modern management techniques

Fuzzy inference system is a rule-based system that uses human common sense or expert knowledge to build a control system or model data. One of the FIS purposes is to extend of traditional data modelling and control methods.

Among traditional fuzzy inference systems algorithms the most famous and popular are Mamdani and Takagi-Sugeno algorithms, which, in particular, could be used to implement a system for the cranes cart operating.

The main advantages of Mamdani method are lack of standards in the rules constructing and fuzzy logic choice that based on experiments and extension expertise. This gives a possibility to describe the experience more intuitively and similarly to the human way. Mamdani algorithm has also the following common features as:

- Applicable when numerical data basis is incomplete and can be extended by human expert knowledge.

- Dependency on defuzzification method: same rules give different results depending on defuzzification function (Centre of Gravity, Minimum, First Maximum).
- Reasoning for defuzzification function choice based only on experiments.

Mamdani inference system is based on the following rule form: R: IF X IS A AND X IS B THEN Y IS C, where R is the rule number, X is the input number, A, B and C are linguistic variables, keyword IS marks a clause, keyword AND marks a conjunction.

Sugeno type FIS uses if sufficient number of basic numerical data is provided. It is effective in the calculation tasks and works good with optimization and adaptive methods that makes it widely used in the tasks of control. If the model parameters vary depending on the configuration and the nature or size of load then use of such fuzzy inference system is impractical because there is a need to create separate models for each of the partitioning area parts to form an appropriate control action, which can significantly affect the system performance.

As a result, traditional FIS include the following issues that affect the process:

- Good interpretability vs. good accuracy.
- Small number of input variables.
- Time consuming tuning.
- Dependency on defuzzification method: same rules give different results depending on defuzzification.
- Require sufficient data basis.

All of these issues limit the usage of the existing fuzzy systems for managing complex technical objects and there is a need to use another system that will compensate them.

One of the newly presented systems that compensate most disadvantages of traditional fuzzy inference systems is *T*-Controller.

5. *T*-Controller fuzzy inference system

T-Controller FIS is an original method of fuzzy logic. It is based on a geometrical data modeling conception (see Chapter 6).

T-Controller is a fuzzy controller – a way to create logical decisions for fuzzy logic system. It uses a fundamentally new method of defuzzification, which allows getting zero methodical error. *T*-Controller Workshop is an application that allows creating fuzzy controllers of *T*-Controller type using verbal user description and applies it to digital data [2].

T-Controller application is shown on the Fig. 3. The ‘Controller’ tab is opened on the screenshot that consist of ‘Input variables’, ‘Input variable membership functions’ and ‘Output variables’ sections. These sections will be described in more details further in this article. The next two tabs are ‘Data’ and ‘Results’. Data tab displays data from the uploaded file. On the Results tab a list of expected and predicted values after fuzzy processing is shown. This list could be saved to user’s PC. Additionally, the following errors like MAD (Median absolute deviation), MAPE (Mean absolute percentage error) and RMSE (Root-mean-square deviation) are calculated and shown on this tab.

T-Controller Workshop as the application provides three working modes:

1. Creation and replacing of controller, using user’s linguistic description.
2. Controller creation with table data.
3. Controller usage with real data.

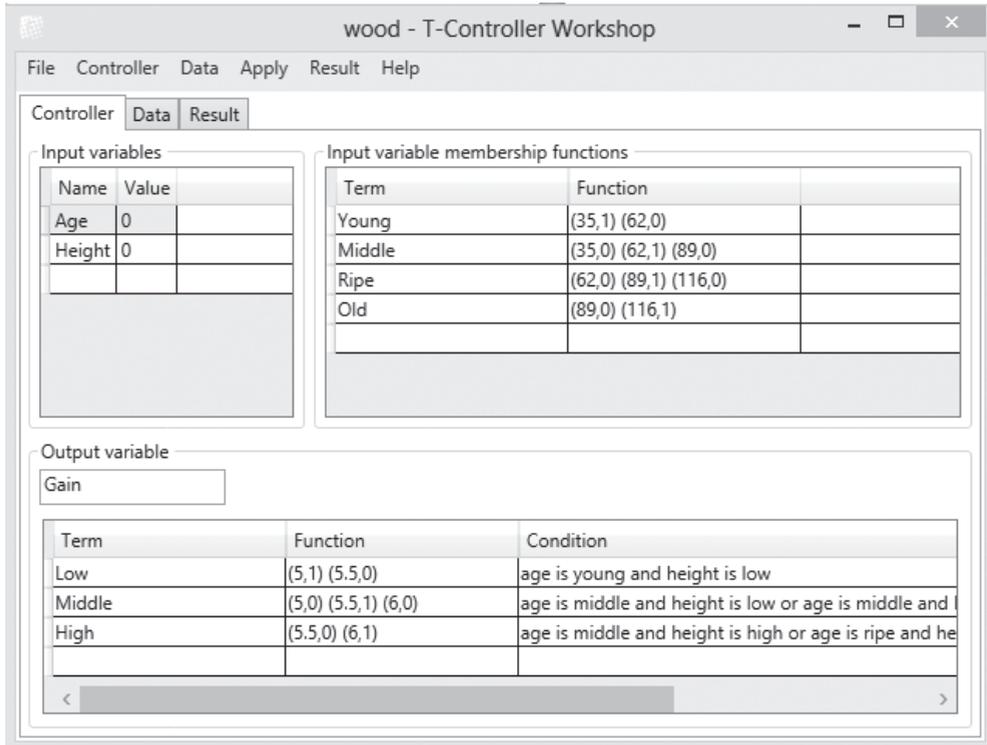


Fig. 3. T-Controller application

5.1. Input variables settings

In order to create controller it is required to set all inputs (input variables), which are used by controller. These values are set in column 'Name' of the table 'Input Variables'. New inputs can be added with mouse double click at an empty table row.

Input values used when controller is applied in a single predict mode and should be filled in.

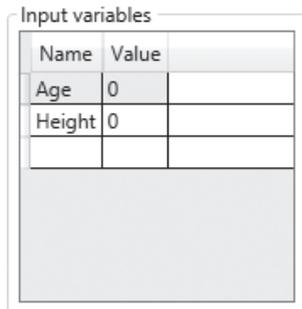


Fig. 4. Controller input settings

The next step is to assign a membership function for each input. In order to do this appropriate name from the table 'Input Variables' should be selected. Linguistic variable names for this input and appropriate membership functions are shown on the Fig. 4 and Fig. 5.

To define input variables membership function it's required to set 2 parameters:

1. Linguistic variable that defines this function in column 'Label'.
2. Coordinates of polyline, which describes this membership function. The coordinates are assigned in text format as a set of tuples and are set in the column 'Function'.

Input variables			Input variable membership functions		
Name	Value		Term	Function	
Age	0		Young	(35,1) (62,0)	
Height	0		Middle	(35,0) (62,1) (89,0)	
			Ripe	(62,0) (89,1) (116,0)	
			Old	(89,0) (116,1)	

Fig. 5. Controller input and appropriate membership functions

5.2. Output variables settings

T-Controller has one output variable by definition. It is described by name, list of membership functions and by the condition, how this point belongs to output variable ranges. These conditions is also called as fuzzy logic rules because they are based only on linguistic terms, which describe input variables.

5.3. Fuzzy logic rules construction

Conditions and rules of fuzzy logic is set as a text in column 'Condition' and row which relates to appropriate output variable range. Fuzzy logic rule consists with assertions in following form:

Assertion: = <input name> is <linguistic variable of this input>

Assertions are united between themselves with conjunction by operator 'AND'.

Conjunction: = <assertion> and <assertion>

Conjunctions are united between themselves in conditions (disjunctions) by operator 'OR'.

Condition: = <conjunction> or <conjunction>

Therefore, each linguistic range of appropriate output variable has just one dependency function and one condition, which defines entering to this range, Fig. 6.

5.4. Data loading

T-Controller Workshop works with data in text format CSV. It's assumed that first row in data file contains table column names, which is stored in this file.

Output variable

Gain

Term	Function	Condition
Low	(5,1) (5,5,0)	age is young and height is low
Middle	(5,0) (5,5,1) (6,0)	age is middle and height is low or age is middle and
High	(5,5,0) (6,1)	age is middle and height is high or age is ripe and he

Fig. 6. Output parameters

Data set also can contain skipped values. Such values are marked as ‘NaN’ and controller will work with not completed data set. It’s shown in following Fig. 7.

wood_improved - T-Controller Workshop

File Controller Data Apply Result Help

Controller Data Result

Age	Height	Gain
NaN	73	41
30	107	48
40	14	52
50	171	57
60	198	61
70	NaN	63
80	238	64
90	253	64
100	268	64
110	28	63
120	29	61
130	2962	58
NaN	99	56

Fig. 7. Loading incomplete data set

5.5. T-Controller advantages

T-Controller has such advantages over traditional FIS:

- The logic inference (and) and the composition (or) are combined into one specific step;

- The number of rules conditioned by features only output variables;
- Fast defuzzification;
- The rules designing procedure is intuitively understood for experts via analysis of possible situations for output variable. *T*-Controller uses the following rule form: R: IF X IS A AND X IS B THEN Y IS C, where R is the rule number, X is the input number, A, B and C are linguistic variables, keyword IS marks a clause, keyword AND marks a conjunction;
- High accuracy of *T*-Controller – high-speed geometrical defuzzification method with zero systematic error (much strict «input» gives much precise «output»);
- The procedure of configuration is faster;
- Simplicity of implementation in both software and hardware version.

6. Geometrical transformation model

The GTP is totally a new concept of the information objects modeling that are presented either tabular data or production rules of the logical conclusions and their combinations. The GTM is intended for applying in the ‘black box’ or ‘gray box’ mode and provided an effective solution for a wide range of problems such as classification with and without the supervisor modes, the time series prediction and forecasting, the main components analysis and factor analysis, the partially lost data or their consolidation recovering, the information protecting and privacy methods implementing and solving the systems of the linear algebraic equations with not defined and redefined systems and multidimensional data visualization [7].

Geometrical transformation model (GT) is universal approximator that realizes training and self-training principles. It is based on the space and time paralleling realized both by hardware and software manners. GT is assigned to remove or reduce negative properties of existing means of information modeling, such as regressive and inductive models, artificial neural networks, usual fuzzy logic controllers and static procedures [9].

Geometrical transformation model has the following properties:

- Quick feed-forward training for previously set number of computational steps that allows both to solve multi-dimension tasks and guarantee the reproducibility of training results;
- Possibility to receive satisfactory solutions on low dimensional training data;
- Possibility to solve tasks in unattended mode;
- Possibility to analyze of inner data structure;
- High precision and improved generalizing properties;
- Effective solution of tasks of computing mathematics (solving of equation systems);
- Simple variants of solution of fuzzy logic and neural network tasks [3].

6.1. Basic suppositions of geometrical transformation model

Information models are based on the use of the available experimental database by definition. Let’s abstract from the obvious problems which may occur during its creation, including the presence of hidden parameters, noises, the ambiguity of data, contradictory of certain dimensions, different density and probability of representation of data in certain planes of the domain. Let’s assign the above features to the factors that increase the general error of the modeling. Influence of these features may be minimized by the usage of the additional

methods for increasing the accuracy. In other words, consider that the modeling object functions are sufficiently represented by the set of the vectors of realizations (observations), generally – cardinality of the continuum. The separation of vector on input and output parameters (attributes) is not essential for the GTM. Each of the vectors is displayed as a point of of the multidimensional space of realizations. There intuitively exist space-similar relations between the elements of this space such as: the concept of the norm or distance from the point to the origin, as well as the metrics for establishing proximity between the implementations. The selected structure of the space realizations determines the way of the comparison and possible transformations of implementations.

Taking into consideration the option the Euclidean space realizations as a linear space over a number field for which there exists a scalar product operator in this field. Note that there exist as the relationship between the individual parameters of vector realizations as relationships between realizations. Then points of the observations of the modeling object are forming his body (hyperbody). This body will have a definite dimension and form depending on the characteristics of the data. Hyper body is characterized by the compactness property where close realizations correspond to the points which are located close in space.

In the absence of linkages, modeling objects body becomes amorphous form and operations of simulation lose all practical meaning. The concept of GTM provides an approximation of certain geometric transformation of the set of available points of modeling objects body implementation, which representing in the aggregate state sample for training and testing. The above transformations that are performed both in training and applying the model, implement specified functions of GTM – consolidation by presenting the body with minimal loss in the coordinate system of reduced dimensions, analysis of PCA and FA by constructing an intermediate coordinate system, orthogonal directions along the last match of the directions of maximum variance, finding of the unknown coordinates of points based on the known projection onto the plane of inputs etc. As a basic geometric transformations of the body object modeling points elected sequential construction of normal (up to a given vector) space (hyper-space) and projection of implementation points onto constructed normal sapce. For these transformations applied formulas expansion for 3D analytic geometry on a space of arbitrary dimension.

7. Crane management system based on *T*-Controller

Industrial cranes are used for transportation of load within a given trajectory. Usually a skilful operator manages this task. During this process, the load might swing in a pendulum-like motion. If the swing exceeds a proper limit, it must be reduced or the operation must be interrupted until the swing stops. One of the ways to improve control efficiency of such kind of process is based on the using of human experience and the fuzzy modeling [8].

The swing manage is realized by controlling of the two main input parameters: the angle relative to the vertical axis of the cart and cart position relative to the start/end point, and by using *T*-Controller software. *T*-Controller is the FIS that uses fuzzy logic rules for an adequate assessment of the power that supplied to the input of the crane. Rules are formed on the formalization of knowledge and professional experience of the experts of certain industry. The generated data by Euler and Runge-Kutta methods are submitted on the Fig. 8.

Time elapsed: 7984 ms.

N	Angle	Distance	Power
0	0	0	6
0	0	0	6
1	-0.001287167544	0.000558657216	5.99600959131429
2	-0.00374367699698	0.00162515160645438	5.98839177423961
3	-0.00725952088386736	0.00315279546206902	5.97748003241379
4	-0.0117319045033526	0.00509870226645418	5.96358069809676
5	-0.0170646915216368	0.00742348220186356	5.94697512712955
6	-0.0231678964122008	0.0100909621528354	5.92792169890832
7	-0.0299572199224076	0.0130679282334051	5.90665765547568
8	-0.0373536240596184	0.0163238890225039	5.8834007926964
9	-0.0452829433668231	0.01983085783831	5.85835101544065
10	-0.0536755295152099	0.0235631525167134	5.83169176773776

Time elapsed: 342 ms.

N	Angle	Distance	Power
0	0	0	6
0	-0.00590929667474927	0.00256543734952061	5.98167544750343
1	-0.0322818244700846	0.0140525987136102	5.89962429490279
2	-0.0709690221798923	0.0311318593293906	5.77762957621864
3	-0.11591171545171	0.0515636106033738	5.63168849569019
4	-0.162521824132765	0.0738491602033111	5.47250599854778
5	-0.207276461887347	0.0969939619150904	5.30718586632078
6	-0.247454150870616	0.120349910342001	5.14037921184285
7	-0.280965212393645	0.143491128589814	4.97506336721562
8	-0.30624359194	0.166169603685165	4.81307425939168
9	-0.322177741551503	0.188234398131931	4.65546858477192
10	-0.328065276966059	0.209611809143588	4.50277279183152

Fig. 8. Generated data by Euler and Runge-Kutta methods

7.1. Linguistic control strategy

On the other hand, the human operator is able to operate a crane without differential equations. The operator did not use the cable lengths sensor. The operator lifts the cart by means of a crane, applies middle power to the crane, to see how the cart is swinging. Depending on the response, he adjusts the engine power to keep the cart a little behind the cranes 'head'. In this position, the maximum speed can be achieved with minimal impact. Toward to the end position, the operator reduces engine power. Thus, the cart gets a little ahead of the cranes 'head' and cart almost reaches the desired location. Then increase engine power so that the 'head' of the crane is above the object while swinging is close to zero. In this case there is no differential equations required to implement this, and all sorts of non-linearity and offsets are eliminated by operators observations for the position of the cart.

Analysis of operator's actions show that the operator uses some 'rules of thumb' to describe his management strategy:

- Start with a medium power.
- If you're still far from the destination point, then adjust the engine power so that the cart has become a little more far for the 'head' of the crane.
- If you are close to the destination, it is necessary to reduce the speed so that cart got a little ahead of the cranes 'head'.
- When the cart is very close to the destination, provide the minimum engine power.
- When the cart has reached its destination and swinging is zero, turn off the engine.

7.2. Implementation of linguistic control strategy

To automate the crane control sensors to determine the position of the cart 'Distance' and its inclination angle 'Angle' are used. Using these inputs to describe the current state of the crane, six rules that can be translated to 'if-then' format are defined:

1. IF Distance = far AND Angle = zero, THEN Power = positive medium.
2. IF Distance = far AND Angle = negative low, THEN Power = positive big.
3. IF Distance = far AND Angle = negative big, THEN Power = positive medium.
4. IF Distance = medium AND Angle = negative low, THEN Power = negative medium.
5. IF Distance = small AND Angle = positive low, THEN Power = negative medium.
6. IF Distance = zero AND Angle = zero, THEN Power = zero.

'If-then' rules always describe the reaction to certain situations: if <situation> then <action>.

In the case of a crane control system, each situation is determined by two conditions. The first term describes the distance value, the second the angle value. Terms are combined by 'AND', which indicates that both conditions must necessarily be carried out for that situation.

8. Conclusion

Today, fuzzy logic items can be found in many industrial products such as electric trains control systems or military helicopters appliances. Active fuzzy logic consumers are the bankers and financiers, as well as experts in the field of political and economic analysis, which everyday tasks require to make the right decisions in difficult circumstances of unpredictable market. They use a fuzzy system to create models of various economic, political and stock situations.

The following advantages of fuzzy systems could be outlined:

1. Ability to handle input data set clearly: for example, dynamic problem when data continuously change over time, value that's impossible to specify uniquely as results of statistical surveys or advertising campaigns;
2. Ability to formalize fuzzy evaluation criteria and comparison: operating 'most', 'possible', 'preferably' and etc. criteria;
3. Ability to conduct quality evaluations as input data and output results: data values and their degree of certainty and its distribution;
4. The possibility of fast simulation of complex dynamic systems and their comparative analysis with a given degree of accuracy, in terms of the system behavior principles described by fuzzy-methods: quickly find the exact values of the variables and make rules to describe them and evaluate various options for output values.

Logistics systems give a possibility to use the experience and the results of experiments to provide much effective solutions for many widespread problems. They do not replace nor compete with traditional control methods. Fuzzy logic extends the ways of implementation of automated control methods that are used in applications and adds the ability to use observations in system management. A simple example is a gantry and bridge cranes that use fuzzy inference system can provide a clear and simple solution to the problem that much more difficult to be solved using traditional management methods.

The use of automated systems for solving everyday tasks enables us to save our time and money. The use of automated control systems to control crane cart swinging avoids the company from unexpected losses that may occur if the crane operator exhausted or sleepy and cannot notice the changes in the system. Ensuring the quality and speed of load transportation saves the company money and raises its rating among others. Therefore, the use of automated management systems in industry is entirely appropriate.

In this paper the automated control system based on the fuzzy logic is presented. The crane management system control cart swings with high accuracy positioning during movement. The simplified model of the crane is represented as a system with one input parameter u (the power of crane) and two output: a (angle of the cart) and x_c (cart position). New fuzzy

inference system *T-Controller* based on geometrical transformation model is used to realize the crane management system. Geometrical transformation model is briefly described and its main properties are listed.

References

- [1] Omar H.M., *Control of Gantry and Tower Cranes*, Dissertation submitted to the Faculty of the Virginia Polytechnic Institute, Virginia 2003, 100.
- [2] *About T-Controller*, Digital source, access by link: <http://tkatchenko.com/t-controller/about-t-controller/>
- [3] Tkachenko O., *Rule-based system of improved accuracy*, Materials of 56th International scientific colloquium, Ilmenau: TU Ilmenau, 2011.
- [4] Tkachenko R., *The new paradigm of the artificial neural networks straightforward dissemination*, Visnyk of the Lviv Polytechnic National University, Computer Engineering and Information Technology, No. 386, 1999, 43-54.
- [5] Burul I., Kolonić F., Matuško J., *The control system design of a gantry crane based on H_{∞} control theory*, MIPRO, Croatia 2010, 183-188.
- [6] Popadic T., Kolonic F., Poljungan A., *A fuzzy control scheme for the gantry crane position and load swing control*, University of Zagreb, 6.
- [7] Tkachenko R., Tkachenko P., Tkachenko O., Schmitz J., *Geometrical data modelling*, Intelligent systems of making decisions and applied aspects of information technology, Proceedings of the conference, Vol. 2, Eupatoria, 2006, 279-285.
- [8] Verbenko I., Tkachenko R., *Fuzzy Methods and Tools for Crane Management System Based on T-Controller*, Journal of Global Research in Computer Science, Vol. 4, No. 3, March 2013, 1-4.
- [9] Tkachenko R., *Accelerated learning of multilayered neural networks on the base of the new paradigm*, Third Conference 'Neural networks and their applications', Kule 1997, 129-130.
- [10] Licata G., *Fuzzy Logic, Knowledge and Natural Language, Fuzzy Inference System – Theory and Applications*, Dr. Mohammad Fazle Azeem (Ed.), InTech, 2012, 504.
- [11] Dadios E.P., *Fuzzy Logic – Controls, Concepts, Theories and Applications*, InTech, 2012, 428.
- [12] Grigorie L., *Fuzzy Controllers, Theory and Applications*, InTech, 2011, 368.
- [13] Azar A.t., *Fuzzy Systems*, InTech, 2012, 216.
- [14] Yung C. Shin, Chengying Xu., *Intelligent systems: modeling, optimization, and control*, CRC Press Taylor & Francis Group, 2009, 456.
- [15] Leondes C.T., *Fuzzy Logic and Expert Systems Applications*, Academic Press, 1998, 417.
- [16] Kalogirou S., *Artificial intelligence in energy and renewable energy systems*, Nova Science Publishers, 2007, 473.
- [17] Koleshko V.M., *Intelligent Systems*, InTech, 2012, 366.
- [18] Kovacic Z., Bogdan S., *Fuzzy controller design: theory and applications*, CRC Press Taylor & Francis Group, 2006, 416.

- [19] Nguyen H.T., *A first course in fuzzy and neural control*, Chapman & Hall/CRC, 2003, 312.
- [20] Baaklini N., Mamdani E.H., *Prescriptive methods for deriving control policy in a fuzzy-logic controller*, Electron. Lett., Vol. 11, 1975, 625-626.
- [21] Bernard J.A., *Use of rule-based system for process control*, IEEE Contr. Syst. Mag., Vol. 8, No. 5, 1988, 3-13.
- [22] E.H. Mamdani, *Applications of fuzzy algorithms for simple dynamic plant*, Proc. IEE, Vol. 121, No. 12, 1974, 1585-1588.
- [23] Zadeh L.A., *Outline of a new approach to the analysis complex systems and decision processes*, IEEE Trans. Syst. Man Cybern., Vol. SMC-3, 1973, 28-44.
- [24] Bridging the Gap Between Conventional and Intelligent Control, IEEE Control Systems Magazine, Vol. 13, No. 3, 1993, 12-18.
- [25] Jantzen J., *A Tutorial On Adaptive Fuzzy Control*, EUNITE, Vol. 2002, 2012, 709-719.

GRZEGORZ NOWAKOWSKI*

EFFECTIVE USE OF LAMBDA EXPRESSIONS IN THE NEW C++11 STANDARD

EFEKTYWNE WYKORZYSTANIE WYRAŻEŃ LAMBDA W NOWYM STANDARDZIE C++11

Abstract

In this paper, the possibility of lambda expressions and methods for their effective use in C++ code in the new C++11 standard have been presented. Studies that compare the execution times of the program and use lambda expressions as well as classical methods have been conducted. The results confirm the effectiveness of lambda expressions with respect to traditional methods.

Keywords: lambda expressions, lambda – introducer, lambda – declarator, compound – statement, function object, STL, the new C++11 standard

Streszczenie

W artykule przedstawiono możliwości wyrażeń lambda oraz zaprezentowano metody ich efektywnego wykorzystania w kodzie języka C++ w nowym standardzie C++11. Przeprowadzono badania porównujące czasy wykonania programu z zastosowaniem wyrażeń lambda i metod klasycznych. Uzyskane wyniki potwierdzają większą efektywność wyrażeń lambda w stosunku do tradycyjnych metod.

Słowa kluczowe: wyrażenia lambda, lambda – introducer, lambda – declarator, compound – statement, obiekt funkcyjny, STL, nowy standard C++11

* M.Sc. Grzegorz Nowakowski, e-mail: gnowakowski@pk.edu.pl, Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

1. Introduction

Each programmer using C++ language standard library is aware of the need to write short functions – predicates, which control algorithms. As a result, the code is being filled with short classes that perform simple operations. Simultaneously, the separation of implementation from the caller significantly impedes the modification and maintenance of the code. One exemplary solution to this problem is the use of lambda expressions which allows for the formation of anonymous function objects defined in the caller. Until the advent of the new C++11 standard, lambda expressions had not been supported by the language itself. The programmer had to refer to external libraries. In this paper, the possibility of lambda expressions and methods of their effective use in C++ code have been presented. Studies that compare the execution times of the program by using both lambda expressions and classical methods have been conducted. The results confirm the effectiveness of lambda expressions with respect to traditional methods. The suggested solutions significantly improve the readability of the code, as well as allowing for the disobeying of namespace classes and functions which are used in the code only once.

2. Theoretical basis

Lambda calculus is a formal system [2] that allows for the defining of functions and for providing them with arguments. The system was introduced by Alonzo Church and Stephen Cole Kleene in 1930. Initially, it was the alternative to the theoretical approach to the fundamentals of mathematics in which the function (intentionally understood as a definition) was the primary concept. Although this project was not successful, it soon became apparent that the lambda calculus is a useful apparatus in the theory of computation. It is also complete in the sense of Turing [4]. It means that all algorithms which are capable of being enrolled in the lambda calculus can be implemented on a Turing machine and vice versa. Later, lambda calculus [2] has proved to be an indispensable tool in the theory of programming languages as a consequence of information technology development, and in practice, became the inspiration for functional programming languages such as LISP.

We take into consideration objects called *lambda-term* [2] in lambda calculus. Traditionally, they are defined as formal expressions of a certain language like regular expressions in algebra and first-order logic. The essential difference is that the lambda-expressions in a remaining equivalence relation (alpha-conversion) are identified with each other (considered as identical) and thus, they are not expressions but the abstraction classes of lambda-expressions (also known as pre-terms).

Let us assume that there is a countable, infinite set of subject variables [1–4]:

- subject variables are *lambda-expressions*,
- if M and N are *lambda-expressions* then (MN) is a *lambda-expression*,
- if M is a *lambda-expression* and x is variable then $(\lambda x.M)$ is a *lambda-expression*.

The expression of the form (MN) is called an **application** [3] who passes (applied) argument N on to the function M . In contrast, the expression of the form $(\lambda x.M)$ is a **functional abstraction** [3] representing an intuitive anonymous function that takes one argument and passes it on to M .

Listing 1. Examples of lambda expressions

1:	$\lambda x.x$ (<i>identity</i>)
2:	$\lambda x.y$ (<i>constant expression</i>)
3:	$\lambda f.\lambda x.x$

Listing 2. Exemplary applications of function to argument

1:	$(\lambda x.xy)xz \rightarrow zyz$ (<i>insert 'z' at the place of all x (x is bound by λ) in the expression XYX)</i>)
2:	$(\lambda a.xy)xz \rightarrow xyx$ (<i>no change, the variable 'a' is not present in the expression</i>)

3. Lambda expressions in C++11 and methods of their effective use in the code

Lambda calculus introduced in the new standard C++11 is very similar to the assumptions presented in the previous section – theoretical basis. The notation [7] is slightly different due to the lack of the symbol λ on most keyboards and some extensions.

In order to start working with lambda expressions or some random mechanism of the new standard C++11, we should use the Microsoft Visual Studio 2012 [5], or GCC version at least 4.6 [6] (on the command line using the '`-std = c ++11`').

In listing 3, some examples of lambda expressions are indicated.

Listing 3. Simple lambda expressions in C ++11

1:	<code>[] (int x) -> int { return x; };</code>
2:	<code>[] (int x, int y) -> int { return x+y; };</code>
3:	<code>[] (int x, int y) -> int { int z = x+y; return z; };</code>

It is easy to notice that none of the examples match the definition presented in the previous section. There is the identity in the first line that has the specified type *int*, which in turn limits its use only to integers. The original expression could be applied to anything and also to itself. The second and the third lines contain a plus sign, which is not on the approved list of symbols. Additionally, the expression in the third line defines a temporary variable *z*.

Lambda expressions in C++11, in comparison with those of a mathematical lambda calculus, except those indicated in the above listing, can perform any instructions associated with external functions and create objects or use external variables. They also constitute a great convenience for programmers especially in the context of the use of STL algorithms [8]. The following figures show lambda expression grammar in C ++11.

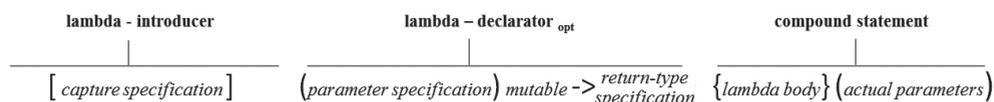


Fig. 1. Lambda expression grammar

Listing 5. Passing parameters by value

```

1: function< int ( ) > fvalue;
2: {
3:     int x=0;
4:     fvalue = [x] { return x+1; };
5: }
6: cout<< fvalue();

```

Listing 5 (which uses a template *function* with a header *functional*) illustrates the fact that when the value of the expression is induced in the sixth line, variable *x* – no longer exists despite the program working correctly. It results from the fact that a copy of the variable *x* is located in a lambda expression. It should be noted that the variables captured by value cannot be modified. It may seem inconsistent with the classical function call where any operations can be made on the copy of received variables. However, if we look at it in terms of a constant object, there seems to be a logical explanation.

However, in case there is a need to modify the captured variables, they should include the keyword *mutable* [8–9] which causes the lambda expression not to be treated as constant, and thus allows such a modification (listing 6).

Listing 6. Use the keyword mutable

```

1: function< int ( ) > fvalue;
2: {
3:     int x=0;
4:     fvalue = [x] ( ) mutable { return ++x; };
5: }
6: cout<< fvalue() << “ ” << fvalue();

```

As it has been mentioned above, in addition to passing parameters by value, there is also passing it by reference [7]. Both types of capture can be combined (listing 7). However, note that if there is capturing by value we have a copy of the object whereas if there is capturing by reference we do not have one. Passing by reference works here more quickly. Yet, do not apply it to a non-existing object because it will result in a runtime error.

Listing 7. Passing parameters by value and references

```

1: int x=1, y=1, z=1;
2: // all variables are passed by reference
   [&] ( ) { cout << x << y << z; }();
3: // z passed by reference, the rest of the variables by value
   [=,&z] ( ) { z = x + y; }();
4: // x passed by value, the rest of the variables by reference
   [&,x] ( ) { y = x; z = x; }();

```

3.2. Lambda-declarator

The second component of the lambda expression [8] is a *lambda-declarator* (Fig. 3), wherein it should be noted that it is optional, and the correct lambda expression cannot include it at all.

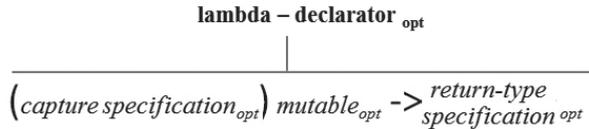


Fig. 3. Lambda-declarator

Lambda expression [12] as well as the ordinary functions can have arguments which are passed on to the function at runtime. It should be remembered that the same rules are applied here as for the function arguments. If the expression consists of more than a single return statement, it must specify the type of the return value. Listing 8 shows examples of declarations of arguments with the unauthorized case (in comments).

Listing 8. Declaration of the list of arguments

1:	<i>// expression takes two parameters by value</i> <i>// specify the return type is not required</i> [] (int x, int y) {return x + y; } (1,2);
2	<i>// compile error</i> <i>// trying to pass x variable rvalue by not constant reference</i> <i>// [] (int &x, int y) {return x + y; } (1,2);</i>
3:	<i>// expression takes one parameter as a constant reference, the second by value</i> <i>// specify the return type is required</i> [] (const int &x, int y) -> int {int z = 3; return x + y + z; } (1,2);
4:	<i>// returning a reference to a temporary variable</i> [] (int x, int y) -> int& {int z = 0; return z; } (1,2);

When we need a lambda expression to apply only once, we can call a lambda function in the place of the definition (assignment to the named variable is unnecessary). The use of a complicated variable declaration by means of the *function* can be replaced by the application of *auto* type [8–9]. It allows for the use of the expression repeatedly without the necessity to define the expression each time. The following considerations have been presented in listing 9.

Listing 10 shows the use of a lambda expression to solve the problem of sorting a (descending) vector that contains elements of an integer. For comparison, the implementation of the same problem in the classical approach has been presented. A programmer who does not know lambda functions would have to write their own sorting function or use the standard library *sort* algorithm. However, in the case of using the *sort* function, they would have to

create a class with overloaded *operator ()*, whose object could perform a comparison. Such an approach would require writing a few lines of code and naming the function for a single use. Additionally there would be a separation here from the implementation of the caller, which would significantly hamper modification and maintenance of a code. It is difficult to disagree with the notion that the use of lambda functions is much simpler and clearer.

Listing 9. Calling the lambda

```

1: // assign a lambda function in place of the definition of the named object
   function< void ( int ) > f_1 = [ ] ( int x ) { cout << "Example 1" <<x; };
2: // assign a lambda function in place of the definition of the named object
   // use of 'auto' (the compiler itself in this case will determine the type of the
   // variable)
   auto f_2 = [ ] ( int x ) { cout << "Example 2" <<x; };
3: // call these functions to named objects
4: f_1(1);
   f_2(2);
5: // lambda function call in place of its definition in the unnamed object
   [ ] (int x) { cout << "Example 3" <<x; } (3);

```

Listing 10. Solving the problem of sorting vector by means lambda expressions

```

1: //new standard C++11
   sort(v.begin(), v.end(), [ ] (int x, int y) { return x > y;});
2: // classical approach
   class comparator
   {
   public:
       bool operator() (int x, int y) const
       {
           return x > y;
       }
   };

   sort (v.begin(), v.end(), comparator());

```

The final component [8] of the *lambda-declarator* is the *return type*. In the previous listings it has been indicated that lambda functions can return a value by means of the *return* like ordinary functions. The difference is that the return type has not been declared anywhere. The standard [11] not only allows for defining the type explicitly, but also imposes an obligation on the compiler automatic determination (only for expressions like *return ...*).

Listing 11 shows the different combinations of returned values by lambda expression with illegal cases (in the comments).

Listing 11. Examples of different combinations of returned values by lambda expression

```

1: // explicitly provide a return type
   int p1 = [] (int x, int y) -> int {return x+y;}(1,2);

2: // no return type – compiler itself determines the type of
   int p2 = [] (int x, int y) -> {return x+y;}(1,2);

3: // abuse (but works by using gcc 4.6)
   // compiler is required to determine the type returned only for expressions like // return ...
   //int p3 = [] (int x, int y) -> {if (x%2) return x; else return y;}(1,2);

4: // compile error - there is a mismatch type declared and returned
   //int p4 = [] () -> int {return false; }();

   // compile error - it is impossible to determine the return type (such as
5: // inconsistent with the specification)
   //int p5 = [] (int x) { if x > 5 return x; else return true;}(1);

```

3.3. Compound-statement

The third component of the lambda expression [8] is a *compound-statement* (Fig. 4). Body expressions can contain anything that the contents of an ordinary method or function would contain.

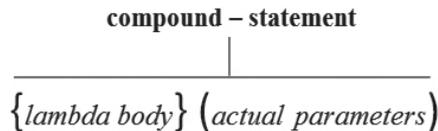


Fig. 4. Compound-statement

4. Practical applicability: function pointers, functors and lambda expressions

The example has been investigated, on the basis of which three approaches of passing information to the STL algorithm have been illustrated: through function pointers, functors and lambdas [8]. A list of random integers has been generated and it has been examined how many of them are divisible by 5 and how many of them are divisible by 7. Generating a sequence of values is based on the use of *vector* <int> array to hold the numbers on the use of the STL *generate* () algorithm to stock the array with random numbers.

The *generate*() function takes a range, specified by the first two arguments, and sets each element to the value returned by the third argument that is a function object taking no arguments. In the described example, the function object is a pointer to the standard *rand*() function.

Listing 12

```

1: vector<int> n(1000);
2: generate(vector.begin(), vector.end(), std::rand);

```

4.1. Passing on information to the STL algorithm by function pointer

In order to count the number of elements divisible by 5 or 7, *count_if()* algorithm has been applied. The first two arguments should specify the range just as in the case of *generate()*. The third argument should be a function object that returns true or false. Thus, the *count_if()* counts all the elements for which the function object returns true. To find elements divisible by 5 or 7, these function definitions have been used:

Listing 13

```

1: bool fcount_5(int x) {return x % 5 == 0;}
2: bool fcount_7(int x) {return x % 7 == 0;}

```

By means of all above definitions, elements have been counted in the following way:

Listing 14

```

1: int count_5 = std::count_if(n.begin(),n.end(),fcount_5);
2: cout << "Count of numbers divisible by 5:" << count_5;

3: int count_7 = std::count_if(n.begin(),n.end(),fcount_7);
4: cout << "Count of numbers divisible by 7:" << count_7;

```

4.2. Passing on information to STL algorithm by functors

A similar functionality has been implemented by means of a functor. The functor is a class object that can be used as the name of the function thanks to the class defining *operator()()* as a class method. The advantage of this functor in this example is the fact that one can use the same functor for both counting tasks. The following definition of the functor has been applied:

Listing 15

```

1: class f_count
2: {
3:     private:
4:         int divisor;
5:     public:
6:         f_mod(int d = 1) : divisor (d) {}
7:         bool operator() (int x) {return x % divisor == 0;}
8: };

```

The elements have been counted as follows:

Listing 16

```

1: count_5 = std::count_if(n.begin(),n.end(),f_count(5));
2: cout << "Count of numbers divisible by 5:" << count_5;
3: count_7 = std::count_if(n.begin(),n.end(),f_count(7));
4: cout << "Count of numbers divisible by 7:" << count_7;

```

The argument `f_count(5)` creates an object that stores the value 5, and `count_if()` uses the created object to call the `operator()()` method, setting the parameter `x` equal to an element of `n`. In order to count the numbers which are divisible by 7 instead of by 5, one has to use `f_count(7)` as the third argument.

4.3. Passing on information to STL algorithm by lambda expression

As it has already been mentioned, lambda expressions allow for applying an anonymous function definition (a lambda) as an argument to functions that are expecting a function pointer or functor. The lambda corresponding to the `fcount_5(int)` function has been defined below:

Listing 17

```

1: [ ] (int x) {return x % 5 == 0;}

```

It resembles the definition of `fcount_5(int)` to a great extent:

Listing 18

```

1: bool fcount_5(int x) {return x % 5 == 0;}

```

In fact, the differences consist in replacing the function name with `[]`. Also, there is no declared return type. The return type is the type that `decltype` would define on the basis of the return value, which would be `bool` in this case. If the lambda did not have a return statement, `void` would be the return type. In the example below, lambda expressions have been applied in the following way:

Listing 19

```

1: count_5 = std::count_if(n.begin(), n.end(), [ ] (int x) {return x % 5 == 0;});
2: cout << "Count of numbers divisible by 5:" << count_5;
3: count_7 = std::count_if(n.begin(), n.end(), [ ] (int x) {return x % 7 == 0;});
4: cout << "Count of numbers divisible by 7:" << count_7;

```

It can be easily noticed that the entire lambda expression has been put in the place of a pointer or a functor constructor. The automatic type deduction for lambda works only if the body consists of a single return statement. In other cases one has to explicitly specify the type of the return value by using a new syntax and later by means of type declaration.

5. Studies comparing the execution times of the program by means of both lambda expressions and classical methods

Tests have been conducted on a PC equipped with an Intel Core i5 2.80 GHz and 4 GB of memory. The test program was based on the application of a lambda expression to solve the problem of summing one hundred thousand products of the individual elements of the two-dimensional array of a randomly selected size range of (1, 100.000), compared to solving the same problem in a classical approach by using standard function. The computation does not matter, random numbers have been selected to avoid any compiler optimizations in this area. As a consequence we are exclusively interested in the difference between standard functions and lambda expressions. Different mechanisms for passing parameters have been tested: (1) lack of variables, (2) one variable in explicitly, (3) one variable by reference, (4) two dimensional arrays explicitly. The program marked the execution times of the implemented test.

Table 1

Execution times of the tests

Test number	Lambda expression [ms]	Standard function [ms]
1	34+-1	35+-1
2	33+-1	36+-1
3	35+-1	35+-1
4	35+-1	34+-1

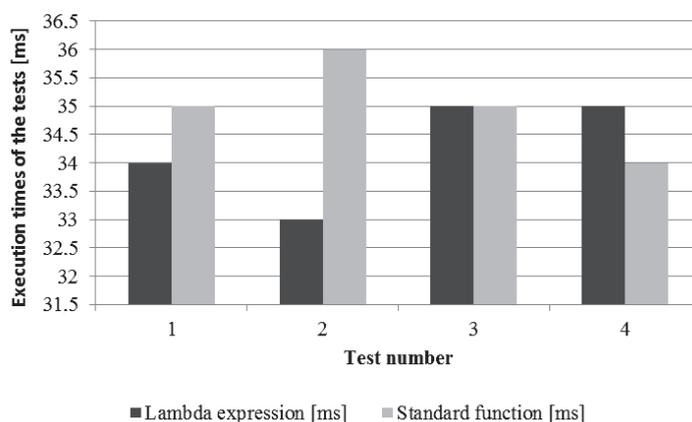


Fig. 5. Execution times of the tests

By analyzing the obtained results of the tests, it was found that both the standard functions and the lambda expression perform in a similar period of time, which means there is no significantly different loss of performance when using lambda. Only in the case of the fourth test were better results obtained by standard functions (Fig. 5). However, taking into account all the possibilities of these expressions mentioned in the article, the obtained results confirm better efficiency of lambda compared to traditional methods.

6. Conclusions

In this paper, the possibility of lambda expressions and methods for their effective use in C++ code in the new C++11 standard have been presented. For this purpose, four qualities have been taken into account: proximity, brevity, efficiency and capability as well as the lambda expression have been juxtaposed with functions and function objects (functors).

Many programmers claim that it is useful to place definitions close to where they are used. This practice allows them to avoid searching for multiple pages of source code to find the definition used in a particular place. It also facilitates modifying the code because the components are immediately modified within the definition. From this standpoint, a lambda expression is the most effective because the definition is at the point of usage. Functions are the least efficient because they cannot be defined inside other functions, so the definition will be located possibly quite far from the point of usage. Function objects in this regard are more efficient than functions, because a class (including a function object class) can be defined inside a function so that the definition of the functor can be located close to the point of use.

In terms of brevity, the function object code is much more extensive than the equivalent function or lambda expression. Functions and lambdas are approximately equally brief. The exception is when we want to use a lambda expression more than once. However, the repeated use of lambda expressions is not really necessary. Although it is admittedly an anonymous function, it can be associated with the name (using the type of *auto*), and then this name may be used as often as desired. In contrast to the ordinary function, a lambda expression can be defined inside a function.

While comparing execution times of the program by means of both lambda expressions and classical methods, lambda expressions proved to be more efficient in comparison with traditional methods.

As far as the issue of the possibility is concerned, lambda expressions offer some additional capabilities. In particular, a lambda can access by name any automatic variable in scope. Variables to be used are captured by having their names listed within brackets. If just the name is used, as in `[x]`, the variable is accessed by value. If the name is preceded by an `&`, as in `[&x]`, the variable is accessed by reference. Using `[&]` provides access to all the automatic variables by reference, and `[=]` provides access to all automatic variables by value. For instance, `[x, &y]` would provide access to `x` by value and `y` by reference, `[&, x]` would provide access to `x` by value and to all other automatic variables by reference, and `[=, &y]` would provide access by reference to `y` and by value to the remaining automatic variables.

The main motivation for adding lambdas to C++ was to enable using a function – like expression as an argument to a function that is expecting a function pointer or functor as an argument. So the typical lambda expression is a test expression or comparison expression that can be written as a single return statement. This keeps the lambda function short and easy to understand and enables the automatic deduction of the return value. However, it is likely that a subset of the ingenious C++ programming community will develop other uses.

References

- [1] Hankin Ch., *Lambda Calculi. A Guide for Computer Scientists*, Clarendon Press, 1994.
- [2] Urzyczyn P., *Rachunek lambda*, wykład monograficzny, (online) homepage: <http://www.mimuw.edu.pl/~urzy/Lambda/erlambda.pdf>, (date of access: 2014-04-01).
- [3] Barendregt H., Dekkers W., Statman R., *Lambda calculus with types*, Cambridge University Press, 2013.
- [4] Barendregt H.P., Manzonetto G., *Turing's contributions to lambda calculus*, 2013.
- [5] Microsoft Visual Studio Professional 2012, (online) homepage: <http://www.microsoft.com/en-us/download/details.aspx?id=30682> (date of access: 2014-04-01).
- [6] GCC 4.6 Release Series, (online) homepage: <http://gcc.gnu.org/gcc-4.6/> (date of access: 2014-04-01).
- [7] Siddhartha R., *Sams Teach Yourself C++ in One Hour a Day (7th Edition)*, Pearson Education, 2012.
- [8] Prata S., *C++ Primer Plus, Sixth Edition*, Addison Wesley, 2011.
- [9] SDJ, (online) homepage: <http://sdjournal.pl/magazine/1778-zarzadzanie-pamiecia-w-sun-jvm> (date of access: 2014-04-01).
- [10] Lambda Expression Syntax, (online) homepage: <http://msdn.microsoft.com/en-us/library/dd293603.aspx> (date of access: 2014-04-01).
- [11] Stroustrup B., *The C++ Programming Language*, Addison Wesley, 2012.
- [12] Stroustrup B., (online) homepage: <http://www.stroustrup.com/C++11FAQ.html> (date of access: 2014-04-01).

ZBIGNIEW KOKOSIŃSKI*, SŁAWOMIR WÓJCIK**

A COMPARISON OF SW/HW IMPLEMENTATIONS OF STREAM CIPHER ENCODERS

PORÓWNANIE IMPLEMENTACJI PROGRAMOWYCH I SPRZĘTOWYCH SZYFRATORÓW STRUMIENIOWYCH

Abstract

In this paper, a new method of stream encoding and decoding is presented. It is developed on the basis of a derangement generator. Stream cipher D has been compared with other stream ciphers – E0, W7 and Phelix. Encoding and decoding algorithms have been implemented in C++ and VHDL programming languages. FPGA synthesis data has been reported for Spartan 3E and Virtex 4 devices from Xilinx. The hardware solution has been tested on the Digilent Nexys 2 500K board. Subsequently, comparative studies have been conducted for software and hardware coders, taking into account average coding time and average throughput for 16 input data files of different sizes. Conclusions resulting from the research are derived.

Keywords: stream cipher, coder, decoder, coder throughput, FPGA

Streszczenie

W artykule przedstawiono nową metodę strumieniowego szyfrowania i deszyfrowania danych w oparciu o generator nieporządków. Szyfr strumieniowy D został porównany ze znanymi szyframi strumieniowymi E0, W7 i Phelix. Algorytmy kodowania i dekodowania zaimplementowano w językach programowania C++ oraz VHDL. Podano dane dotyczące syntezy urządzeń sprzętowych w układach programowalnych FPGA typu Spartan 3E oraz Virtex 4 firmy Xilinx. Rozwiązania sprzętowe zostały przetestowane na płycie Digilent Nexys 2 500K. W badaniach porównawczych zbudowanych szyfratorów programowych i sprzętowych uwzględniono średni czas szyfrowania oraz średnią przepustowość dla 16 plików danych o różnych rozmiarach. Sformułowano wnioski z przeprowadzonych badań.

Słowa kluczowe: szyfr strumieniowy, szyfrator, deszyfrator, przepustowość szyfratora, FPGA

* Ph.D. Zbigniew Kokosiński, email: zk@pk.edu.pl, Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

** M.Sc. Sławomir Wójcik, Faculty of Electrical and Computer Engineering, Cracow University of Technology (currently with Ericpol, Cracow).

1. Introduction

Data encryption and security is one of the key issues in modern computer and telecommunication systems [4]. A large number of cryptographic systems have been developed having different characteristics. The most popular systems like DES, AES are supported either by government agencies or telecommunication companies while many others are developed and supported by independent private enterprises.

Encryption algorithms usually belongs to one of the two groups: they use block ciphers or stream ciphers. While software encoders are dominating the market, there are also many hardware implementations. Depending on the encryption method, usually one of the two implementations mentioned above is more efficient, e. g. software implementation of LFSR-based encoders is slower than dedicated hardware solution.

Several research results relating to FPGA implementations of stream ciphers were described in [14]. The Stream cipher VMPC (Variably Modified Permutation Composition), was proposed and developed in 2004 by Bartosz Żółtak [18, 19] on the basis of one-way function. It is easy to implement both in software and in hardware. Recent research papers relating to VMPC encryption technology as well as the present software version of VMPCrypt 4 are available at [19]. In 2009, VMPC stream cipher was successfully implemented and tested in FPGA [6].

In this paper, four different stream ciphers, implemented both in hardware and software, are compared. The selected ciphers are:

- E0 – used in wireless data transmission via Bluetooth interface [7];
- W7 – a one-time candidate for a successor of A5 in mobile GSM technology [15];
- Phelix – dedicated for 32-bit platforms, combines encryption with MAC (Message Authentication Code) [16];
- D – a new method, developed recently on the basis of the derangement generation [12].

The software and hardware encoders are characterized by data processing time and throughput computed experimentally.

In the next section, the concept and basic properties of set derangements are explained. In section 3, the D stream cipher is introduced. Section 4 contains a short description of software implementations. FPGA implementations in Xilinx Spartan and Virtex devices are described in section 5. Section 6 brings a comparison of software and hardware encoders. In the last section, some conclusions and remarks are added.

2. Derangements

The D stream cipher introduced in this paper is developed on the basis of a generation of a specific class of n -permutations with no constant points (no 1-cycles) called derangements. Combinatorial properties of derangements are described in depth in [5, 9]. Several methods for the generation of all set derangements sequentially or in a parallel linear array model are published in the literature [1–3, 8, 13].

The representation of partial derangements is derived from a representation of permutations by iterative decomposition of symmetric permutation group S_n into cosets [12]. Some particular properties of derangements are also established.

Now we introduce representations of the considered combinatorial objects by means of integer sequences (codewords) defined as choice functions of indexed families of sets.

Let $\langle A_i \rangle_{i \in I}$ denote an indexed family of sets $A_i \subseteq A$, where: $A = \{1, \dots, n\}$, $I = \{1, \dots, n\}$, $1 \leq n$. Any mapping f which ‘chooses’ one element from each set A_1, \dots, A_n is called a choice function of the family $\langle A_i \rangle_{i \in I}$. If, for every $i \neq j$, a supplementary condition: $a_i \neq a_j$, for $a_i \in A_i$ and $a_j \in A_j$ is satisfied then any choice function $\alpha = \langle a_i \rangle_{i \in I}$ that belongs to the indexed family $\langle A_i \rangle_{i \in I}$ is called n -permutation of the set A .

Let us now define permutations with forbidden positions and derangements [12]. A permutation π of n -element set $A = \{1, \dots, n\}$ with a forbidden position i is the sequence $\langle \pi(1), \pi(2), \dots, \pi(n) \rangle$, where $\pi(i) \neq i$, for some $1 \leq i \leq n$.

Let $\langle P_i \rangle_{i \in I}$ be an indexed family of sets $P_i \subseteq A$, where $P_i = \{1, \dots, i\}$, $1 \leq i \leq n-1$, and $P_n = P_{n-1}$. Any choice function $\alpha = \langle p_i \rangle_{i \in I}$, that belongs to Cartesian product $\times_{i \in I} P_i$ represents a permutation of A with a forbidden position i if and only if:

$$(p(i) \neq i) \vee [(p(i) = i) \Rightarrow \exists j : (i < j \leq n) \wedge p(j) = i] \quad (1)$$

Any n -permutation with n forbidden positions $i \neq \pi(i)$, $1 \leq i \leq n$, is called a derangement $\delta(n)$ with the forbidden set A .

Let $\langle D_i \rangle_{i \in I}$ be an indexed family of sets $D_i \subseteq A$, where $D_i = P_i$, $1 \leq i < n$, and $F = \{f_1, f_2, \dots, f_k\}$, $F \subseteq I = \{1, \dots, n\}$, $0 \leq k \leq n$, be the forbidden set. Any choice function $\delta(n, k) = \langle d_i \rangle_{i \in I}$ that belongs to Cartesian product $\times_{i \in I} D_i$ represents a derangement of A if and only if:

$$\forall d(i) \in F : (d(i) \neq i) \vee [(d(i) = i) \Rightarrow \exists j : (i < j \leq n) \wedge d(j) = i] \quad (2)$$

The new cipher D belongs to a group of derangement ciphers, working on bits or strings. There exist $\frac{n!}{e}$ different derangements of n -element set. The generation algorithms for derangements can be found in [1–3, 8, 12, 13]. For $n = 32$, $D(32) \approx \frac{32!}{e} \approx 10^{35}$.

3. D stream cipher

The new cipher D , proposed by the first author of the article, belongs to a group of derangements ciphers, working on bits or strings. However, encoding scheme on the basis of derangement operation can not provide nontrivial encodings of specific strings like 0 or 1 sequences. Therefore, the generated derangements are processed further with the help of a key stream S_i generated by a linear feedback shift register (LSFR) – see Fig. 1.

The hardware-oriented algorithm for generating set derangements is developed in the parallel counter model augmented by a triangular permutation network and is a modification of the permutation generation algorithm [11].

The triangular permutation network is built of two-state cells (2-permuters) [10, 11]. Each cell requires a separate control signal. The permutation network can perform $n - 1$ transpositions $(P(i), P(k))$, i.e. can produce any n -permutation of its inputs on outputs.

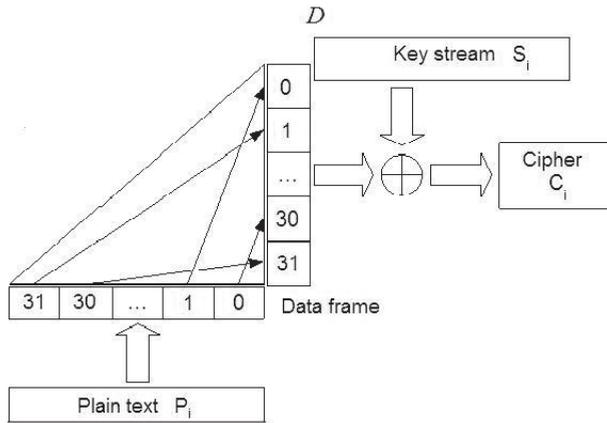


Fig. 1. The idea of D encoding scheme for $n = 32$ with the permutation network and a key stream

The control sequences are produced in $O(1)$ average time per generated object. The output sequences are then obtained from the control sequences in $O(n)$ time.

The control circuit is organized in the following way [17]. With every i -th column of the triangular network ($1 \leq i \leq n$) the i -th ring counter is associated with the initial state from the ‘1-out-of- i ’ code. All column counters form the parallel counter with $n!$ different states. Clock enable signal for the i -th ring counter is a product of carry signals (overflows) from all ring counters preceding it. The state of the permutation network is controlled by $n - 1$ synchronous up-down counters (UDC), where $UDC(i)$ counts mod $(i + 1)$ depending on the cipher bit $C(i)$. For $C(i) = 1$, $UDC(i)$ counts up, otherwise $UDC(i)$ counts down. We assume, that $k = UDC(i)$.

The asynchronous setup of each ring counter and global reset for all ring counters is provided. If the j -th bit of the i -th ring counter $b_i^j = 1$, for $1 \leq j \leq (i - 1)$, then in the i -th column of the network only one cell denoted by $C[i, j]$ is activated to perform the corresponding transposition τ_i^j . If $b_i^i = 1$, $1 \leq i \leq n$, then all cells in the i -th column are in the ‘identity’ state.

After setting the initial state of the network, the control circuit generates consecutive states of network in a constant time (one clock period) and the permutation network generates subsequent configurations representing permutations. In order to recognize a derangement permutation, an additional logic based on formula (2) is needed [12], and, on average, $(e - 1)$ extra clock periods are required to find such a permutation.

Valid n -derangements are detected by a logic function V checking if the condition given in (2) is satisfied:

$$V = \prod_{i=1}^{n-1} FV(i)v_i = \prod_{i=1}^{n-1} FV(i) \left(\overline{b_i^i} + \sum_{j=i+1}^n b_j^i \right) \tag{3}$$

where:

FV – a binary forbidden set vector: $FV(i) = 1$ iff position i is forbidden, otherwise $FV(i) = 0$;

v_i – the function detecting if the condition (5) for the forbidden position i is satisfied; in fact, for technical reasons, v_i should be rewritten in the form:

$$v_i = (\overline{b_i} + ((\dots(b_{i+1}^i + b_{i+2}^i) + \dots) + b_{n-1}^i) + b_n^i) \quad (4)$$

The above logic functions can be computed in $O(n)$ time which matches the network propagation delay. Because the size of the network is limited and the constant factor hidden in the function $O(n)$ is very low, for most applications we may assume that consecutive network configurations are generated in constant time.

The hardware complexity of the generator is $O(n^2)$, and the network propagation delay is $O(n)$. For practical applications, the networks size is limited and the propagation delay can be considered constant.

4. Software implementations

An application in C++ has been developed in MS Visual Studio 2005 for MS Windows platforms with MS .NET Framework 2.0 installed. The compiler has been set for maximum speed.

The user interface provides selection of paths to access input files and write output files into a given destination. A secret key section is common for all algorithms. It is possible to generate a key, read/write from/to a file, input the key via a keyboard. A key length varies from 8 to 256. Selection of the cipher tab provides setting additional parameters for that cipher. It is possible to generate public parameter (nonce) and provide the MAC-tag for the message authentication (MAC – Message Authentication Code).

After setting all necessary parameters the type of operation (encryption or decryption) is selected. The log window allows the user to trace the consecutive steps of program setting and execution. The progress of data processing is visualized. The main window of the application is shown in Fig. 2.

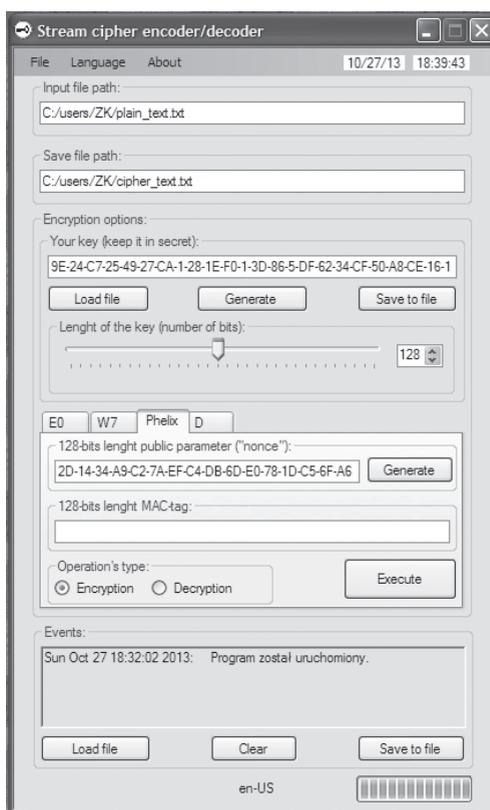


Fig. 2. GUI of the stream cipher encoder/decoder

5. Hardware implementations

The stream cipher encoder/decoder has been implemented in VHDL in Xilinx WebPack ISE v.8.2.03i. The destination device Spartan 3E-500 FG320 on Digilent Nexys2 development board has been used. The FPGA circuit has got 500 000 equivalent gates. A USB port is provided for the power supply and communication between the PC and FPGA. Embedded SDRAM has a capacity of 16MB. A quartz oscillator runs with 50 MHz frequency.

Data transmission between PC and FPGA memory is always 8-bit, but data processing within FPGA is either 8-bit (E0 and W7 ciphers) or 32-bit in buffered mode (Phelix and D ciphers).

During the synthesis phase, we have used options Optimization goal – Speed and Optimization Effort – Normal. In the implementation phase, the option Optimization Strategy – Area has been used. The collected data from the synthesis reports are presented in Table 1.

E0 encoder employs the simplest architecture, while D encoder employs the most complex architecture. The complexity of the structure also has influence on the minimal clock period. Normalization of the clock period for all encoders at 20 [ns] has become possible by means of the flag system that was introduced for synchronization of internal transitions in all finite state machines (FSM) that control the work of encoders.

As we will see in the next section, the clock frequency does not necessarily influence the device's throughput: f. i. W7 and E0 encoders, with the highest and the lowest throughput respectively, have very similar minimal clock periods.

Table 1

Synthesis data for Spartan 3E-500 device

Encoder	Input IOB	Output IOB	Bi-direct. IOB	Number of LUTs	Number of slices	Number of Gates	Minimal clock period [ns]
E0	19	56	24	1555	963	24 711	11.681
W7	19	56	24	3186	1685	38 769	11.758
Phelix	19	56	24	5721	3088	65 528	27.729
D	19	56	24	7900	4151	63 188	116.653
Available resources	232 (all types)			9312	4656	500 000	–

Table 2

Synthesis data for Virtex4 family

Encoder	Number of LUTs	Number of slices	Number of gates	Minimal clock period [ns]
E0	1802	1069	19 855	4.875
W7	3282	1808	33 488	4.684
Phelix	5683	3054	58 579	13.224
D	7826	4088	55 177	64.719
Available resources	12 288	6144	–	–

In Table 2, synthesis data for Virtex 4 family of Xilinx FPGAs are shown for comparison with the Spartan 3 device. There are several differences between implementations resulting from architectural differences. The most interesting result is a significantly higher possible speed of the Virtex 4 implementations.

6. Comparison of software and hardware encoders

In the conducted experiments, the application presented in section 3 was tested on a 2.0 GHz computer with 3.0 GB RAM, running under 32-bit Windows Vista OS. Speed of encoding was measured for 16 data files within the range of 1-16 MB which corresponds with the maximum memory size 16 MB in the hardware encoder implementations on Spartan 3E FPGA. The linear growth of encoding time with file size is observed in Fig. 3. The average throughput of 115,72 Mbit/s for Phelix, 27,78 Mbit/s for D, 11,57 Mbit/s for E0 and 7,70 Mbit/s for W7 was obtained.

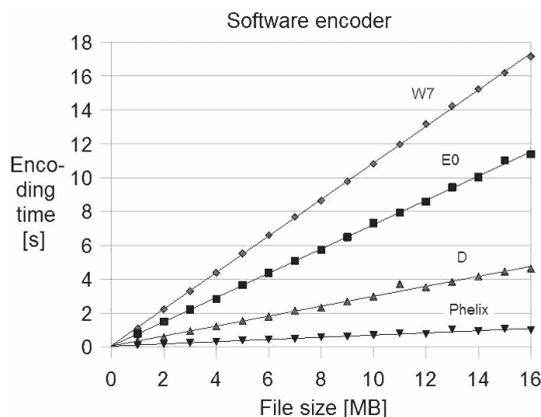


Fig. 3. Comparison of encoding times of the software encoders (W7, E0, D and Phelix)

In the conducted experiments, the hardware encoders presented in section 3 were tested on Xilinx Spartan 3E FPGA. Speed of encoding was measured for 16 data files within the range of 1-16 MB. The linear growth of encoding time with file size is observed in Fig. 4. The average throughput of 16,26 Mbit/s for W7, 12,69 Mbit/s for D, 12,67 Mbit/s for Phelix and 10,20 Mbit/s for E0 was obtained.

Simultaneously, W7 has got the lowest throughput among software encoders. It justifies a conclusion that encoders composed on LFSR are devoted mostly to hardware implementations. E0 encoder, which is also built on LFSR, delivers another data. Its hardware version takes the last place and its software version the 3rd place. Its low hardware throughput is due to an inefficient key stream generator, which produces only one key bit in one clock cycle. Implementations of encoders processing 32-bit characters (Phelix and D) are definitely the fastest among the tested software versions. The Phelix encoder outperforms all other software encoders. Also its hardware version has the high throughput.

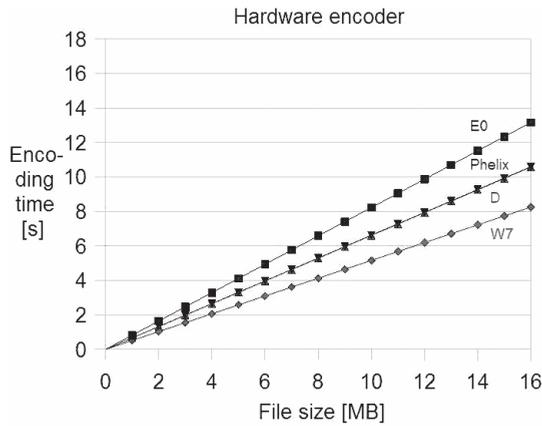


Fig. 4. Comparison of encoding times of the hardware encoders (E0, Phelix, D and W7)

The comparison of hardware and software encoder throughputs is depicted in Fig. 5. The most efficient hardware implementation of the encoder algorithm is that for W7. Its FPGA implementation reveals higher throughput than the software version, while the clock frequency of Nexys 2 board is 40 times lower than that of the processor.

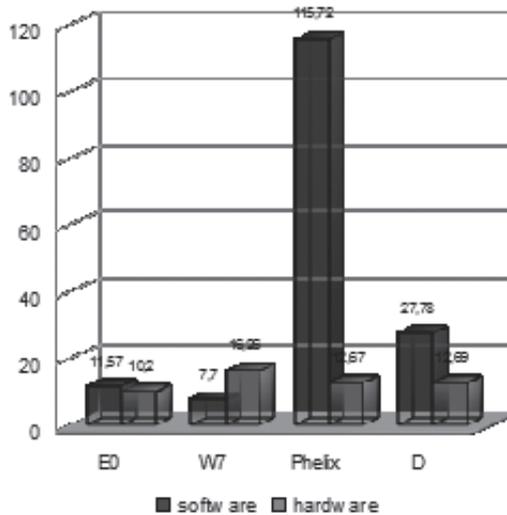


Fig. 5. Comparison of SW/HW encoder throughputs (E0, W7, Phelix, D)

7. Concluding remarks

All four encoding schemes were successfully implemented and tested in software and hardware. The software implementation of the new D stream cipher implemented by the authors is much better than E0 and W7 in terms of the average throughput, but the winning

algorithm in this category is Phelix. The differences between hardware versions of encoders are less visible. The fastest hardware encoding provides W7, while D and Phelix encoders occupy the second place. The slowest one in this category is the E0 encoder.

Properties of the D stream cipher were not verified via cryptanalysis so far. It is expected that selection of key scheduling scheme shall play an important role. In order to increase robustness of the proposed method on cryptanalytic attacks, application of derangements $D(n)$ for n different than the power of two might be considered.

It is possible to develop encoding schemes similar to D encoder on the basis of other derangement generation algorithms [1, 3, 9, 13]. Alternative hardware encoder may be constructed with the parallel derangement generator in the linear array model [2].

The idea of using derangements instead of classical permutations may lead to the modification of VMPC one-way function [18, 19] into the VMDC (Variable Modified Derangement Composition) one-way function. In this way, the VMPC encryption algorithm would become the VMDC encryption algorithm. However, many details of the former scheme, like the key scheduling algorithm, should be adapted to the new cipher. It would be interesting to compare cryptographic properties of both variants.

References

- [1] Akl S.G., *A new algorithm for generating derangements*, BIT 20, 1980, 2-7
- [2] Akl S.G., Calvert J.M., Stojmenović I., *Systolic generation of derangements*, Proc. Int. Workshop on Algorithms and Parallel VLSI Architectures II, Elsevier, 1992, 59-70.
- [3] Baril J.-L., Vajnovszki V., *Gray code for derangements*, Discrete Applied Mathematics, 140, 2004, 207-221.
- [4] Denning D.E.R., *Cryptography and data security*, Addison-Wesley, 1982.
- [5] Erickson M.J., *Introduction to Combinatorics*, Wiley Interscience, 83, 1996, 119-120.
- [6] Gajos T., *A hardware implementation of VMPC stream cipher encoder and decoder in programmable logic*, M.S. Thesis, Cracow University of Technology, Cracow 2009 (in Polish).
- [7] Gehrman C., Persson J., Smeets B., *Bluetooth Security*, Artech House, 2004.
- [8] Gupta P., Bhattacharjee G.P., *A parallel derangement generation algorithm*, BIT 29, 1989, 14-22.
- [9] Graham R.L., Knuth D.E., Patashnik O., *Concrete Mathematics*, 2nd ed., Addison-Wesley Publishing Company, 1994, 194-195.
- [10] Kokosiński Z., *On generation of permutations through decomposition of symmetric groups into cosets*, BIT 30, 1990, 583-591 (available at www.pk.edu.pl/~zk/pubs.html).
- [11] Kokosiński Z., *Circuits generating combinatorial configurations for sequential and parallel computer systems*, Cracow University of Technology, Cracow, Poland, Monograph, 160, 1993.
- [12] Kokosiński Z., *On parallel generation of partial derangements, derangements and permutations*, Proc. Int. Conf. Parallel Processing and Applied Mathematics PPAM'2007, Gdansk, Poland, Lecture Notes in Computer Science, Vol. 4967, 2008, 219-228 (available at www.pk.edu.pl/~zk/pubs.html).

- [13] Korsh J.F., LaFolette P., *Constant time generation of derangements*, Information Processing Letters, Vol. 90, 2004, 181-186.
- [14] Rogawski M., *Stream ciphers in FPGA structures*, IX Konferencja Zastosowań Kryptografii, Enigma'2005, Warszawa 2005 (in Polish).
- [15] Thomas S., Deven A., Berson T., Gong G., *The W7 stream cipher Algorithm*, Internet draft, April 2002.
- [16] Whiting D., Schneier B., Lucks S., Muller F., *Phelix fast encryption and authentication in a single cryptographic primitive*, ECRYPT Stream Cipher Project Report 2005/027, 2005.
- [17] Wójcik S., *A hardware implementation of stream encoder and decoder of the data in programmable Spartan 3 FPGA devices*, M.Sc. Thesis, Cracow University of Technology, Cracow 2010 (in Polish).
- [18] Żółtak B., *VMPC one-way function and stream cipher*, Proc. Int. Conf. Fast Software Encryption FSE'2004, Delhi, February 2004.
- [19] VMPCrypt: <http://www.vmpcrypt.pl/>

ADRIENN BURUZS*, MIKLÓS. F. HATWÁGNER**, LÁSZLÓ T. KÓCZY***

MODELING INTEGRATED SUSTAINABLE WASTE MANAGEMENT SYSTEMS BY FUZZY COGNITIVE MAPS AND THE SYSTEM OF SYSTEMS CONCEPT

MODELOWANIE ZINTEGROWANYCH ZRÓWNOWAŻONYCH SYSTEMÓW GOSPODAROWANIA ODPADAMI ZA POMOCĄ ROZMYTYCH MAP POZNAWCZYCH ORAZ KONCEPCJI SYSTEMU SYSTEMÓW

Abstract

This paper describes the problems relating to the complexity of modern waste management systems. We present a new approach to selecting a better waste management solution. For a large and complex system it is extremely difficult to describe the entire system by a precise mathematical model. Therefore, we propose the use of Fuzzy Cognitive Maps (FCM), its combination with the Bacterial Evolutionary Algorithm (BEA) and the system of systems approach to support the planning and decision making process of integrated systems.

Keywords: sustainability, integrated waste management system (IWMS), fuzzy cognitive map (FCM), bacterial evolutionary algorithm (BEA), system of systems (SoS) approach

Streszczenie

W niniejszym artykule opisano problemy związane ze złożonością nowoczesnych systemów gospodarowania odpadami. Przedstawiono nowe podejście pozwalające na wybór lepszego rozwiązania gospodarki odpadami. W przypadku dużego i złożonego systemu, opisanie całego systemu za pomocą dokładnego modelu matematycznego przysparza ogromnych trudności. Stąd, w artykule zaproponowano użycie rozmytych map poznawczych oraz ich połączenia z bakteryjnymi algorytmami ewolucyjnymi, a także ujęcie systemu systemów w celu wspomaganie procesu planowania i podejmowania decyzji w zintegrowanych systemach gospodarki odpadami.

Słowa kluczowe: zrównoważony zintegrowany system gospodarki odpadami, rozmyte mapy poznawcze, bakteryjny algorytm ewolucyjny, ujęcie systemu systemów

* M.Sc. Adrienn Buruzs, e-mail: buruzs.adrienn@freemail.hu, Department of Environmental Engineering, Faculty of Engineering, Széchenyi István University, Győr.

** Ph.D. Miklós. F. Hatwágner, Department of Information Technology, Faculty of Engineering, Széchenyi István University, Győr.

*** Prof. D.Sc. Ph.D. László T. Kóczy, Department of Automation, Faculty of Engineering Sciences, Széchenyi István University Győr; Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics.

Symbols

- V_k – the state k of the system
 N – the matrix of the system which contains the weight ω_{ij}
 $\lambda > 0$ – determines the steepness of the continuous function f

1. Introduction

Waste is one of the most visible environmental problems in the world [1]. Integrated waste management systems (IWMS) are real elements of our everyday life, therefore problems generated from these systems are real problems. The waste management system consists of a whole set of activities related to treating, transporting or recycling the waste materials. Waste management has evolved from the simple transportation of waste to landfills to complex systems, including several treatment and landfill techniques [2]. Modern waste management presents a high level of complexity. The purpose of waste management is to provide sanitary living conditions, to reduce the amount of materials that enters or leaves the society and to encourage the reuse of materials in the society [1].

Sustainability is an essential goal for the planning and management of natural resources. A system is sustainable if it is appropriate to the local conditions in which it operates from several perspectives. In addition, if it is capable to maintain itself over time without reducing the resources needed [3]. Sustainable waste management means less reliance on landfill and greater amounts of recycling and composting [1]. Sustainable IWMS should be environmentally efficient, economically affordable and socially acceptable [4], this way providing a comprehensive interdisciplinary framework for addressing all problems of managing urban solid waste. Realizing sustainable development, especially of the waste management sector, is therefore a great challenge.

Achieving sustainability in waste management requires an integrated approach. A system is integrated if it uses a range of inter-related collection and treatment options, involves all stakeholders, and takes into account interactions between the waste management system and other urban systems [3]. Thus, the selection of a better waste management solution requires many aspects to be considered.

The rest of this paper is structured as follows. Section 2 describes the history and background of sustainable waste management and introduces the driving factors of the IWMS. (Hereinafter, in this paper the following expressions ‘driving factors’, ‘key drivers’ and ‘concepts’ have the same meaning; they stand for ‘factors’ that are the determining component of the waste management systems). Section 3 presents the methodological approach of the simulations by two Computational Intelligence tools: Fuzzy Cognitive Maps (FCM) and Bacterial Evolutionary Algorithms (BEA). Section 4 describes the results of the simulations. In chapter 5, we introduce the system of systems approach (SoS) and describe the basic subcomponents of the waste management system. Finally, a summary is given in Section 6 where future research intentions are described within the framework of SoS and the sustainability factors of waste management.

2. History and background

The IWMS has to be an economically affordable, environmentally effective and socially acceptable system. Among others, it includes the practical aspects of waste management (i.e. transport, treatment and disposal) and the attitudes of citizens (how they feel about source separation, recycling, incineration etc.). The evolution of waste management from truck and dump, to the highly integrated systems requires an investment of both time and resources [44].

Numerous studies introduce the history of waste management. According to [45], until the 1960s, municipal waste management was concentrated only on the collection and transportation of waste from households to the disposal facilities without any separation, which in the majority of cases, were local dumps or landfills. Processes were planned or optimised merely on the basis of efficiency in terms of costs. Environmental effects were only marginally taken into account. In the second phase, waste treatment and landfilling technologies were improved. After [46], in the 1970s, the goals of the municipal waste management systems were simply to optimize waste collection routes for vehicles or to locate appropriate transfer stations. In the 1980s, the focus was extended to encompass municipal waste management on a system level, minimizing the costs. This was the first time that the aspect of waste as a resource was taken into consideration. Complex waste management systems were first introduced and further developed from the 1980s onward. In the 1990s, specific treatment technologies for several types of waste were introduced, together with advanced landfill technologies [47]. With the transition from waste management to materials management, tools are needed that consider all aspects and effects of waste management [44].

In the preliminaries of this research, we investigated the conditions of sustainability of IWMS and determined its six driving factors. According to a general consensus in the literature these are the following: *environmental; economic; social; institutional; legal and technical* factors [3–8]. These factors are the ‘key drivers’ of a sustainable IWMS that influence why the system operates as it does.

In Table 1, the main factors and some examples of their respective subsystems are introduced.

We have accepted this approach as well-founded. However, some of the results of our present research motivate us to re-validate the inputs by the stakeholders in a later phase of the investigation. The level of modelling that is commonly presented in the relevant literature is not sufficient to determine the weight of each factor, therefore a more detailed approach to modeling is needed.

Modern IWMS are complex and are inherently comprised of a large number of interacting components. These systems have nonlinear behavior and cannot simply be derived from the summation of analyzed individual component behaviors. In this application, we were interested in investigating under what conditions an IWMS may be sustainable.

The modeling of complex systems requires new methods that can utilize the existing knowledge and human experience. These methods are equipped with sophisticated characteristics such as optimization and identification qualities [9]. It is obvious that uncertainties involved with waste management represent vagueness rather than probability. Fuzzy sets and fuzzy logic are suitable to construct a formal description and a mathematically manageable model of systems and processes with such uncertainties. Due to the incompleteness and multiple uncertainties occurring in sustainable waste management systems, we proposed

the use of FCM to support the planning and decision making process. By the observation of the model and its time dependent behavior, we can determine under what conditions the long-term sustainability of a regional waste management system could be ensured.

Table 1

Factors of IWMS and their respective subsystems

Factors	Subsystem elements
Environmental factors	Emissions; Climate change; Land use; Recovery and recycling targets; Depletion of natural resources; Human toxicity
Economic factors	Efficiency at subsystem level; Efficiency at system level; Available funding/subsidies; Equity; System costs and revenues; Pricing system for waste services, Secondary materials market
Social factors	Public opinion; Public participation in the decision making process; Risk perception; Employment; Local demographics – population density, household size and household income; Public resistance (NIMBY – Not In My BackYard, LULU – Locally Unacceptable Land Use)
Institutional factors	Local and regional politics and planning; Managerial conditions and future directions; Institutional and administrative structure of waste management
Legal factors	Relevant legislation (international, national, regional and municipal)
Technical factors	Collection and transfer system; Treatment technologies; Waste stream composition and change

FCM is an ideal tool for modeling multi-attribute systems, especially when they incorporate such ‘soft’ parameter as human factors, environmental characteristics or societal concepts [10]. Our goal was to develop an objective, state-of-the-art model and ‘tool kit’ that could be used to take highly informed and focused decisions regarding sustainable integrated solid waste management on a regional level.

This study aims to provide a method, which uses the BEA algorithm to develop FCM connection matrices based on historical data consisting of one sequence of state vectors. In contrast, some other methods introduced alternative approaches, which require a whole set of such sequences. The goal of the simulation was to assess the sustainability of the IWMS by investigating the FCM methodology applying the BEA with a holistic approach [11, 12].

3. Methodological Approach

In the next two subchapters the applied Computational Intelligence tool kit will be briefly described.

FCM is a very convenient and simple tool for modelling complex systems. It is rather popular due to its simplicity and user friendliness. Its one disadvantage is that it is not able to extrapolate properly from the available time series data, it always converges to a set of

‘plateaus’, i.e. an assumed stable state. The present research deploys the FCM and applies the BEA for parameter optimization.

3.1. Fuzzy Cognitive Map

On the basis of a FCM’s development, during the first step in the design process, the number and features of concepts are determined by a group of experts. After the identification of the main factors affecting the topic under investigation, each stakeholder is asked to describe the existence and type of the causal relationships among these factors and then assesses the strength of these causal relationships using a predetermined scale, capable of describing any kind of relationship between two factors, positive and negative.

Starting from the primary elements of a FCM, the i -th concept denotes a state, a procedure, an event, a variable or an input of the system and is represented by C_i ($i = 1, 2, \dots, n$). Another component of a FCM is the directed edge which connects the concepts i and j . Each edge includes a weight w_{ij} which represents the causality between concepts C_i and C_j . The values of the concepts are within the range $[0, 1]$, while the values of the weights belong to the interval $[-1, 1]$. A positive value of the weight w_{ij} indicates that an increase (decrease) in the value of concept C_i results to an increment (decrement) of the concept’s value C_j . Similarly, a negative weight w_{ij} indicates that an increase (decrease) in the value of concept C_i results in a decrement (increment) of the concept’s value C_j , while a zero weight denotes the absence of a relationship between C_i and C_j (Fig. 1). Considering the interrelations between the concepts of a FCM, the corresponding adjacency matrix can easily be formed.

Usually, it is accepted that causality is not self reflexive, i.e., a concept cannot cause itself, which means that the weight matrix always has ‘0-s’ in its diagonal [48]. Otherwise the component would grow without limits.

The description of the inference mechanism, which represents the behaviour of the physical system, lies in the interpretation of FCM’s mathematical formulation. After the initialization of the FCM and the determination of concept activation values by experts, concepts are ready to interact. As is obvious, the activation of a concept influences the values of concepts that are connected to it. At each step of interaction (simulation step), every concept acquires a new value that is calculated according to equations (Equation 1 and 2) and the interaction between concepts continues until fixed equilibrium is reached, a limit cycle is reached, or a chaotic behaviour is observed [49].

The mathematical description of our FCM system is a simple loop:

$$V_{k+1} = f(N \cdot V_k) \quad (1)$$

where:

- V_k – the state k of the system,
- N – the matrix of the system which contains the weight w_{ij} , and

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (2)$$

where $\lambda > 0$ determines the steepness of the of the continuous function f .

Several models have been developed in recent decades to support decision making in IWMS to monitor present conditions, to assess future risks and to visualize alternative futures [13, 14]. Many environmental problems would benefit from models based on the experts' knowledge [15], among them IWMS modelling as well. The methodology extracts the knowledge from the stakeholders and exploits their experience of the system's model and behaviour.

In the development of the FCM, in the first step of the design process, the number and features of the constituting factors were determined by the relevant literature, as it was mentioned beforehand. These factors are supposed to be combined together in a single system, with mutual interactions. We have conducted an online survey where each one of the stakeholders was asked to describe the existence and type of the causal relationships among the six concepts and then to assess the strength of these using a predetermined simple scale, capable of describing any kind of relationship between a pair of factors, both positive and negative. Thus, from each interviewee, theoretically, a different hypothetical FCM could be established. As a positive aspect of this study, we have to notice that the participants were highly motivated to take part in the survey process without the need to understand the mathematical background of the methodology.

The 75 individual maps were however, merged into a representative, collective map. In this phase, we were primarily interested in investigating how the stakeholders perceived the future prospects of the IWMS.

FCMs are fuzzy graph structures representing causal reasoning. Causality is represented here as a fuzzy relation of causal concepts. FCM may be used for the dynamic modelling of systems. The FCM approach uses nodes corresponding to the factors and edges for their interactions, to model different aspects in the behaviour of the system. These factors interact with each other in the FCM simulation, presenting the dynamics of the original system [16]. FCMs have been described as the combination of Neural Networks and Fuzzy Logic. Thus, learning techniques and algorithms can be borrowed and utilized in order to train the FCM and adjust the weights of its interconnections [17].

3.2. The Bacterial Evolutionary Algorithm

In order to optimise the FCM-based model, the BEA was chosen because our previous experiences and results with various benchmark data sets revealed that BEA and Bacterial Memetic Algorithms (BMA) were among the most efficient evolutionary algorithms [18, 19]. This was especially true for the variants equipped with the most appropriate and suitable operators (see below). Several papers presented comparisons of these algorithms with other evolutionary and population based heuristics, e.g. when the goal was fuzzy rule-based learning of various physical models [19, 20], or when the Permutation Flow Shop Problem had to be optimised under certain conditions [21, 22].

The BEA was originally proposed by Nawa and Furuhashi in the late 1990s as a new evolutionary algorithm [23, 24]. This algorithm was established as a further development of the already existing Pseudo-Bacterial Genetic Algorithm [25] and the classical Genetic Algorithm itself [26, 27]. The name of the algorithm indicates that its operations are similar to the process of the evolution of bacteria. A possible solution of a problem is represented by an individual bacterium. The BEA keeps a record of all available bacteria, i.e., solutions, called the bacterium population. Using the two main operators, bacterial mutation and gene

transfer, it creates successive generations of the population until some kind of termination condition is fulfilled. Finally, the best bacterium of the last generation is considered as the result, i.e., the best approximation of the optimal solution. During the simulation process, the bacterial mutation creates new versions of bacteria with random modifications. In other words, this operator is liable for the exploration of the search space. Depending on some parameters governing the spread or deviation of the mutation results, its properties balance between ‘globalness’ and convergence speed. The other operator, namely gene transfer, combines the genetic information of pairs of bacteria. Thus it performs the exploitation of the genetic data. Further details can be found e.g. in [28].

Some major benefits of the operators are that they realize elitism without additional computational efforts, and the implementation of them is very straightforward. The properties of the algorithm are similar to the ones of other evolutionary algorithms, even though our experience shows that for most types of problems, it provides better approximation and convergence than the others. It cannot typically determine the exact solution of the examined problem, however, it approximates the global optimum. Theoretically, the accuracy might be arbitrarily good and the probability of finding the exact optimum in discrete problems might be arbitrarily large [29]. On the other side, BEA is able to optimise or solve complex problems even if they are not continuous, noisy, high-dimensional, non-linear or multimodal. Several researchers proposed new operators or modifications to improve the algorithm which are various BMAs [30, 31]. In these cases, the main idea is to decrease the number of objective function evaluations using a local search algorithm (e.g. the rather efficient, but also rather complicated, Levenberg-Marquardt algorithm) [32, 33]. Other researchers proposed modified gene transfer operators to allow parallel computation of the objective values [34].

In the literature, there are some references provide which a good overview on the soft computing tools [50–52].

4. Results

In the first simulation, our starting point was a fixed connection matrix. In this approach, we studied the changes of the importance values of the factors over time.

The second experiment was about parameter identification using BEA. The connection matrix of FCM was determined so that the difference between the original time series of concepts given in literature and the generated ones using this matrix should be as small as possible.

4.1. Results with the FCM Simulation

The goal of this first experiment [35] was to assess the sustainability of the IWMS by investigating the FCM methodology with a holistic approach. First, the input data, then the experience obtained during the simulation are presented and finally, the results are introduced.

The model consists of the expert system database which is based on human expert experience and knowledge obtained from the questionnaires. Namely, the initial draft connection matrix is the data gathered and averaged from the survey process shown in Table 2. This model includes the identification of concept nodes and the relationships among them, these are represented by edges.

Table 2

The initial draft of the connection matrix

	C1	C2	C3	C4	C5	C6
C1	0	0.8	0.6	0.6	0.4	0.4
C2	0.6	0	0.6	0.6	0.4	0.4
C3	0.8	0.6	0	0.6	0.4	0.4
C4	0.4	0.6	0.4	0	0.4	0.4
C5	0.6	0.6	0.4	0.6	0	0.6
C6	0.4	0.4	0.4	0.4	0.4	0

The factors in the matrix are represented as follows:

- C1 – technical factor;
- C2 – environmental factor;
- C3 – economic factor;
- C4 – social factor;
- C5 – legal factor;
- C6 – institutional factor.

The other input data set was the range of historical data consisting of sequences of the state vectors. According to [2–9], the trend of the studied factors was assessed by values between 0 and 1 from the 1980s to the 2010s. The sequences of the state vector were designed on the basis of the literature and therefore it may be assumed that they soundly specify the role of the factors according to changes in the legislation, the available techniques, the social attitude, and the economic and institutional environment, as a time series (see Table 3, columns t_0 – t_4).

Table 3

The sequences of the state vectors

	t_0	t_1	t_2	t_3	t_4	FCM averages
Technical	0.20	0.35	0.60	0.75	0.80	0.80
Environmental	0.15	0.20	0.40	0.60	0.80	0.71
Economic	0.10	0.15	0.30	0.50	0.70	0.62
Social	0.10	0.15	0.20	0.40	0.60	0.56
Legal	0.10	0.30	0.50	0.70	0.80	0.71
Institutional	0.10	0.20	0.30	0.50	0.60	0.58

Unfortunately, the number of available data is very small. From the literature on waste management modelling, only such a small amount of the data can be acquired.

FCM uses fuzzy values to represent the states of factors (concepts) in different moments (time series data) and to describe the strength of connections between the factors.

During the simulation, we determined different values for λ in order to see how the parameter influenced the results of the simulation. The simulation was always started with the input of the above data. The simulation resulted in different iterations according to the value of λ . We scaled the initial state of the system in the $[0, 1]$ interval and we used this model and ran the simulation for 10 iteration cycles. The results are presented below.

From Fig. 1, it can be observed that the system converges to an equilibrium state which is robust to the initial state variation, however, the value of λ is different in each simulation. The estimated optimal value of λ may be determined by comparing the obtained results with the expert system database.

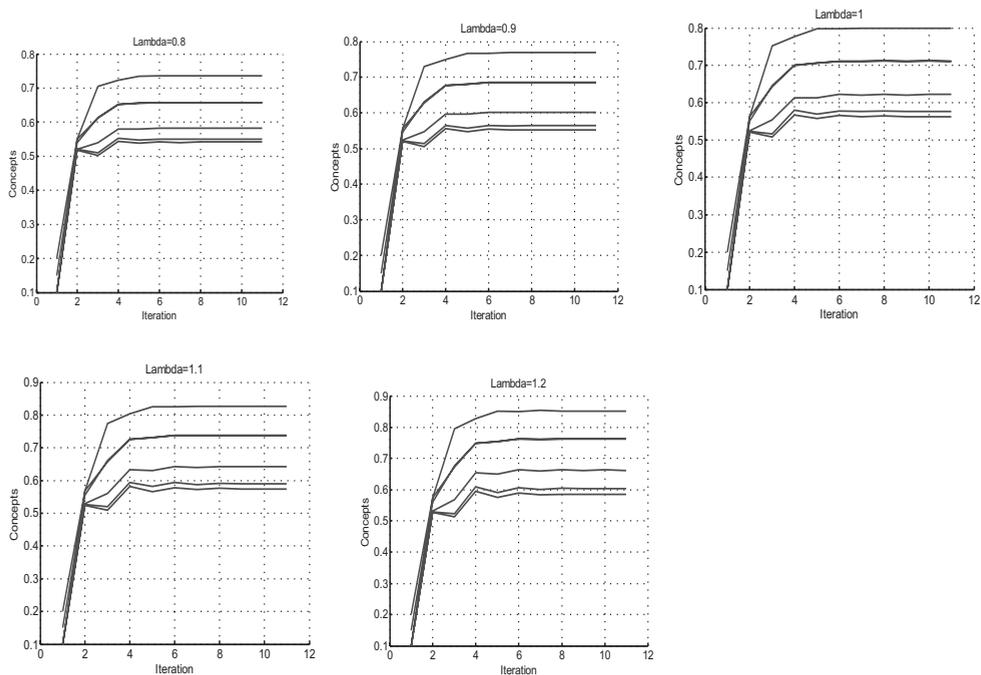


Fig. 1. The model simulation with $\lambda = 0.8; 0.9; 1; 1.1$ and 1.2

It may be observed that in the FCM model, all factors converged rather fast to a steady state. After the first five iterations, the transient behaviour seems to end and the FCM approaches an obviously stable state where each concept assumes a constant value ('plateau', depending on λ , between 0.5 and 0.9). While the qualitative behaviour of the simulation result is virtually independent of the steepness, the actual constant values to which the concept influence state converges are more or less similar, thus after normalization, the results are very consistent.

The initial states of the factors are known from Table 2. The final states of the concepts computed for each λ are shown in Table 4.

However, our strong assumption is that the time series is the most influencing input data in the modeling, to confirm this assumption we will also check in which way different expert matrices influence the results in the next phase of the research.

The final state of the concepts computed for each λ

	λ				
	0.8	0.9	1	1.1	1.2
C1	0.736	0.768	0.799	0.827	0.853
C2	0.659	0.685	0.711	0.737	0.762
C3	0.583	0.602	0.621	0.641	0.662
C4	0.541	0.552	0.563	0.574	0.585
C5	0.659	0.685	0.711	0.737	0.762
C6	0.551	0.563	0.577	0.590	0.603

The average results of simulation with different λ values are presented in the last column of Table 3. As integrated waste management systems are sophisticated and complex systems, priorities and targets need to be set up at the early stage of planning and implementation. Assuming, that the initial values are estimated more or less correctly by the experts, we might conclude the following main statement of the paper: the ranking of the factors below influencing the sustainability of the waste management systems shows the way how the roles and weights of the factors should be considered within an IWMS in order to ensure environmental efficiency, economical affordability and social acceptability, this way providing a comprehensive interdisciplinary framework for addressing all problems of managing urban solid waste.

1. C1 (technical factor),
2. C2 (environmental factor) and C5 (legal factor),
3. C3 (economic factor),
4. C6 (institutional factor),
5. C4 (social factor).

On the basis of this investigation, the priority sequence of factors or components in the waste management systems at the regional level might be declared.

According to the simulation, the first or most important issue is what materials are managed, treated and disposed of and how (features of the collection, transfer and treatment systems, e.g. material recovery, organic material treatment, thermal treatment, and final disposal). Then, the environmental and legal factors, economic issues of the system are following. Finally, the list closes with the social factor where the main issue is to accept the IWMS and to participate in its activities. However, the public plays an important role in sustainable waste management for which the awareness of waste reduction, segregation and recycling need to be enhanced.

We set up the FCM model of the IWMS, and implemented its structure in a way that its parameters and weights were flexibly variable. Even though the FCM model was proposed for the integrated analysis of the sustainability factors of the IWMS on a regional level, the validity of the method is depending on the reliability of the input data. As they were obtained from a wide scope of experts, we are convinced that by using the proposed new approach, sustainable waste management systems may be directly planned and established, at least in any more or less closed geographical area.

4.2. The Identification of the Elements of the Connection Matrix Using BEA

In our second experiment [36], the model uses two different sets of input data. The sources of these two sets are different. One set is based on observations that may be considered more or less objective; observations on the trend of the studied factors in the time period from the 1980s till the 2010s. It is obvious that measuring the mutual influence of various factors within a complex phenomenon, like waste management is not easy. Nevertheless, it might be assumed that the time series published in the related literature [3–8] is based on a consensus concerning the interrelationship of the concepts playing a determinative role in the procedure of waste management, thus these values are widely supported by independent observations and manually calculated partial models. In this research, the following data will be considered ‘objective’, even though they are not obtained by ‘measurements’ of some automatic machinery, but by the observation and evaluation of humans involved in the management of the procedure. It must be clearly understood that our learning model is based on these ‘objective’ data and therefore, it makes it unnecessary to continuously consult the experts in order to obtain up-to-date but entirely subjective data again and again.

Nevertheless, in order to speed up the learning procedure, and to some extent, out of scientific curiosity, we used the data collected from the above mentioned survey. It must be stressed that the results of these questionnaires (which were compared, and the medium values selected for each matrix element as the ‘typical subjective values’ of the given influence) were used only as initial values for the learning procedure, under the assumption that starting with more or less realistic values would speed up the convergence of the matrix to the stable ‘objective’ values. It turned out during the optimization, that the convergence speed is quite high with randomly generated start population as well, thus, prudent composition of the bacteria in the first generation was not an important issue. It is nevertheless interesting to compare the ‘subjective’ mutual influence values obtained from the questionnaires and the ‘objective’ matrix obtained from the time series observed starting with the data from the 1980’s. On the basis of the gathered data, we constructed the initial draft of the connection matrix (Table 1), including identification of concept nodes and their mutual relationships represented by the graph edges.

Simulation in this context consisted of computing the states of the system described by the state vector over a number of successive iterations. In every iteration cycle, the state vector specifies the current values of all factors (the nodes) in a particular moment. The values of the given states (nodes) are obtained from the preceding iteration values of all the nodes which exert influence on the given node through cause-effect relationship. The transformation function is used to confine the weighted sum to the range set to $[0, 1]$. This normalization hinders the absolute quantitative analysis, but allows the comparison between nodes, which are attached by fuzzy activity degrees (defined as ‘active’: 1, ‘inactive’: 0 or ‘active to a certain degree’: values between 0 and 1), see [37].

During the optimisation of our FCM with BEA, forced mutation [38] was used to increase the otherwise very low value of genetic diversity, to speed up computations in this manner. Forced mutation is a simple and easily implementable operator that slightly modifies some bacteria in the population if they seem very similar (typically in the final generations of the optimization). Forced mutation was applied in all subsequent generations after gene transfer.

The value of λ used by the transformation function was represented by the first gene of the bacteria. The following 30 genes corresponded to the elements of the 6×6 connection matrix (without the elements of the main diagonal, which were not stored).

The FCM determined the values of the factors in the subsequent iterations using the connection matrix. The goal of using the BEA heuristics was to find a connection matrix that minimizes the difference between the state values obtained from the literature (see Table 2) and the generated values of the factors. This difference d is expressed in Equation 3.

$$d = \sum_{i=1}^6 \left([c_i]_t - [\hat{c}_i]_t \right)^2 \tag{3}$$

where $[c_i]_t$ denotes the real and $[\hat{c}_i]_t$ the calculated values of factors.

The results of the optimization are contained in the connection matrix presented in Table 5. Here $\lambda = 1$, which resulted in $d = 0.727$ between the obtained and the state vectors suggested by experts. It is rather surprising how far the interrelation coefficients obtained by automatic learning (based on the more or less objective data of the time series observed) are from the coefficients calculated from the median of the experts' questionnaires. We have no doubt that the matrix obtained by learning is rather independent from subjective elements, especially as it resulted from data obtained throughout a relatively long observation period. The fact that expert opinions differ so much from the objective reality definitely poses a question of how deep the insight of waste management experts may be wherever the system on hand is constituted from a set of complex technical, environmental and social subsystems consisting of several mutually influencing (and rather fluctuating) factors.

Table 5

The resulting optimized connection matrix

	C1	C2	C3	C4	C5	C6
C1	0	-0.393	1	-1	1	0.753
C2	0.212	0	1	-1	1	-1
C3	-0.722	1	0	-1	1	-1
C4	-1	0.377	-1	0	1	-1
C5	1	1	0.749	-1	0	-1
C6	-1	0.821	-1	-1	1	0

While in this approach we tried to optimize parameters with the help of the BEA and thus obtained a single set of results for the connection matrix in an alternative research [11], we found that results obtained with various, non-optimal steepness values λ , the results differed essentially only in the scaling. After normalization, all estimated time series predictions converged to essentially the same limit values.

5. Present research

From the unexpected results, the fact that the connection matrix obtained from the observation data is so thoroughly different from the matrix given by the experts that the obvious question arises of whether the approach and the objective results are mathematically stable enough in terms of the uncertainty of the observed values. It is also evident, that the program performed the simulation with different levels of credibility. In cases where input data correspond to reality, the method is suitable for simulating the problem and providing accurate results.

Based on the above results, in this paper, we propose the application of the systems of systems (SoS) approach to regional IWMS.

A system is the collection of main factors and their interrelationships gathered together to form a whole greater than the sum of its parts [41]. The knowledge necessary for managing complex projects, for the development complicated of systems, has not kept pace with the increasing complexity and integration of these projects themselves. This increased complexity has permitted some to establish distinctions among systems projects and to propose a framework of systems called the system of systems (SoS) [42].

Despite the fact that a waste management system consists of only six main factors, it is obvious now that properly overviewing the whole procedure needs an approach based on the systems of systems concept [36]. The latter approach is namely suitable to handle problems with essentially different types of system components' where interoperability and seamless interfacing is necessary. The application of this approach then easily leads to the unexpected emerging phenomena – such as the surprising values in the resulting connection matrix. The results obtained by the FCM model are unambiguously such emerging features that will necessarily lead to the re-evaluation of the knowledge and views of environmental engineers dealing with waste management.

From the unexpected results, the fact that the mutual influence matrix obtained from the observation data is so thoroughly different from the matrix given by the experts that the obvious question arises of whether the approach and the objective results are mathematically stable enough in terms of the uncertainty of the observed values. It is also evident, that the program performed the simulation with different levels of credibility. In cases where the input data correspond to reality, the method is suitable for simulating the problem and providing accurate results.

Based on the above results and conclusion, we propose the application of the systems of systems (SoS) theory.

The challenge with the SoS emerges in the interoperability and interfacing of the component systems. SoS integration is a method to pursue development, integration, interoperability, and optimization of systems to enhance performance, but it definitely needs a view that includes all views of the disciplines associated with the constituent systems.

We intended to resolve the contradictions between the previous models generated from observed time series on one side, and experts' estimated influence degrees on the other side, and to go below the level of generally recognized components, decomposing the factors into up to around fifty subcomponents, partly revealing interconnections among the main factors on a primary level. In order to be able to establish this extremely complex and completely novel model of IWMS, we applied the SoS approach which is shown in Fig. 2.

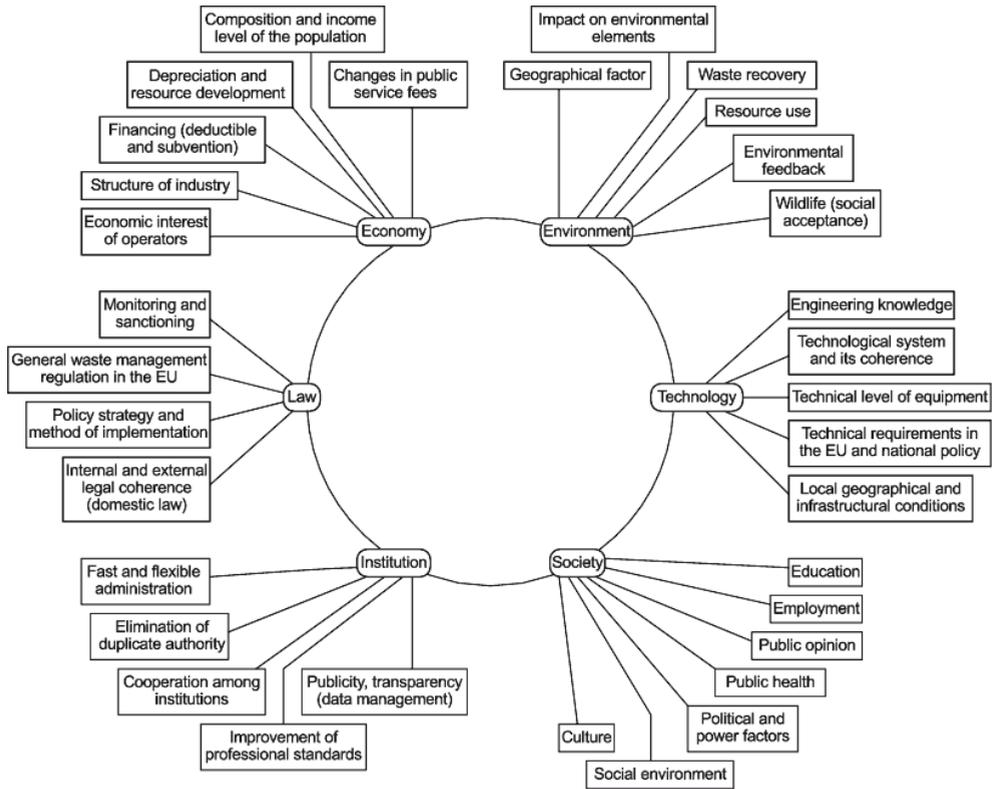


Fig. 2. Basic SoS approach: main factors and subcomponents of regional IWMS

This guaranteed that among subsystems of different types and with various influence surfaces complete interoperability and seamless interfacing could be provided, and thus a deeply justifiable and relevant hierarchical adaptive FCM network model of IWMS can be established that may be used for actually determining the optimal inputs belonging to any intended change in the sustainable states while adequately predicting any unexpected emerging phenomena as well.

In the close future, our intention is to also validate the developed model by experts with the help of the Delphi method and SWOT analysis. The expected results of the future investigation may help to determine the essential steps towards solving this complex problem in the long term and obtain technologies for the sustainable maintenance of the municipal waste management system.

6. Summary

The sustainable decision-making model is a combination of FCM and BEA soft computing tools. The proposed model provides an effective means of assisting in determining the

main effecting elements of IWMS in the decision-making process and to solve real world waste management problems. The model can quantify and qualify the degree of efficiency of the factors. The sustainable decision making model not only accommodates economic, environmental and social factors simultaneously, but also incorporates legal, institutional and technical issues.

In the future, we intend to carry out a sensitivity analysis of the above methods. In cases where the input data correspond to reality, the method is suitable for simulating the problem and providing accurate results. The speed and convergence of the learning method need to also be investigated by hybrid and combined evolutionary and memetic algorithms which were proven to be better than other simple algorithms.

In the recent past, some emerging economies have gone through a very rapid industrial development which resulted in an increase of their GDP. Because of the sharp rise in the production volume, it is very important to alleviate societal, economic and environmental concerns over the increased rate of resource consumption and waste production [43]. We also wish to extend our research to the investigation of the waste management of emerging countries.

The authors would like to thank to the National Science Research Fund (OTKA) K105529, K108405 and the Social Renewal Operational Programme (TÁMOP) 4.1.1.C -12/1/KONV-2012-0017 grant for the support of the research.

References

- [1] Demirbas A., *Waste management, waste resource facilities and waste conversion processes*, Energy Conservation and management, 52, 2011, 1280-1287.
- [2] Salhofer S., Wassermann G., Binner E., *Strategic environmental assessment as an approach to assess waste management systems. Experiences from an Austrian case study*, Environmental Modelling & Software, 22, 2007, 610-618.
- [3] van de Klundert A., Anschutz J., *Integrated sustainable waste management: the selection of appropriate technologies and the design of sustainable systems is not (only) a technological issue*, Paper prepared for the CEDARE/IETC Inter-regional Workshop on Technologies for Sustainable Waste Management, 13–15 July 1999 Alexandria.
- [4] Morrissey A.J., Browne J., *Waste management models and their application to sustainable waste management*, Waste Management, 24, 2004, 297-308
- [5] Wilson E.J., McDougall F.R., Willmore J., *Euro-Trash: Searching Europe for a More Sustainable Approach to Waste management*, Resources Conservation and Recycling, 31, 2001, 327-346.
- [6] Langa D.J., Binder C.R. et al., *Material and Money Flows as a Means for Industry Analysis of Recycling Schemes. A Case Study of Regional Bio-Waste Management*, Resources, Conservation and Recycling, 49, 2006, 159-190.
- [7] den Boer J., den Boer E., Jager J., *LCA-IWM: A Decision Support Tool for Sustainability Assessment of Waste Management Systems*, Waste Management, 27, 2007, 1032-1045.
- [8] Thorneloe S.A., Weitz K., Barlaz M., Ham R.K., *Tools for Determining Sustainable Waste Management Through Application of Life-Cycle Assessment: Update on U.S.*

- Research*, Seventh International Waste Management and Landfill Symposium V, 1999, 629-636.
- [9] Kosko B., *Fuzzy cognitive maps*, Int. J. Man-Machine Studies, 24, 1986, 65-75.
- [10] Perusich K., *System Diagnosis Using Fuzzy Cognitive Maps*, Cognitive Maps, Karl Perusich (Ed.), InTech, 2010.
- [11] Buruzs A., Pozna R.C., Kóczy L.T., *Developing Fuzzy Cognitive Maps for Modeling Regional Waste Management Systems*, Civil-Comp Press, Proceedings of the Third International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering Computing, Y. Tsompanakis, (Ed.), Civil-Comp Press, Stirlingshire, Scotland 2013.
- [12] Buruzs A., Hatwágner M. F., Pozna R.C., Kóczy L.T., *Advanced learning of fuzzy cognitive maps of waste management by bacterial algorithm*, IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), Joint. IEEE, 2013.
- [13] Hung M., Ma H., Yang W., *A novel sustainable decision making model for municipal solid waste management*, Waste Management, 27, 2007, 209-219.
- [14] Papageorgiou E., Kontogianni A., *Using Fuzzy Cognitive Mapping in Environmental Decision Making and Management: A Methodological Primer and an Application*, International Perspectives on Global Environmental Change, S. Young (Ed.), InTech, 2012.
- [15] Özesmi U., Özesmi S.L., *Ecological models based on people's knowledge: a multi-step fuzzy cognitive mapping approach*, Ecological Modelling, Vol. 176, Issues 1–2, 15, 2004, 43-64.
- [16] Stylos C.D., Georgopoulos V.C., Groumpos P.P., *The use of Fuzzy cognitive maps in modeling systems*, Proceedings of 5th IEEE Mediterranean Conference on Control and Systems, Paphos, Cyprus, July 21–23 1997.
- [17] Stylos C.D., Georgopoulos V.C., Groumpos P.P., *The Use of Fuzzy Cognitive Maps in Modeling Systems*, Proc. of 5th IEEE Mediterranean Conference on Control and Systems, Paphos 1997.
- [18] Balázs K., Kóczy L.T., Botzheim J., *Comparative Investigation of Various Evolutionary and Memetic Algorithms*, Computational Intelligence in Engineering, Studies in Computational Intelligence 313, I.J. Rudas, J. Fodor, J. Kacprzyk (eds.), Springer 2010, 129-140.
- [19] Balázs K., Botzheim J., Kóczy L.T., *Comparison of Various Evolutionary and Memetic Algorithms*, Proc. of the International Symposium on Integrated Uncertainty Management and Applications, IUM 2010, Ishikawa 2010, 431-442.
- [20] Dányádi Z., Balázs K., Kóczy L.T., *A Comparative Study of Various Evolutionary Algorithms and Their Combinations for Optimizing Fuzzy Rule-based Inference Systems*, Scientific Bulletin of 'Politechnica' University of Timisoara, Romania, Transactions on Automatic Control and Computer Science, Vol. 55, No. 69, 2010, 247-254.
- [21] Balázs K., Kóczy L.T., *Constructing Dense, Sparse and Hierarchical Fuzzy Systems by Applying Evolutionary Optimization Techniques*, Applied and Computational Mathematics, Vol. 11, No. 1, 2012, 81-101.
- [22] Balázs K., Horváth Z., Kóczy L.T., *Different Chromosome Based Evolutionary Approaches for the Permutation Flow Shop Problem*, Acta Polytechnica Hungarica, Vol. 2, No. 2, 2012, 115-138.

- [23] Nawa N.E., Furuhashi T., *Fuzzy System Parameters Discovery by Bacterial Evolutionary Algorithm*, IEEE Transactions on Fuzzy Systems, Vol. 7, No. 5, 1999, 608-616.
- [24] Nawa N.E., Furuhashi T., *A Study on the Effect of Transfer of Genes for the Bacterial Evolutionary Algorithm*, Second International Conference on Knowledge-Based Intelligent Electronic System, L.C. Jain, R.K. Jain (eds.), Adelaide 1998, 585-590.
- [25] Nawa N.E., Hashiyama T., Furuhashi T., Uchikawa Y., *A Study on Fuzzy Rules Discovery Using Pseudo-Bacterial Genetic Algorithm with Adaptive Operator*, Proc. of IEEE International Conference on Evolutionary Computation, ICEC'97, 1997.
- [26] Bäck T., Fogel D.B., Michalewicz Z., *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press, 1997.
- [27] Goldberg D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.
- [28] Nawa N.E., Furuhashi T., *Fuzzy System Parameters Discovery by Bacterial Evolutionary Algorithm*, IEEE Transactions on Fuzzy Systems, Vol. 7, No. 5, 1999, 608-616.
- [29] Pintér J.D., *Global Optimization in Action*, Kluwer Academic Publishers, Dordrecht 1996.
- [30] Botzheim J., Cabrita C., Kóczy L.T., Ruano A.E., *Fuzzy Rule Extraction by Bacterial Memetic Algorithms*, International J. of Intelligent Systems, Vol. 24, 2009, 312-339.
- [31] Gál L., Kóczy L.T., *Advanced Bacterial Memetic Algorithms*, Acta Technica Jaurinensis, Series Intelligentia Computatorica, Vol. 1, No. 3, 2008, 225-243.
- [32] Botzheim J., Cabrita C., Kóczy L.T., Ruano A.E., *Fuzzy Rule Extraction by Bacterial Memetic Algorithm*, IFSA, Beijing 2005, 1563-1568.
- [33] Gál L., Botzheim J., Kóczy L.T., *Modified Bacterial Memetic Algorithm used for Fuzzy Rule Base Extraction*, CSTST'08 Proc. of the 5th international conference on Soft computing as transdisciplinary science and technology, ACM, NY, USA, 2008, 425-431.
- [34] Hatwágner F.M., Horvath A., *Parallel Gene Transfer Operations for the Bacterial Evolutionary Algorithm*, Acta Technica Jaurinensis, Vol. 4, No. 1, 2011, 89-112.
- [35] Buruzs A., Pozna R.C., Kóczy L.T., *Developing Fuzzy Cognitive Maps for Modelling Regional Waste Management Systems*, Y. Tsompanakis, (ed.), Proc. of the Third International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Civil-Comp Press, Paper 19, Stirlingshire, UK, 2013.
- [36] Buruzs A., Hatwágner M.F., Pozna R.C., Kóczy L.T., *Advanced Learning of Fuzzy Cognitive Maps of Waste Management by Bacterial Algorithm*, IFSA World Congress and NAFIPS Annual Meeting, IEEE, 2013, 890-895.
- [37] Stach W., Kurgan L., Pedrycz W., Reformát M., *Genetic Learning of Fuzzy Cognitive Maps*, Fuzzy Sets and Systems, 153, 2005, 371-401.
- [38] Hatwágner F.M., Horvath A., *Parallel Gene Transfer Operations for the Bacterial Evolutionary Algorithm*, Acta Technica Jaurinensis, Vol. 4, No. 1, 2011, 89-112.
- [39] Hatwágner F.M., Horvath A., *Maintaining Genetic Diversity in Bacterial Evolutionary Algorithm*, Annales Univ. Sci. Budapest, Sec. Comp, Vol. 37, Budapest 2012, 175-194.
- [40] Jamshidi M. (ed.), *Systems of System Engineering. Innovation for the 21th Century*, John Wiley & Sons, Inc. Hoboken, New Jersey, 2009, 480.
- [41] Boardman J., Sauser B., *System of Systems – the meaning of of*, Proc. of the 2006 IEEE/ SMC International Conference on System of Systems Engineering, Los Angeles 2006.

- [42] Boardman J., Sauser B., *From prescience to emergence: taking hold of systems of systems management*, n.a.
- [43] Al-Maaded M., Madi N.K., Kahraman R., Hodzic A., Ozerkan N.G., *An Overview of Solid Waste Management and Plastic Recycling in Qatar*, J. Polym Environ, 20, 2012, 186-194.
- [44] Wilson E.J., McDougall F.R., Willmore J., *Euro-Trash: Searching Europe for a More Sustainable Approach to Waste management*, J. of Resources Conservation and Recycling, 31, 2001, 327-346.
- [45] Shmeleva S.E., Powell J.R., *Ecological-economic Modelling for Strategic Regional Waste Management System*, J. of Ecological Economics, 59, 2006, 115-130.
- [46] Hung, M.-L., Ma, H.-W., Yang W.-F., *A Novel Sustainable Decision Making Model for Municipal Solid Waste Management*, J. of Waste Management, Vol. 27, 2, 2007, 209-219.
- [47] Salhofer S., Wassermann G., Binner E., *Strategic Environmental Assessment as an Approach to Assess Waste Management Systems. Experiences from an Austrian Case Study*, J. of Environmental Modelling & Software, Vol. 22, 5, 2007, 610-618.
- [48] Carvalho J.P., *On the Semantics and the Use of Fuzzy Cognitive Maps in Social Sciences*, WCCI 2010 IEEE World Congress on Computational Intelligence, CCIB, Barcelona 2010.
- [49] Ketipi M.K., Koulouriotis D.E., Karakasis E.G., Papakostas G.A., Tourassis V.D., *A Flexible Nonlinear Approach to Represent Cause-effect Relationships in FCMs*, J. of Applied Soft Computing, 12, 2012, 3757-3770.
- [50] Kecman V., *Learning and Soft Computing, Support Vector machines, Neural Networks and Fuzzy, Logic Models*, The MIT Press, Cambridge 2001.
- [51] Jang J.R., ANFIS, *Adaptive-network-based fuzzy inference system*, IEEE Trans. Syst., Man, Cybern., 23, 1993, 665-685.
- [52] Frayman Y., Wang I.P., *Data mining using dynamically constructed recurrent fuzzy neural networks*, Research and Development in Knowledge Discovery and Data Mining, Vol. 1394, 1998, 122-131.

CONTENTS

M. Zając: Control of electric drive by means of inverse dynamics	3
V. Maksymovych, O. Harasymchuk, Y. Kostiv: Poisson pulse sequence generators based upon modified Geffe generators.....	17
Z. Kokosiński: A parallel dynamic programming algorithm for unranking set partitions.....	29
K. Schiﬀ: Ant colony optimization algorithm for the 0-1 knapsack problem.....	39
I. Verbenko, R. Tkachenko: Gantry and bridge cranes neuro-fuzzy control by using neural-like structures of geometric transformations.....	53
G. Nowakowski: Effective use of lambda expressions in the new C++11 standard.....	69
Z. Kokosiński, S. Wójcik: A comparison of SW/HW implementations of stream cipher encoders	83
A. Buruzs, M.F. Hatwagner, L.T. Kóczy: Modeling integrated sustainable waste management systems by fuzzy cognitive maps and the system of systems concept.....	93

TREŚĆ

M. Zając: Sterowanie napędu elektrycznego metodą dynamiki odwrotnej	3
V. Maksymovych, O. Harasymchuk, Y. Kostiv: Generatory ciągów impulsów Poissona oparte na zmodyfikowanych generatorach Geffego	17
Z. Kokosiński: Równoległy algorytm programowania dynamicznego do konwersji liczb porządkowych w podziały zbioru	29
K. Schiﬀ: Algorytmy mrówkowe dla 0-1 problemu plecakowego	39
I. Verbenko, R. Tkachenko: Neuronowo-rozmyte sterowanie suwnicami bramowymi i pomostowymi z użyciem struktur przekształceń geometrycznych o charakterze neuronowym ...	53
G. Nowakowski: Efektywne wykorzystanie wyrażeń lambda w nowym standardzie C++11	69
Z. Kokosiński, S. Wójcik: Porównanie implementacji programowych i sprzętowych szyfratorów strumieniowych	83
A. Buruzs, M.F. Hatwagner, L.T. Kóczy: Modelowanie zintegrowanych zrównoważonych systemów gospodarowania odpadami za pomocą rozmytych map poznawczych oraz koncepcji systemu systemów	93

