

Polska Akademia Nauk Oddział w Gdańsku

Komisja Informatyki

# **Metody Informatyki Stosowanej**

Nr 1/2008 (Tom 13)

Szczecin 2008

**Metody Informatyki Stosowanej**  
**Kwartalnik Komisji Informatyki Polskiej Akademii Nauk Oddział w Gdańsku**

Komitet Naukowy:

Przewodniczący:

*prof. dr hab. inż. Henryk Krawczyk, czł. koresp. PAN, Politechnika Gdańska*

Członkowie:

*prof. dr hab. inż. Michał Biatko, czł. rzecz. PAN, Politechnika Koszalińska*

*prof. dr hab. inż. Janusz Kacprzyk, czł. koresp. PAN, Instytut Badań Systemowych PAN*

*prof. dr hab. inż. Leszek Rutkowski, czł. koresp. PAN, Politechnika Częstochowska*

*prof. dr inż. Jerzy Soldek, Politechnika Szczecińska*

*prof. dr hab. inż. Andrzej Straszak, Instytut Badań Systemowych PAN*

Recenzenci:

*prof. dr hab. inż. Marian Adamski, Uniwersytet Zielonogórski*

*prof. dr hab. inż. Zbigniew Banaszak, Politechnika Koszalińska*

*prof. dr hab. inż. Włodzimierz Bielecki, Politechnika Szczecińska*

*prof. dr hab. inż. Ryszard Budziński, Uniwersytet Szczeciński*

*prof. dr hab. inż. Georgy Kukharev, Politechnika Szczecińska*

*dr hab. Małgorzata Łatuszyńska, prof. US, Uniwersytet Szczeciński*

*prof. dr hab. inż. Andrzej Piegat, Politechnika Szczecińska*

*prof. dr hab. inż. Orest Popov, Politechnika Szczecińska*

*prof. dr inż. Jerzy Soldek, Politechnika Szczecińska*

*dr hab. inż. Antoni Wiliński, prof. PS, Politechnika Szczecińska*

Redaktor naczelny:

*Antoni Wiliński*

Sekretarz redakcji:

*Piotr Czapiewski*

ISSN 1898-5297

ISBN 978-83-925803-2-4

Wydawnictwo:

Polska Akademia Nauk Oddział w Gdańsku

Komisja Informatyki

Adres kontaktowy: ul. Żołnierska 49 p. 104, 71-210 Szczecin

Druk: Pracownia Poligraficzna Wydziału Informatyki Politechniki Szczecińskiej.

Nakład 500 egz.

# Spis treści

OD REDAKCJI . . . . .	5
<i>Ewa Adamus</i> KIERUNKOWE ZBIORY PODOBIENSTWA A PROBLEM NIEKOMPLETNOŚCI DANYCH . . . . .	7
<i>Dariusz Burak</i> KOMPILATOR ZRÓWNOLEGLAJĄCY WYSPECJALIZOWANY NA ALGORYTMY SZYFROWANIA . . . . .	17
<i>Piotr Czapiewski</i> ZASTOSOWANIE ZBIORÓW PRZYBLIŻONYCH I SYSTEMU ROSE2 DO INDUKCJI REGUŁ DECYZYJNYCH NA POTRZEBY SYSTEMU AUTOMATYCZNEGO OBROTU PARĄ WALUTOWĄ . . . . .	25
<i>Larisa Dobryakova, Evgeny Ochin</i> WBUDOWANIE WIADOMOŚCI W DANE MEDIALNE NA PODSTAWIE ANALIZY DWÓCH NAJMNIEJ ZNACZĄCYCH BITÓW KONTENERA . . . . .	39
<i>Stawomir Jaszczak, Krzysztof Gorący, Waldemar Pankanin</i> MODUŁ EKSPERTYZ W MULTIMEDIALNYM ATLASIE WAD LAMINATÓW . . . . .	47
<i>Ekaterina Kamenskaya, Georgy Kukharev</i> RECOGNITION OF PSYCHOLOGICAL CHARACTERISTICS FROM FACE . . . . .	59
<i>Jerzy Korostil, Łukasz Nozdrzykowski</i> OPRACOWANIE ROZPROSZONEGO KLUCZA STEGANOGRAFICZNEGO . . . . .	75
<i>Marcin Korzeń</i> PORÓWNANIE WŁASNOŚCI HEURYSTYK DO WYZNACZANIA REDUKTÓW MINIMALNYCH . . . . .	83
<i>Dorota Majorkowska-Mech, Alexandr Tariov</i> ALGORYTMY WIELOPOZIOMOWEJ DEKOMPOZYCJI I REKONSTRUKCJI FALKOWOPODOBNEJ DANYCH 2D . . . . .	93
<i>Kesra Nermend</i> PROBLEM OBIEKTÓW NIETYPOWYCH W BADANIACH TAKSONOMICZNYCH . . . . .	105
<i>Adam Nowosielski</i> IDENTYFIKACJA CZŁOWIEKA NA PODSTAWIE ZESTAWU OBRAZÓW TWARZY POCHODZĄCYCH Z SEKWENCJI VIDEO . . . . .	113
<i>Marcin Pluciński</i> POSZUKIWANIE OPTYMALNEJ STRATEGII EKSPLOKACJI Z ZASTOSOWANIEM UCZENIA ZE WZMOCNIENIEM . . . . .	127
<i>Magdalena Szaber</i> ROZPOZNAWANIE TWARZY DLA OBRAZÓW WIDZIALNYCH I PODCZERWIENI Z ZASTOSOWANIEM FUNKCJI TRANSFORMACJI W PRZESTRZENI CECH . . . . .	139
<i>Alexandr Tariov</i> STRATEGIE RACJONALIZACJI OBLICZEŃ PRZY WYZNACZANIU ILOCZYNÓW MACIERZOWO-WEKTOROWYCH . . . . .	147

---

*Antoni Wiliński, Piotr Czapiewski*

BADANIA MOŻLIWOŚCI WYBORU OPTYMALNEJ ZMIENNEJ OBSERWOWANEJ JAKO  
ZMIENNEJ PREDYKCYJNEJ W ZBIORZE DANYCH O RYNKACH FINANSOWYCH . . . 159

*Przemysław Klęsk*

O UKRYTYCH MODELACH MARKOWA I ICH ZASTOSOWANIU DO ROZPOZNAWANIA  
CIĄGÓW ZNAKÓW PISMA ODRĘCZNEGO . . . . . 175

*Luiza Kuśkiewicz, Jarosław Wątróbski*

INTEGRACJA METOD AHP I SWOT W FORMUŁOWANIU STRATEGII DZIAŁANIA  
PRZEDSIĘBIORSTWA . . . . . 191

*Dariusz Rzońca, Jan Sadolewski, Andrzej Stec, Zbigniew Świder, Bartosz  
Trybus, Leszek Trybus*

ŚRODOWISKO PROGRAMISTYCZNE DLA ROZPROSZONYCH MINISYSTEMÓW  
KONTROLNO-POMIAROWYCH . . . . . 199

# Od Redakcji

Szanowni Państwo

Oddajemy w Państwa ręce kolejny numer kwartalnika *Metody Informatyki Stosowanej*. Jest to pierwszy numer w 2008 roku, a trzeci od momentu przekazania tytułu pod egidę Komisji Informatyki Polskiej Akademii Nauk Oddział w Gdańsku. Pragniemy przy tej okazji poinformować, że 29 lutego 2008 r. pismo nasze trafiło na listę czasopism punktowanych Ministra Nauki i Szkolnictwa Wyższego, uzyskując 4 punkty.

Serdecznie zapraszamy do współpracy Autorów z całej Polski, zainteresowanych publikowaniem prac z zakresu szeroko pojętej informatyki stosowanej, a w szczególności z następujących dziedzin:

- metody sztucznej inteligencji,
- komputerowe modelowanie i symulacja,
- klasyfikacja i analiza danych,
- rozpoznawanie wzorców,
- grafika komputerowa i przetwarzanie obrazów,
- inżynieria systemów informatycznych,
- inżynieria oprogramowania,
- architektura systemów komputerowych,
- przetwarzanie równoległe i rozproszone,
- bezpieczeństwo systemów informatycznych,
- technologie internetowe,
- problemy społeczeństwa informacyjnego.

Wszystkie zamieszczane publikacje poddane zostają wnikliwej recenzji dokonywanej przez samodzielnych pracowników naukowych reprezentujących kilka krajowych ośrodków akademickich. Szczegółowe informacje dotyczące wymogów formalnych oraz trybu nadsyłania prac znajdują Państwo w witrynie <http://pan.wi.ps.pl>.

Życząc udanej lektury wyrażamy nadzieję, że stanie się ona dla Państwa inspiracją do intensywnej aktywności publikacyjnej.

*Antoni Wiliński*

Redaktor Naczelny



# Kierunkowe zbiory podobieństwa a problem niekompletności danych

Ewa Adamus

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*The problem of the incomplete data is quite common especially in the case of the actual measurement samples. In this connection, it has been vastly commented in the literature, especially in the rough set theory. The rough set theory was meant as a tool for imprecise and inconsistent information systems. The aim of this work is to supplement the incomplete data relying on the relations designed to this problem (similarity and tolerance relation). Basing on the opposite information to the incomplete object we know the area of permitted values for this object. The method proposed in the article works on the assumption that we possess with the opposite information to the supplemented sample in our information system.*

## Słowa kluczowe:

*rough set theory, tolerance relation, similarity relation, opposite information, directional classes of similarity*

## 1. Wstęp

W rzeczywistych zbiorach danych bardzo często pojawia się problem niekompletności próbek pomiarowych. Przyczyny tego mogą być różnorakie, zupełnie przypadkowe, zamierzone (bardzo często spotykane w ankietach), techniczne (związane z możliwościami pomiarowymi aparatury) bądź inne.

Formalnie niekompletnością określamy brak wartości w pewnych atrybutach wektora wejściowego próbki. Zakładamy, że pojęcie to nie odnosi się do wektora z atrybutami wyjściowymi:

**Definicja 1.1.** *Dla zbioru  $A$  atrybutów warunkowych, zdefiniowanego na uniwersum przykładów  $U$ , mówimy, że **obiekt  $x \in U$  jest niekompletny**, jeżeli dla przynajmniej jednego atrybutu warunkowego  $a$  ( $a \in A$ ), obiekt ten nie ma zdefiniowanej wartości:  $f(a, x) = *$ <sup>1</sup>, przy czym wartość atrybutu decyzyjnego  $d$  jest zawsze zdefiniowana ( $f(d, x) \neq *$ ).*

*Gdzie:  $f(a, x)$  – funkcja informacyjna, taka że  $\forall a \in A, x \in U f(a, x) \in V_a$ ,  $V_a$  – dziedzina atrybutu  $a$ .*

---

<sup>1</sup> m.in. w pracy [6] przyjęto odpowiednio oznaczenia: „?” do istotnych oraz „\*” do nieistotnych brakujących wartości

Brak wartości w pewnych atrybutach wektora wejściowego próbki może być różnie interpretowany. W pracy [11] nieznanne wartości atrybutu podzielone zostały na dwie grupy:

- **niedostępne** wartości,
- **brakujące** wartości.

Do pierwszej kategorii zaliczyć należy sytuacje, kiedy wartość określonego atrybutu jest „ostatecznie niedostępna” (przykładowo nie mamy fizycznej (technicznej) możliwości zarejestrowania pewnych pomiarów). Drugą grupę należy interpretować jako wartości, które obiekt mógłby posiadać, lecz z różnych przyczyn nie zostały zarejestrowane. W związku z tym, że przyczyny nieznanności wartości dla pewnych atrybutów mogą być różne, stosuje się odpowiednio odmienne podejścia do tego problemu.

W literaturze dotyczącej problemu niekompletności danych przyjęto formalny podział tego zjawiska. Ponieważ wiele metod bazuje na pewnych założeniach co do rodzaju niekompletności danych, poniżej przedstawiono ogólnie przyjętą klasyfikację [9].

Jeżeli zdefiniujemy macierz  $M$  indyktorów niekompletności, o tym samym rozmiarze co tabela z danymi  $D$ , o wartościach elementów równych 1, dla zaobserwowanych wartości w  $D$  oraz 0 dla wartości brakujących. Wówczas podział rodzaju niekompletności danych, z formalnego punktu widzenia, wygląda następująco [8]:

- **MCAR** (*missing completely at random*) –  $M$  jest niezależne od  $D$  :  $P(M|D) = P(M)$ , czyli na podstawie części zaobserwowanej nie jesteśmy w stanie dokonać prawidłowej predykcji wartości brakującej. Prawdopodobieństwo, że pewien przypadek (przykład) nie posiada wartości dla określonego atrybutu, nie zależy od wartości zdefiniowanych tego przypadku, ani od części wektora bez wartości.
- **MAR** (*missing at random*) – prawdopodobieństwo braku wartości może zależeć od  $D_{obs}$  ale musi być niezależne od  $D_{mis}$ . Formalnie  $M$  jest niezależne od  $D_{mis}$  :  $P(M|D) = P(M|D_{obs})$ .
- **NI** (*nonignorable*) lub **NMAR** (*not missing at random*). Prawdopodobieństwo braku wartości zależy od nie zaobserwowanej części. Formalnie  $M$  jest zależne od  $D$ , czyli  $P(M|D)$  nie upraszcza się.

Najprostszym podejściem, w przypadku stwierdzenia braku wartości w pewnych atrybutach wektora wejściowego próbki jest usunięcie takich próbek ze zbioru danych. Taki sposób postępowania nie ma znacząco ujemnego skutku, jeżeli dysponujemy licznym, w miarę dobrze zdefiniowanym – w przestrzeni rozważań – zbiorem danych, w którym dodatkowo niekompletne dane nie stanowią istotnej części naszego zbioru. W przeciwnym przypadku możemy oszacować brakujące wartości lub zmodyfikować metodę tak, aby uwzględniała niekompletne dane. Pierwszy przypadek, będzie stanowił wówczas, jeden z etapów wstępnego przetwarzania danych. Generalnie w literaturze spotkać możemy następujący podział metod postępowania z problemem niekompletności [5]:



1. Metody sekwencyjne (*preprocessing methods*).
  - a) Dane z brakującymi wartościami nie są uwzględniane w analizie danych (*complete – case analysis, listwise deletion, casewise deletion*).
  - b) Szacowanie brakujących wartości – zazwyczaj w ramach wstępnego przygotowania danych.
2. Metody równoległe („*parallel methods*”).

W przypadku metod sekwencyjnych, niekompletny zbiór danych przekształcały jest w wariant kompletny, po czym stosowana jest odpowiednia metoda obliczeniowa.

Druga kategoria – metody równoległe – niekompletne próbki pomiarowe uwzględniane są we właściwym procesie obliczeniowym. Metody są odpowiednio dostosowywane do możliwości braku wartości w pewnych atrybutach wektora wejściowego próbki – wykorzystywana jest wyłącznie zdefiniowana informacja w niekompletnej próbce. W tym przypadku chodzi o takie dostosowanie istniejących metod, aby „tolerowały” one niekompletne próbki. Podstawową zaletą takiego podejścia – w stosunku do poprzedniej kategorii – jest fakt, że nie wprowadzamy do systemu dodatkowego błędu związanego z szacowaniem brakujących wartości.

## 2. Warunkowe uzupełnianie niekompletnych danych, dopełnieniami klas podobieństwa

W przypadku teorii zbiorów przybliżonych, nie możemy dokonać jednoznacznej klasyfikacji tej teorii do jednej z wymienionych kategorii postępowania z niekompletnością. W tym aspekcie teoria ta wyróżnia się na tle pozostałych metod (patrz [1]). Niekompletna próbka bowiem, w zależności od kontekstu (wzajemnych relacji z pozostałymi obiektami w bazie wiedzy), zostanie zaakceptowana z całkowitą pewnością do dalszej analizy (znajdzie się w dolnym przybliżeniu danego konceptu decyzyjnego), bądź przypuszczalnie (znajdzie się w obszarze granicznym). W przypadku uwzględnienia niekompletnego obiektu wnosi on jedynie posiadaną informację (bez uzupełniania) – akceptowane są wyłącznie zdefiniowane wartości niekompletnego obiektu. W sytuacji, gdy niepełna próbka znajdzie się w obszarze granicznym klasy decyzyjnej stanie się tym samym obiektem wątpliwym, co do przynależności do konceptu decyzyjnego. Celem niniejszej pracy jest uwzględnienie kolejnego wariantu postępowania z niekompletnością, czyli warunkowego uzupełniania niekompletnych danych. Warunkowego, gdyż wymogiem uzupełniania jest obecność spójnej informacji przeciwnej (reprezentującej inne koncepty decyzyjne) do obiektu uzupełnianego.

Zakładamy, że dla niekompletnego obiekt  $x \in U$ , zaklasyfikowanego do uzupełniania, mamy określoną klasę obiektów, do których  $x$  jest podobny:  $S^{-1}(x)$ . W celu uzupełnienia  $x$  uwzględniamy informację przeciwną do reprezentowanej przez zbiór  $S^{-1}(x)$ , czyli jego dopełnienie (gdzie  $x$  to hipotetyczny, niekompletny obiekt kwalifikujący się do uzupełnienia).  $S^{-1}(x)$  można interpretować jako zbiór stanowiący koniunkcję obiektów, do których  $x$  jest podobny (mówimy, że obiekt jest podobny do obiektu  $x_1$  i do  $x_2, \dots, x_n$ , gdzie  $n$  – liczność zbioru  $S^{-1}(x)$ ).

W związku z tym, informacją przeciwną do  $S^{-1}(x)$  ( $\overline{S^{-1}(x)}$ ) – zgodnie z prawami de Morgana – będzie alternatywa negacji poszczególnych obiektów:

$$\bigwedge_{x_i \in S^{-1}(x)} \overline{S^{-1}(x)} = \overline{x_1} \vee \overline{x_2} \vee \dots \vee \overline{x_n}. \quad (1)$$

Ogólnie oraz wstępnie przedstawione założenie, co do możliwości wykorzystania informacji przeciwnej, w celu ewentualnego uzupełnienia obiektu niekompletnego, będzie podstawą do dalszych rozważań. Podejście to może być stosowane na etapie uczenia nadzorowanego, a w analizowanym przypadku do indukcji efektywniejszych reguł decyzyjnych.

Poniżej przedstawiono ogólny algorytm postępowania z niekompletnym obiektem  $x$  przy założeniu, że dla tablicy decyzyjnej  $DT = (U, A \cup \{d\})$  oraz obiektu  $x \in U$  mamy określoną klasę obiektów, do których  $x$  jest podobny:  $S^{-1}(x)$ . Dodatkowo  $Sr^{-1}(x) = S^{-1}(x) \setminus \{x\}$  oraz  $Sr_K^{-1}(y)$  jest kierunkowym zbiorem podobieństwa dla  $y \in Sr^{-1}(x) \wedge S^{-1}(y) = \{y\}$ . Opisowy schemat przyjętego algorytmu postępowania z niekompletną próbką, można przedstawić w następujący sposób (dokładniejsze informacje dostępne m.in. w [3, 2]):

1. J. grupa obiektów, do których podobny jest element niekompletny  $x$  ( $S^{-1}(x)$ ), reprezentuje wyłącznie jeden koncept  $X$ , wówczas  $x$  wchodzi w skład dolnego przybliżenia zbioru  $X$ . : **akceptacja wyłącznie zdefiniowanej informacji w niepełnej próbce.**
2. W przeciwnym przypadku, analizujemy zbiór  $Sr^{-1}(x)$ :
  - 2.1. J. zbiór  $Sr^{-1}(x)$  w całości reprezentuje przeciwny koncept decyzyjny, wówczas **uzupełniamy dopełnieniem klasy podobieństwa.**
  - 2.2. W przeciwnym przypadku, analizujemy składowe zbiory  $Sr^{-1}(x)$  – zbiory  $Sr_K^{-1}(y)$  – kierunkowe klasy podobieństwa:
    - 2.2.1. j. w ramach przynajmniej jednej kierunkowej klasy podobieństwa brak jednolitości, co do przynależności do konceptu decyzyjnego, wówczas : **x znajdzie się w obszarze granicznym klas decyzyjnych.**
    - 2.2.2. w przeciwnym przypadku : **uzupełniamy dopełnieniem klasy podobieństwa.**

Na uzupełnienie niekompletnego obiektu  $x$  składa się dopełnienie klasy obiektów, do których niekompletny jest podobny ( $S^{-1}(x)$ ), za wyjątkiem takich obiektów należących do klasy relacji tolerancji obiektu  $x$  ( $T(x)$ ), na podstawie których możemy dokonać jednoznacznej klasyfikacji ( $S^{-1}(y) \subseteq -Y \vee S^{-1}(y) \subseteq Y$ ).

$$C(S^{-1}(x)) \setminus \{y : y \in T(x) \wedge (S^{-1}(y) \subseteq -Y \vee S^{-1}(y) \subseteq Y)\}. \quad (2)$$

### 3. Eksperymenty

Przedstawiona metoda warunkowego uzupełniania niekompletnych danych dopełnieniami klas podobieństwa, została zweryfikowana dla rzeczywistych próbek pomiarowych. W tym celu wykorzystano zbiór danych ze znanego repozytorium

UCI [4]. W pierwszym etapie obliczeń zastosowano podejście dla niekompletnych próbek, o brakujących wartościach, dla których faktyczne wartości istnieją, lecz nie zostały zaobserwowane, bądź zarejestrowane. W tym celu zbiór o nazwie 'diabetes' został zdekompetywany w stopniu od 2 do 30 procent, z krokiem równym 2, wg schematu przedstawionego w dalszej części referatu. Analizowany zbiór składa się z 768 próbek opisanych przy pomocy 8 atrybutów warunkowych o ciągłej reprezentacji. Atrybut decyzyjny posiada dwie klasy.

We wszystkich przypadkach, weryfikacji metod dokonano korzystając z procedury *k-krotnej walidacji krzyżowej*, dla  $k = 10$ .

W celu zdekompetywania próbek pomiarowych zbioru 'diabetes' posłużono się schematem zgodnym z taksonomią rodzaju niekompletności przyjętą w literaturze oraz opisaną już we wcześniejszej części referatu<sup>2</sup>.

W celu wygenerowania reguł zastosowano algorytm LEM2 (Learning from Examples Module, version 2) [7], polegający na wyliczaniu lokalnych pokryć, par wartości atrybutów, tworzących minimalne kompleksy. Dodatkowo, algorytm ten wykorzystuje elementy teorii zbiorów przybliżonych, gdyż reguły generowane są na podstawie przybliżeń zbiorów decyzyjnych. Uzasadnienie zastosowania algorytmu w pracy, wynika z możliwości swobodnego zastosowania wybranej relacji (podobieństwa bądź tolerancji), w przypadku obecności niekompletnych danych.

Celem eksperymentów obliczeniowych jest porównanie wydajności klasyfikatorów regułowych wygenerowanych przy zastosowaniu klasycznej – niesymetrycznej relacji podobieństwa bez uzupełniania, z wariantem metody uwzględniającym uzupełnianie niekompletnych danych, czyli metodą warunkowego uzupełniania niekompletnych danych dopełnieniami klas podobieństwa.

W przypadku zastosowania do klasyfikacji obiektu, nieuporządkowanego zbioru reguł może wystąpić jeden z następujących scenariuszy [10]:

1. Obiekt jest pokrywany przez jedną regułę.
2. Do klasyfikacji obiektu pretenduje wiele reguł.
3. Część warunkowa obiektu nie zgadza się z żadną z reguł.

Dla sytuacji 1, klasyfikacja obiektu jest jednoznacznie określona przez regułę, której część warunkowa zgadza się z obiektem. Jeżeli dodatkowo występuje zgodność co do wartości atrybutu decyzyjnego, obiekt jest poprawnie zaklasyfikowany.

<sup>2</sup> Jeżeli zdefiniujemy macierz  $M$  indyktorów niekompletności, o tym samym rozmiarze co tabela z danymi  $D$ , o wartościach elementów równych 1, dla zaobserwowanych wartości w  $D$  oraz 0 dla wartości brakujących. Dodatkowo, jeżeli  $pr =$  stopień niekompletności/100;  $n$  – liczba pseudo-losowa, o rozkładzie równomiernym, z zakresu  $(0, 1)$ ;  $i, j$ : nr wiersza oraz kolumny macierzy  $M$ , wówczas dla niekompletności typu:

- *MCAR*: j.  $n < pr$  to  $M_{ij} = 0$ , w przeciwnym przypadku  $M_{ij} = 1$ ,
- *MAR*: losowo wybrano: 1 atrybut, dla którego zastosowano schemat *MCAR*; 4 atrybuty:  $\{j1, j2, j3, j4\}$ , dla których: j.  $X_{i,j1} < 0.7$  i  $n < pr$  wówczas  $M_{i,j2} = 0$ ; j.  $X_{i,j3} < 0.3$  oraz  $n < pr$  wówczas  $M_{i,j4} = 0$ ,
- *NI*: losowo wybrano: 2 atrybuty, dla których zastosowano schemat *MAR*; 1 atrybut – *MCAR*; 3 atrybuty:  $\{j4, j5, j6\}$ , dla których: j.  $n < pr$  oraz  $Y_i = 0$  to  $M_{i,j4} = 0$ , j.  $Y_i = 1$  to  $M_{i,j5} = 0$  oraz  $M_{i,j6} = 0$ .

W przypadku dwóch pozostałych sytuacji, w literaturze można spotkać wiele opracowań metod przeznaczonych do radzenia sobie z tym problemem. W pracy [10] przedstawione są niektóre z nich. Natomiast w rozprawie przyjęto następujący, uproszczony schemat postępowania:

1. Jeżeli część warunkowa obiektu zgadza się z opisem wielu reguł i wszystkie te reguły wskazują na tę samą klasę decyzyjną co obiekt, wówczas obiekt uznawany jest za poprawnie zaklasyfikowany. W przeciwnym przypadku, jeżeli część decyzyjna przynajmniej jednej reguły nie zgadza się z klasą obiektu, wówczas obiekt uznawany jest za błędnie zaklasyfikowany.
2. Jeżeli część warunkowa obiektu nie zgadza się z żadną z reguł, wówczas obiekt uznawany jest za źle sklasyfikowany.

W celu przeprowadzenia oceny zastosowanych reguł, wygenerowanych na podstawie próbek treningowych bez uzupełniania oraz z warunkowym uzupełnianiem, zastosowano następujące miary:

1. **Średnia liczba poprawnych reguł** ( $\bar{R}_c$ ) wyliczona jako:

$$\bar{R}_c = \frac{\text{liczba poprawnych reguł}}{\text{całkowita liczba reguł}}, \quad (3)$$

przy czym reguła uznawana jest za „poprawną”, jeżeli część decyzyjna wszystkich pokrywanych przez nią przykładów zgadza się z klasą wskazywaną przez samą regułę. Wyniki dla etapu testowania przedstawia rys. 1 – dla indukcji reguł oraz klasyfikacji danych testujących z użyciem relacji podobieństwa bez uzupełniania oraz z uzupełnianiem.

2. **Średnia liczba poprawnie zaklasyfikowanych próbek** ( $\bar{S}_c$ ) obliczona jako:

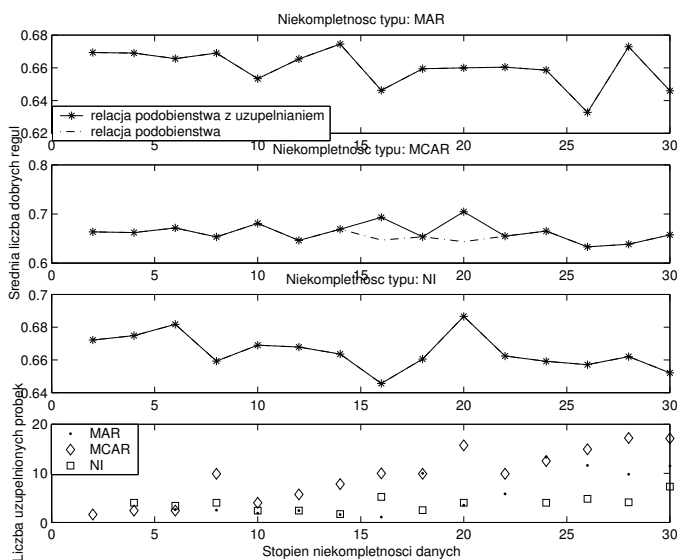
$$\bar{S}_c = \frac{\text{liczba prawidłowo zaklasyfikowanych próbek}}{\text{całkowita liczba próbek}}. \quad (4)$$

Miara ta w literaturze często określana jest również jako *trafność klasyfikowania*, *dokładność* lub *sprawność* [10]. Wyniki obliczeń umieszczono na diagramie 2 – dla indukcji reguł oraz klasyfikacji danych testujących z użyciem relacji podobieństwa bez uzupełniania oraz z uzupełnianiem.

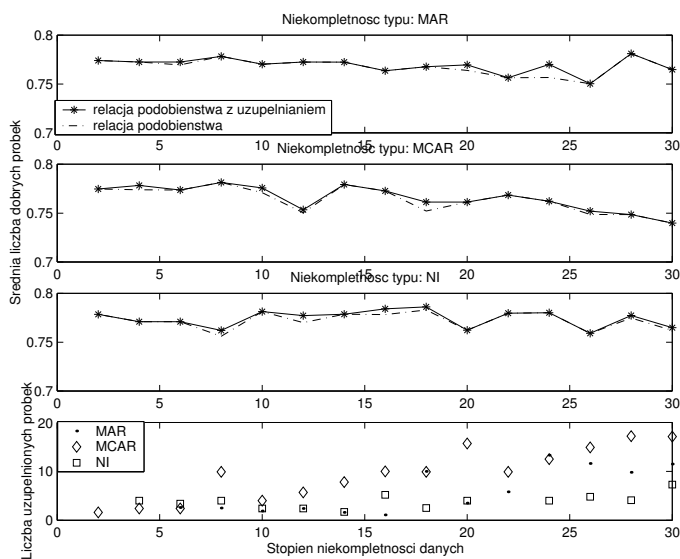
## 4. Wnioski

W pracy przedstawiona została metoda warunkowego uzupełniania niekompletnych danych dopełnieniami klas podobieństwa. Wnioski dotyczące metody:

1. W przypadku zastosowanego sposobu uzupełniania nie obciążamy istniejących obiektów rzeczywistych, elementami sprzecznymi. Co prawda, teoria zbiorów przybliżonych dostarcza narzędzi przeznaczonych do radzenia sobie z tym problemem, jednak w sytuacji sprzeczności powstałej w wyniku uzupełniania nie wiemy, który obiekt jest rzeczywistym, pomiarowym, a który powstałym w wyniku uzupełniania.



Rysunek 1. Średnia liczba poprawnych reguł dla etapu testowania oraz statystyka uzupełnionych próbek dla relacji podobieństwa z uzupełnianiem i bez uzupełniania. Zbiór danych 'diabetes'.



Rysunek 2. Średnia liczba poprawnie zaklasyfikowanych próbek dla etapu testowania oraz statystyka uzupełnionych próbek dla relacji podobieństwa z uzupełnianiem i bez uzupełniania. Zbiór danych 'diabetes'.

2. Przy uzupełnianiu dopełnieniami klas podobieństwa nie tracimy informacji o unikalnych obiektach niekompletnych oraz obiektach niekompletnych, co prawda nie unikalnych, ale na podstawie których można dokonać jednoznacznej klasyfikacji (czyli podobnych wyłącznie do obiektów tej samej – co niekompletny – klasy decyzyjnej).
3. Warunkowe uzupełnianie dopełnieniami klas podobieństwa, zwiększa dokładność przybliżenia danego zbioru (współczynnik przybliżenia). Wynika to z faktu, że zbiór obiektów, powstałych w wyniku takiego uzupełniania, całkowicie zawiera się w dolnym przybliżeniu rozważanego zbioru.
4. Uzupełnianie wybranych, niekompletnych danych wymaga znajomości wartości ich klasy decyzyjnej. Z tego względu – omówioną metodę – można stosować dla etapu uczenia nadzorowanego, w celu indukcji efektywniejszego klasyfikatora.
5. Mankamentem metody jest jej wrażliwość na rozmiar wektora wejść systemu informacyjnego. Przy znacznej liczbie atrybutów warunkowych proces obliczeniowy może ulec wydłużeniu, ze względu na potrzebę obliczania dopełnień klas podobieństwa. Jednak przy obecnej technologii, czynnik ten nie powinien mieć istotnego znaczenia, zważywszy powszechność zjawiska jakim jest problem niekompletnych danych oraz potrzebę opracowywania narzędzi do radzenia sobie z tym problemem. W przypadku metody rozważanej w referacie, alternatywą dla ewentualnego wzrostu czasu obliczeń jest możliwość uzyskania dodatkowych informacji na podstawie niekompletnych danych pomiarowych.

## Bibliografia

- [1] Adamus E. *Przegląd metod stosowanych do badań nad niekompletnymi danymi pomiarowymi*. VIII Sesja Informatyki 2003, s. 387-395
- [2] Adamus E. *Metoda warunkowego uzupełniania niekompletnych danych dopełnieniami klas podobieństwa*. Rozprawa Doktorska, Politechnika Szczecińska, Wydział Informatyki, Instytut Sztucznej Inteligencji i Metod Matematycznych, 2007
- [3] Adamus E. *Predicting incomplete data on the basis of non symmetric similarity relation*. Advances in Information Processing and Protection, Springer-Verlag 2007
- [4] Blake C. L., Newman D. J., Hettich S., Merz C. J. *UCI Repository of machine learning databases*. [online] <http://www.ics.uci.edu/~mllearn/MLRepository.html>, University of California, Irvine, Dept. of Information and Computer Sciences, 1998
- [5] Grzymala J. W., Grzymala W. J. *Handling missing attribute values*. W: Maimon O., Rokach L. *Data Mining and Knowledge Discovery Handbook*. Springer US 2006, s. 37-57
- [6] Grzymala-Busse J. W. *Characteristic Relations for Incomplete Data: A Generalization of the Indiscernibility Relation*. Proceedings of the RSTC, the Fourth International Conference on Rough Sets and Current Trends in Computing, Springer-Verlag, 2004, Lecture Notes in Artificial Intelligence 3066, s. 244-253
- [7] Grzymala-Busse J. W. *Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction*. Transactions on Rough Sets, Springer-Verlag 2004
- [8] King G., Honaker J., Joseph A. *Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation*. American Political Science Review, vol. 95, 2001, s. 49-69

- [9] Little R. J., Rubin D. B. *Statistical Analysis with Missing Data, Second Edition*. John Wiley and Sons 2002
- [10] Stefanowski J. *Algorytmy indukcji reguł decyzyjnych w odkrywaniu wiedzy*. Rozprawa habilitacyjna, Wydawnictwo Politechniki Poznańskiej, Seria Rozprawy nr 361., 2001
- [11] Stefanowski J., Tsoukiàs A. *Incomplete information tables and rough classification*. Int. Journal of Computational Intelligence, 2001





# Kompilator zrównoleglający wyspecjalizowany na algorytmy szyfrowania

Dariusz Burak

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*Implementation issues of a parallelizing compiler based on an automatic parallelizing method of selected encryption algorithms are described. The compiler is devoted for parallelizing encryption algorithms written in the C language. The implementation of the compiler is based on the following well-known tools: Petit and OpenMP API. Petit is applied for finding data dependency relationships in the source code. The OpenMP standard is chosen for representing parallelism of encryption algorithms.*

## Słowa kluczowe:

*parallelizing compiler, loop transformations, OpenMP, data dependence analysis, encryption algorithms*

## 1. Wprowadzenie

W dzisiejszych czasach problem szybkiego tworzenia równoległych implementacji algorytmów szyfrowania na podstawie sekwencyjnych kodów źródłowych jest bardzo istotny z punktu widzenia użytkowników sieci komputerowych. Wraz z rozwojem Internetu wzrasta również ilość informacji przesyłanych poprzez statystycznego użytkownika sieci. Wymusza to działania mające na celu skrócenie czasu szyfrowania i deszyfrowania danych. Zważywszy na coraz większą powszechność technologii wieloprocesorowej ze wspólną pamięcią oraz technologii wielordzeniowej istnieje potrzeba efektywnego wykorzystania ich mocy obliczeniowych. Naturalnym sposobem wyzyskania ich mocy obliczeniowej jest zrównoleglenie czasochłonnych obliczeniowo algorytmów. Możliwe są tutaj dwa podejścia: zrównoleglenie ręczne – w oparciu o czasochłonną i skomplikowaną analizę zależności danych lub zrównoleglenie automatyczne, tzn. przebiegające w ramach kompilacji kodu źródłowego algorytmu sekwencyjnego. Z uwagi na szereg istotnych zalet (szybkość i prostota działania oraz relatywnie niski koszt wykonania) zdecydowano się na utworzenia kompilatora zrównoleglającego opartego na automatycznej metodzie zrównoleglenia. Jednakże jedną z głównych wad rozwiązań automatycznych istniejących obecnie jest stosunkowo niska jakość generowanego kodu równoległego, co wynika z ich uniwersalnego przeznaczenia- kompilatory zrównoleglające SUIF [1, 2], Polaris [3], Parafrase-2 [4] i inne. W trosce o zapewnienie maksymalnej efektywności działania kompilatora, tzn. uzyskania przyspieszenia pracy algorytmów szyfrowania na

poziomie zbliżonym do przyspieszenia maksymalnego (wyliczonego na podstawie prawa Amdahla [5]) i przy uwzględnieniu specyfiki oraz złożoności poszczególnych klas algorytmów szyfrowania (szyfry blokowe, strumieniowe, asymetryczne) oraz ich szczegółowych implementacji opracowano metodę automatycznego zrównoleglenia wybranych algorytmów szyfrowania [6], która zostanie zaimplementowana w ramach opracowywanego kompilatora zrównoleglającego. W celu utworzenia kompilatora niezbędne jest wykorzystanie istniejących narzędzi: programu Petito do analizy zależności danych oraz OpenMP API – standardu programowania równoległego oraz utworzenie nowych narzędzi – wynikających ze specyfiki opracowywanego zagadnienia.

## 2. Program Petit

Petit jest jednym z kluczowych komponentów projektu Omega przeznaczonego do analizy i transformacji programów, który powstał w Uniwersytecie Maryland [7]. Służy on do przeprowadzania analizy zależności oraz transformacji pętli zapisanych w języku Petit. Główne możliwości programu Petit są następujące [7]:

- analiza zależności oparta na wartości (ang. *value-based dependence analysis*) oraz na pamięci (ang. *memory based dependence analysis*),
- nieliniowa analiza zależności oparta na symbolach nieinterpretowanych funkcji (ang. *uninterpreted function symbols*),
- analiza skalarna,
- prywatyzacja zmiennych,
- rozszerzenie zmiennych skalarnych na zmienne tablicowe (ang. *scalar/array expansion*),
- możliwość zintegrowania analizy zależności wykonywanej za pomocą programu Petit w ramach kompilatorów zrównoleglających.

## 3. OpenMP API

OpenMP jest standardem korporacyjnym wspieranym przez producentów sprzętu, twórców oprogramowania oraz twórców aplikacji definiującym interfejs programowania aplikacji (ang. *Application Programming Interface (API)*) wykorzystywanym do bezpośredniego wyrażania równoległości w wielowątkowym modelu dla systemów ze wspólną pamięcią, zaimplementowanym dla różnych platform, którego cechą charakterystyczną jest ukończenie specyfikacji wymagań (obecnie aktualną wersją standardu jest wersja 2.5 z maja 2005r.) [8], [9].

OpenMP API rozszerza języki programowania – Fortran, C i C++ o [9]:

- dyrektywy kompilatora (bazujące na dyrektywie `#pragma` języków C i C++),
- funkcje biblioteki czasu wykonania (ang. *run-time library functions*),
- zmienne środowiskowe (ang. *environment variables*)

służące do wyrażania równoległości kodu źródłowego. Umieszczone w kodzie programu sekwencyjnego dyrektywy OpenMP, rozszerzające sekwencyjny model programowania o konstrukcje modelu SPMD (ang. *Single Program Multiple Data*),

w którym każdy procesor wykonuje ten sam program działając na własnym zestawie danych, konstrukcje podziału pracy (ang. *work-sharing constructs*), konstrukcje synchronizacji (ang. *synchronization constructs*) oraz konstrukcje środowiska danych (ang. *data environment constructs*) umożliwiają kompilatorowi wygenerowanie wielowątkowych programów wykonywalnych [9]. Zrównoleglenie kodu odbywa się w oparciu o model równoległego wykonywania programu – fork-and-join bazujący na operacjach rozwidlenia (ang. *fork*) oraz połączenia (ang. *join*). Wykonywanie programu rozpoczyna wątek pojedynczy zwany wątkiem nadrzędnym (ang. *master thread*), który w momencie napotkania konstrukcji równoległej tworzy grupę wątków (ang. *team of threads*). Każdy z wątków grupy wykonuje instrukcje znajdujące się w zasięgu dynamicznym rejonu równoległego, za wyjątkiem niektórych instrukcji podziału pracy (ang. *work-sharing constructs*). Po zakończeniu wykonywania rejonu równoległego następuje synchronizacja barierowa, a następnie wątek nadrzędny kontynuuje wykonywanie części sekwencyjnej programu (pozostałe wątki są niszczone), aż do momentu napotkania kolejnego rejonu równoległego (kod źródłowy może zawierać wiele rejonów równoległych, a rejonów równoległe mogą być zagnieżdżone, zatem program w czasie wykonywania może być wielokrotnie poddawany operacjom rozwidlenia i połączenia). Ilość tworzonych wątków może być różna dla różnych rejonów równoległych – możliwe jest wykorzystanie mechanizmu wątków dynamicznych (ang. *dynamic threads*) [9].

#### 4. Metoda automatycznego zrównoleglenia wybranych algorytmów szyfrowania

Metoda automatycznego zrównoleglenia wybranych algorytmów szyfrowania bazuje na wykorzystaniu kodu źródłowego danego algorytmu sekwencyjnego poprzez znalezienie fragmentów kodu potencjalnie równoległych, ich odpowiednie przekształcenie, a następnie zrównoleglenie. Zrównoleglenia algorytmów szyfrowania odbywa się na poziomie danych, gdyż ten poziom zrównoleglenia gwarantuje maksymalny stopień zużytkowania możliwości obliczeniowych komputerów wieloprocessorowych ze wspólną pamięcią [10]. Wymaga to odpowiednich transformacji kodu źródłowego algorytmów sekwencyjnych. Metoda automatycznego zrównoleglenia wybranych algorytmów szyfrowania obejmuje trzy etapy transformacji pętli przeznaczonych do zrównoleglenia [6]:

- transformacje wstępne – umożliwiające przeprowadzenie analizy zależności,
- transformacje usuwające, bądź redukujące istniejące zależności w kodzie źródłowym blokujące proces zrównoleglenia- zależności odwrotne, zależności po wyjściu, zależności proste oraz zależności sterowania,
- transformacje umożliwiające przetwarzanie równoległe pętli.

Opracowano trzy niezależne od siebie procedury zrównoleglenia pętli programowych występujących w implementacjach algorytmów szyfrowania oparte na przedstawionej powyżej kolejności transformacji kodu źródłowego [6]:

- Procedurę 1 – dla szyfrów blokowych pracujących trybie ECB,

- Procedurę 2 – dla pozostałych standardowych trybów pracy szyfrów blokowych,
- Procedurę 3 – dla szyfrów strumieniowych oraz szyfrów asymetrycznych.

Wszystkie procedury zrównoleglenia pętli zawierają transformacje składowe zgromadzone w ramach trzech grup transformacji. Grupę pierwszą tworzą transformacje wstępne, grupę drugą transformacje usuwające, bądź redukujące istniejące zależności w kodzie źródłowym algorytmów, a grupę trzecią transformacje umożliwiające przetwarzanie równoległe pętli programowych.

W celu uzyskania optymalnej wersji programu poprzez przebadanie wszystkich możliwych wariantów (ciągów) transformacji metoda automatycznego zrównoleglenia wybranych algorytmów szyfrowania wykorzystuje koncepcję kompilacji iteracyjnej, zaproponowaną przez Knijnenburga, Kisukiego i O’Boyle’a w [11, 12, 13]. Zastosowanie oryginalnych procedur zrównoleglenia pętli programowych skutecznie rozwiązuje problem znacznej pracochłonności wykonywania testowych wersji programu w ramach kompilacji iteracyjnej z uwagi na znaczne ograniczenie liczby możliwych wariantów (6 możliwości w przypadku Procedury 1, trzy możliwości w przypadku Procedury 2, dwie możliwości w przypadku Procedury 3 [6]).

## 5. Propozycja kompilatora oraz zestawu narzędzi wspierających automatyczne zrównoleglenie algorytmów szyfrowania

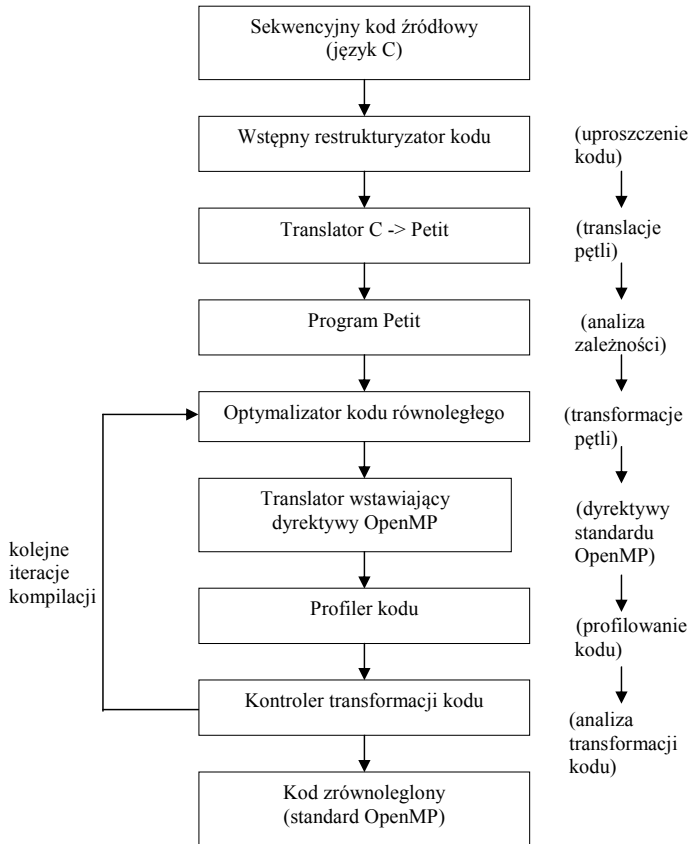
W skład zestawu narzędzi wspierających proces automatycznego zrównoleglenia algorytmów szyfrowania (Rys.1) wchodzi:

- wstępny restrukturyzator kodu,
- translator z języka C na język Petit,
- program Petit,
- optymalizator kodu równoległego,
- translator wstawiający dyrektywy standardu OpenMP,
- profiler kodu,
- kontroler transformacji kodu.

### 5.1. Wstępny restrukturyzator kodu

Zadaniem wstępnego restrukturyzatora kodu źródłowego zapisanego w języku C jest umożliwienie przeprowadzenia analizy zależności danych z zastosowaniem programu Petit. Z uwagi na specyfikę kodów źródłowych algorytmów szyfrowania oraz ograniczenia programu Petit do głównych zadań restrukturyzatora należy:

- konwersja zmiennych wskaźnikowych równoważnymi im zmiennymi tablicowymi,
- zastąpienie indeksowania tablic według stałych wartości na indeksowanie według zmiennych indeksowych pętli for (w ciałach pętli for),



Rysunek 1. Proces iteracyjnego zrównoleglenia z wyszczególnieniem zestawu narzędzi wspierających proces zrównoleglenia w ramach kompilatora zrównoleglającego wyspecjalizowanego na algorytmy szyfrowania

- zastąpienie kodu źródłowego zawierającego złożone struktury danych (struktura, unia) kodem zawierającym jedynie proste struktury danych,
- zastąpienie pętli typu do równoważnymi pętlami typu for,
- zastąpienie pętli typu while równoważnymi pętlami typu for,
- usunięcie z kodu źródłowego algorytmów instrukcji goto, switch, break, continue oraz return (ze środka ciała pętli programowych),
- zastąpienie wywołań funkcji składowych umieszczonych w pętlach programowych ciałami tychże funkcji.

## 5.2. Translator z języka C na język Petit

Zadaniem translatora jest zamiana kodu źródłowego poszczególnych pętli danego algorytmu szyfrowania zapisanego w języku C na równoważny kod zapisany w języku Petit (o składni zbliżonej do języka Fortran).

### 5.3. Program Petit

Program Petit (<http://www.cs.umd.edu/projects/omega/>) jest to narzędzie istniejące w domenie publicznej służące do automatycznego wykonywania analizy zależności pętli (krótka charakterystyka programu Petit – patrz pkt. 2).

Zadaniem programu Petit jest znalezienie istniejących relacji zależności dla poszczególnych pętli programowych występujących w kodzie źródłowym algorytmów szyfrowania.

### 5.4. Optymalizator kodu równoległego

Optymalizator kodu równoległego ma na celu zużytkowanie istniejącej gruboziarnistej oraz drobnoziarnistej równoległości kodu [14] w oparciu o technikę kompilacji iteracyjnej. Wybór odpowiedniego ciągu transformacji pętli dokonywany jest w następujący sposób. W pierwszej kolejności wybierany jest wariant procedury zrównoleglenia (wynikający z typu algorytmu szyfrowania) zawierający wszystkie transformacje cząstkowe wchodzące w skład danej procedury usuwające, bądź redukujące istniejące zależności w kodzie źródłowym blokujące proces zrównoleglenia. W przypadku, jeśli którejkolwiek z transformacji cząstkowych nie można zastosować z uwagi na specyfikę danej pętli, wówczas kompilator informuje o braku możliwości zastosowania pierwszej z szeregu transformacji tworzących daną sekwencję transformacji oraz generuje błąd transformacji danej pętli, co skutkuje tym, że kod zrównoleglony pętli nie powstaje. W kolejnej iteracji kompilacji wykorzystywany jest kod źródłowy przekształcony w ramach poprzedniej iteracji kompilacji, czyli zawierający wszystkie transformacje poprzedzające transformację, której nie można było zastosować, a z kolei ta transformacja zostaje obecnie opuszczona. Jeżeli w wyniku opuszczenia powyższej transformacji cząstkowej uzyskamy sekwencję transformacji, która będzie się składać wyłącznie z transformacji, które można będzie zastosować, wówczas kompilator kompiluje kod zrównoleglony. W przeciwnym wypadku, kompilator znów informuje o braku możliwości zastosowania pierwszej z szeregu transformacji tworzących sekwencję transformacji uzyskaną w wyniku opuszczenia niedozwolonej transformacji. Analogicznie jak to miało miejsce w ramach poprzedniej iteracji kompilacji kompilator wykorzystuje wiedzę uzyskaną podczas poprzednich iteracji kompilacji i opuszcza kolejną niedozwoloną transformację. Powyższy mechanizm powtarza się, aż do uzyskania sekwencji transformacji zawierającej jedynie transformacje dozwolone dla danej pętli.

Zadania szczegółowe optymalizatora kodu równoległego to:

- pobranie informacji generowanych przez program Petit,
- prywatyzacja zmiennych,
- dystrybucja pętli (ang. *loop distribution*),
- wyrównanie pętli (ang. *loop alignment*),
- łączenie pętli (ang. *loop fusion*),
- zmiana kolejności wykonywania pętli (ang. *loop interchange*),

- konwersja zmiennych na odpowiadające im tablice danych indeksowane według zmiennej indeksowej pętli z zastosowaniem operatora adresu (&).

## 5.5. Translator wstawiający dyrektywy standardu OpenMP

Wiele kompilatorów języka C, zarówno komercyjnych (np. Intel® C++ Compiler) jak i wywodzących się ze środowiska Open Source (np. GCC, the GNU Compiler Collection) wspiera standard programowania równoległego OpenMP. (krótka charakterystyka OpenMP API – patrz pkt.3).

Zadaniem translatora wstawiającego dyrektywy standardu OpenMP API jest dostarczenie zestawu dyrektyw do reprezentacji równoległości kodu źródłowego algorytmów szyfrowania.

## 5.6. Profiler kodu

Zadaniem profilera kodu jest:

- ekstrakowanie dynamicznego zachowania kodu równoległego (całego programu oraz poszczególnych pętli programowych),
- dostarczenie wyników pracy do kontrolera transformacji kodu.

## 5.7. Kontroler transformacji kodu

Zadaniem kontrolera transformacji kodu jest:

- określenie pętli czasochłonnych obliczeniowo,
- uniemożliwienie transformacji pętli zawierających operacje wejścia/wyjścia,
- wyliczenie maksymalnego przyspieszenia danej implementacji algorytmu szyfrowania,
- zarządzanie kolejnością wprowadzania poszczególnych wariantów transformacji składających się z ciągu transformacji cząstkowych zgodnie z Procedurą 1, Procedurą 2 lub Procedurą 3 zrównoleglenia pętli (w zależności o klasy algorytmu szyfrowania).

## 6. Wnioski

Opracowanie kompilatora równoleglającego wyspecjalizowanego na algorytmy szyfrowania wymaga integracji istniejących narzędzi (program Petit, OpenMP) oraz narzędzi wynikających ze specyfiki rozpatrywanego problemu (wstępny restryktoryzator kodu, translator kodu z języka C na język Petit, optymalizator kodu równoległego, profiler kodu, kontroler transformacji kodu). Połączenie w ramach jednego kompilatora proponowanych wyżej narzędzi, a w szczególności najlepszego istniejącego obecnie narzędzia do przeprowadzenia analizy zależności (Petit), standardu programowania równoległego (OpenMP) oraz optymalizatora kodu umożliwi efektywne zrównoleglenie kodu źródłowego algorytmów szyfrowania. W toku dalszych prac w pierwszej kolejności należy zaimplementować translator C->Petit

oraz wstępny restrukturyzator kodu celem umożliwienia przeprowadzenia analizy zależności pętli programowych.

## Bibliografia

- [1] Muchnick S. *Advanced Compiler Design and Implementation*. Morgan Kaufmann Publishers, 1997
- [2] Wilson R., et al. *The SUIF compiler system: A parallelizing and optimizing research compiler*. SIGPLAN, 1994
- [3] Blume W., et al. *Effective Automatic Parallelization with Polaris*. International Journal of Parallel Programming, 1995
- [4] Polychronopoulos C., et al. *The Structure of Parafrase-2: An Advanced Parallelizing Compiler for C and Fortran, Languages and Compilers for Parallel Computing*. MIT Press, 1990
- [5] Bielecki W. *Esentials of Parallel and Distributed Computing*. Informa, 2002
- [6] Burak D. *Metoda automatycznego zrównoleglenia wybranych algorytmów szyfrowania*. Rozprawa doktorska, Politechnika Szczecińska, Wydział Informatyki, 2007
- [7] Kelly W., et al. *New User Interface for Petit and Other Extensions*. User Guide, 1996
- [8] *OpenMP Application Program Interface*. Version 2.5, May 2005
- [9] Chandra R., et al. *Parallel Programming in OpenMP*. Morgan Kaufmann, 2000
- [10] Nahum E., et al. *Towards High Performance Cryptographic Software*. Third IEEE Workshop on the Architecture and Implementation of High Performance Communications Subsystems (HPCS), CT, 1995
- [11] Knijnenburg P., Kisuki T., O'Boyle M. *Iterative compilation. Embedded Processor Design Challenges: Systems, Architectures, Modelling and Simulation – SAMOS*, 2002
- [12] Knijnenburg P., Kisuki T., O'Boyle M. *Combined selection of tile sizes and unroll factors using iterative compilation*. Journal of Supercomputing, 2003
- [13] Knijnenburg P., Kisuki T., Gallivan K. *Cache Models for Iterative Compilation*. Proc. Euro-Par, 2001. Springer Lecture Notes in Computer Science 2150, 2001, s. 254-261
- [14] Allen R., Kennedy K. *Optimizing compilers for modern architectures: A Dependence-based Approach*. Morgan Kaufmann Publishers, Inc., 2001



# Zastosowanie zbiorów przybliżonych i systemu ROSE2 do indukcji reguł decyzyjnych na potrzeby systemu automatycznego obrotu parą walutową

Piotr Czapiewski

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*The paper presents the attempt of applying rough sets theory and ROSE2 software to build the automated currency pair trading system. ROSE2 system's rule induction module was used to extract rules from the feature space built upon moving averages differences and other technical analysis indicators. The rules obtained were evaluated through the simulation of a trading system.*

## Słowa kluczowe:

*rough sets, ROSE2, rule induction, automated trading, moving averages*

## 1. Wprowadzenie

Problem predykcji wartości instrumentów finansowych pojawia się zwykle w kontekście wspomaganie decyzji inwestora. Stosowane bywa zarówno podejście regresyjne, w którym przewidujemy przyszłe wartości ceny [15], jak i klasyfikacyjne, w którym wskazujemy właściwą decyzję dotyczącą kupna, sprzedaży lub powstrzymania się od transakcji [1, 14]. Podejmowane były liczne próby wykorzystania narzędzi z obszaru sztucznej inteligencji i eksploracji danych, w tym także zbiorów przybliżonych, dając często bardzo dobre rezultaty [1, 13, 14, 15]. Powszechnie przyjmuje się, że tego rodzaju metody sprawdzają się w przypadku chaotycznych rynków znacznie lepiej, niż klasyczne metody statystyczne [9, 15].

Rezultaty zastosowania zbiorów przybliżonych do predykcji zachowania rynku znajdziemy m.in. w artykułach [1, 13]. Prace te dotyczą jednak handlu średnio- lub długoterminowego, często na mało płynnych rynkach, i jak już wspomniano – należy je rozpatrywać w kategorii wspomaganie decyzji inwestycyjnych, a nie handlu automatycznego *sensu stricte*. Ponadto dotyczą one okresu sprzed kilku lub kilkunastu lat, podczas gdy rynki finansowe, a w szczególności rynek walutowy, w ostatnich latach diametralnie zmieniły swoje oblicze, zwiększając płynność, nieprzewidywalność i zbliżając do teoretycznego rynku efektywnego. W literaturze coraz częściej znaleźć możemy opinie, że na współczesnych, płynnych rynkach klasyczne metody wywodzące się z analizy technicznej sprawdzają się coraz gorzej [6]. Tym bardziej interesujące staje się wykorzystanie komputera jako narzędzia do szybkiego, krótkoterminowego handlu automatycznego. Narzędzia, które będzie

w stanie wykryć krótkotrwałe wahania rynku i zareagować na nie znacznie szybciej, niż byłby to w stanie zrobić inwestor ręcznie dokonujący transakcji. Niniejszy artykuł prezentuje próbę budowy takiego systemu w oparciu o indukcję reguł z użyciem zbiorów przybliżonych i programu ROSE2.

## 2. Teoria zbiorów przybliżonych

Podstawowym celem teorii zbiorów przybliżonych jest opanowanie niepewności i nieprecyzyjności informacji [2, 7]. Zastosowanie zbiorów przybliżonych pozwala na indukcję reguł decyzyjnych w przypadku, gdy zbiór danych opisujących zagadnienie jest wewnętrznie sprzeczny.

W przypadku danych pochodzących z rynków finansowych, zwłaszcza danych wysokiej częstotliwości, niespójność informacyjna jest zwykle bardzo wyraźna. Sytuacje opisane tymi samymi wartościami atrybutów warunkowych, lecz prowadzące do różnych zachowań ceny, występują bardzo często. Statystyczna przewaga na korzyść jednego kierunku zmiany ceny bywa minimalna, a co za tym idzie, wydobywanie deterministycznych reguł okazuje się niezwykle trudne lub wręcz niemożliwe.

Wśród najważniejszych zalet teorii zbiorów przybliżonych, predestynujących ją do zastosowania w automatycznych systemach transakcyjnych, wymienić warto następujące fakty:

- generowane reguły są jasne, zrozumiałe dla człowieka i łatwe do zaimplementowania w systemie automatycznego handlu;
- istnieje możliwość uwzględnienia dowolnych atrybutów – ciągłych, dyskretnych oraz kwalifikatywnych;
- TZP pozwala na efektywną budowę reguł dla sprzecznych wewnętrznie danych.

Najczęściej zbiory przybliżone stosowane są do klasyfikacji. W przypadku systemu transakcyjnego każdy obiekt wejściowy, charakteryzujący chwilową sytuację na rynku, będzie przypisywany do jednej z dwóch klas: rekomendacja kupna lub sprzedaży. W stosunku do typowego zadania klasyfikacji pojawiają się tu dwie istotne różnice. Po pierwsze, dopuszczalny, a wręcz pożądanym jest brak kompletności reguł – nie każdy obiekt musi zostać zaklasyfikowany. Lepiej nie podjąć decyzji o dokonaniu transakcji niż podjąć błędną. Po drugie, obiekty wejściowe ułożone są chronologicznie, tworzone na podstawie szeregu czasowego. Należy uwzględnić zależności czasowe, stąd zbiory uczące i testowe budowane są z następujących po sobie próbek, nigdy nie pochodzą z losowania.

Do podstawowych pojęć stosowanych w teorii zbiorów przybliżonych należą *system informacyjny* oraz *tablica decyzyjna*. Systemem informacyjnym nazywamy uporządkowaną czwórkę [7]:

$$SI = (U, Q, V, f) \tag{1}$$

gdzie  $U$  oznacza zbiór rozważanych obiektów,  $Q$  – zbiór cech,  $V$  – zbiór wartości przyjmowanych przez cechy, a funkcja informacyjna  $f$  przyporządkowuje wartości poszczególnych atrybutów dla każdego z obiektów.

Tablicą decyzyjną nazywamy uporządkowaną piątkę [7]:

$$DT = (U, C, D, V, f) \quad (2)$$

gdzie  $C$  oznacza zbiór atrybutów warunkowych, a  $D$  – zbiór atrybutów decyzyjnych. W dalszej części artykułu zostanie przedstawiony proces tworzenia systemu informacyjnego oraz tablicy decyzyjnej dla problemu wspomaganie decyzji transakcyjnych.

### 3. Dane wejściowe

#### 3.1. Specyfikacja atrybutów warunkowych

Przedmiotem opisywanych badań jest szereg czasowy złożony z historycznych wartości kursu pary walutowej EUR/USD. Zbiór danych tworzy 6198 próbek, notowanych w odstępach jednogodzinnych i obejmujących rok 2006. Notowania pochodzą z zasobów firmy MetaQuotes Software Corp., a pobrane zostały za pośrednictwem platformy MetaTrader<sup>3</sup>.

Do opisu chwilowej sytuacji na rynku zastosowane zostały wskaźniki należące do najstarszych i najpopularniejszych narzędzi analizy technicznej, mianowicie różnice średnich kroczących o odmiennych okresach. Średnią kroczącą  $MAn$  o okresie  $n$  definiujemy jako średnią wartość ostatnich  $n$  wartości kursu [4, 5]:

$$MAn_t = \frac{1}{n} \sum_{i=0}^{n-1} x_{t-i} \quad (3)$$

Przez  $mMAn$  oznaczmy różnicę dwóch średnich kroczących o okresach  $m$  i  $n$ , przy czym od krótszej średniej zawsze odejmujemy dłuższą:

$$mMAn_t = MAm_t - MAn_t, \quad m < n \quad (4)$$

Tradycja stosowania systemu transakcyjnego opartego na przecięciach dwóch średnich kroczących o różnych okresach sięga lat trzydziestych XX wieku [5]. Za sygnał kupna lub sprzedaży instrumentu finansowego przyjmuje się moment przecięcia przez różnicę  $mMAn$  wartości zero. System w tak prostej postaci generuje sygnały z opóźnieniem, sprawdza się tylko w sytuacji, gdy występuje wyraźny, długoterminowy trend, przy handlu średnio- lub długoterminowym na mało płynnych rynkach. Na współczesnych płynnych rynkach walutowych, znacznie bliższych teorii rynku efektywnego, system taki nie przynosi zadowalających rezultatów [6].

Opisywany tu model stanowić ma rozszerzenie systemu przecięcia średnich kroczących poprzez poszukiwanie reguł w przestrzeni złożonej z wielu par średnich kroczących. Zakładając, że różnice średnich o różnych okresach odpowiadają pewnym zakresom harmonicznym występujących w szeregu czasowym, możliwe powinno być wydobycie z nich pewnych prawidłowości związanych z krótkoterminowymi wahaniami cen.

<sup>3</sup> Patrz <http://www.metaquotes.net>

Dodatkowo przestrzeń atrybutów uzupełniono o jeden z popularnych wskaźników analizy technicznej, tzw. oscylator stochastyczny. Wskaźnik ten wykorzystywany jest przede wszystkim do wykrycia punktów zwrotnych krótkoterminowych cyklów rynku, a definiuje się go następująco [4]:

$$\%K_t = 100 \cdot \frac{C_t - L_t^{14}}{H_t^{14} - L_t^{14}} \quad (5)$$

gdzie  $C_t$  oznacza aktualną cenę zamknięcia, a  $L^{14}$  i  $H^{14}$  to odpowiednio najniższa i najwyższa wartość ceny w okresie ostatnich 14 interwałów czasowych (w omawianym przypadku – 14 godzin).

Liczba atrybutów oraz okresy średnich kroczących i oscylatora stochastycznego dobrane zostały w sposób arbitralny. Przyjęto 4 atrybuty warunkowe:

$$Q = \{q_1, q_2, q_3, q_4\} \quad (6)$$

Oznaczenia atrybutów przedstawia tabela 1. Atrybuty  $q_1$  do  $q_3$  zbudowano na bazie średnich kroczących o okresach 5, 7, 10 i 20. Dla oscylatora przyjęto okres 14 interwałów.

Tabela 1. Atrybuty warunkowe

Oznaczenie atrybutu	Reprezentowany wskaźnik
$q_1$	5MA7
$q_2$	5MA10
$q_3$	5MA20
$q_4$	%K(14)

Przeźren atrybutów użytą w omawianym systemie można oczywiście rozszerzyć, uwzględniając inne wskaźniki analizy technicznej, średnie kroczące i oscylatory o innych okresach lub też pochodne atrybutów czy atrybuty przesunięte w czasie. Jedną z zalet zbiorów przybliżonych jest możliwość uwzględnienia informacji fundamentalnych (politycznych, makroekonomicznych), w tym danych kwalifikatywnych i opinii ekspertów.

Okresy średnich kroczących oraz oscylatora dobrane zostały arbitralnie, podobnie sama liczba użytych atrybutów. Docelowo należy rozważyć możliwość automatycznej selekcji cech – ogromna liczba potencjalnych atrybutów sugeruje, iż selekcja cech w ramach teorii zbiorów przybliżonych (znajdowanie reduktu) może być nieefektywna. W planach dalszych badań autor uwzględni m.in. ewolucyjną selekcję atrybutów warunkowych.

### 3.2. Wstępne przetwarzanie danych

W opisywanym podejściu szereg czasowy traktujemy jako ciąg chronologicznie ułożonych próbek, bez uwzględniania bezwzględnego czasu ich pobrania. Rozważenia wymaga problem kompletności oraz poprawności danych.

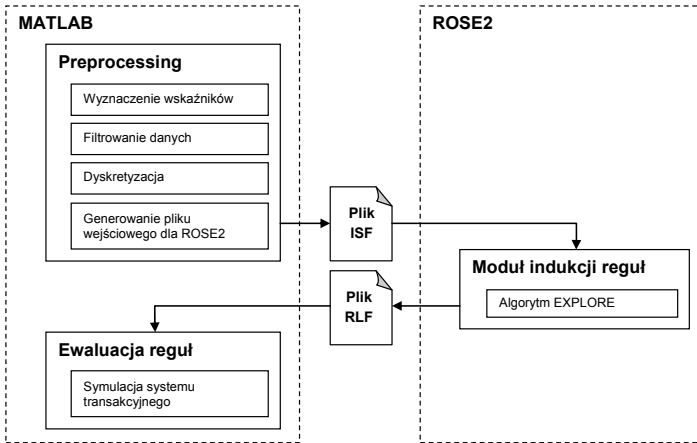
Analiza danych wykazała brak wartości wykraczających poza oczekiwany zakres zmienności i odbiegających od reszty próbek (*outliers*), można zatem uznać poprawność danych za wystarczającą i niewymagającą korekty. Kwestia brakujących wartości nie wymaga w opisywanym przypadku specjalnego rozstrzygnięcia. Jak już wspomniano, nie jest brany pod uwagę bezwzględny czas pobrania konkretnej wartości, a jedynie chronologiczne następstwo próbek. Wszystkie wskaźniki obliczane są bezpośrednio z podstawowego szeregu cen, a nie pobierane z zewnętrznego źródła, nie występuje zatem problem synchronizacji poszczególnych atrybutów. Ponadto istotą użycia zbiorów przybliżonych jest opanowanie niepewności i wewnętrznej niespójności danych, zatem pojedyncze przypadki nieprawidłowych danych wejściowych nie mogą rzutować w istotny sposób na jakość wygenerowanych reguł decyzyjnych.

Wejściowy szereg czasowy należy zamienić na postać akceptowaną w teorii zbiorów przybliżonych. Jedną z najprostszych i najbardziej efektywnych metod stosowanych w takich przypadkach jest ułożenie kolejnych wskaźników (atrybutów) w kolumnach tabeli informacyjnej i potraktowanie każdego wiersza (a więc zbioru wskaźników obliczonych dla pojedynczej chwili czasowej) jako odrębnego obiektu podlegającego klasyfikacji [1]. Shen i Loh [1] proponują 5-etapową procedurę przygotowywania tablicy informacyjnej:

1. Tworzenie kolumn początkowych – obliczenie wartości wskaźników.
2. Synchronizacja – w omawianym przypadku nie jest konieczna, wartości już po wygenerowaniu są właściwie ułożone. Synchronizacja będzie niezbędna jeśli zechcemy zastosować wskaźniki przesunięte (opóźnione).
3. Uwzględnienie brakujących danych – nie jest konieczne z przyczyn podanych powyżej. Jeśli w szeregu podstawowym brakowało pojedynczej próbki, to w tabeli informacyjnej zabraknie całego wiersza danych, co nie będzie miało istotnego wpływu na wynik indukcji reguł.
4. Uzupełnienie atrybutów czasowych – dodanie informacji o wystąpieniu pewnych zdarzeń. Autorzy na tym etapie umiejscawiają wskazanie właściwej dla danego wiersza decyzji. Należy zatem dokonać analizy przyszłych wartości ceny i podać decyzję prowadzącą w danym punkcie do osiągnięcia zysku.
5. Stworzenie szeregu pochodnego – wygenerowanie na podstawie powyższych danych ostatecznej wersji tabeli decyzyjnej. W opisywanym eksperymencie na tym etapie generowany jest plik wejściowy dla systemu ROSE2.

#### 4. Zastosowanie systemu ROSE2

ROSE2 (*Rough Sets Data Explorer*) jest systemem implementującym podstawowe techniki teorii zbiorów przybliżonych [3]. Pozwala na interaktywną analizę danych, dyskretyzację, automatyczną indukcję reguł za pomocą trzech różnych algorytmów, a także testowanie otrzymanych reguł. System ma budowę modułową – poszczególne moduły komunikują się za pośrednictwem plików tekstowych o strukturze opisanej w dokumentacji [11]. Pozwala to na łatwą integrację ROSE2 ze środowiskiem Matlab lub docelowo z automatycznym systemem transakcyjnym.



Rysunek 1. Modułowy schemat środowiska eksperymentalnego

W opisywanym eksperymencie wykorzystano moduł indukcji reguł decyzyjnych. Wstępne przetwarzanie danych realizowane jest po stronie środowiska Matlab, tam też tworzony jest plik wejściowy dla modułu indukcji reguł. Po automatycznym wywołaniu ROSE2 w trybie wsadowym, wygenerowane przezeń i zapisane do pliku reguły decyzyjne są odczytywane i poddawane ewaluacji. Schemat umiejscawiający system ROSE2 w eksperymencie przedstawiono na rysunku 1. Zastosowany model interakcji pozwoli na rozszerzenie systemu w przyszłości, np. o automatyczną ewolucyjną dyskretyzację.

## 5. Opis eksperymentu

### 5.1. Budowa tabeli decyzyjnej

Zbiór uczący przygotowany został na podstawie danych obejmujących 8 początkowych tygodni (960 próbek) roku 2006, po pominięciu pierwszych 20 wartości szeregu ze względu na konieczność obliczenia średniej kroczącej  $MA20$ .

Dyskretyzację przeprowadzono dobierając arbitralnie granice przedziałów, wspólne dla atrybutów  $q_1$ - $q_3$  oraz odrębne dla  $q_4$ . Sposób dyskretyzacji odzwierciedla tradycyjne poglądy analityków technicznych. W przypadku różnic średnich kroczących interesuje nas zatem nie tyle dokładna wartość atrybutu, co określenie jego znaku bądź wskazanie wartości „bliskiej zera”. Podobnie w przypadku oscylatora – określamy przedziały odpowiadające wyprzedaniu rynku, wykupieniu oraz okresowi trendu. W naturalny sposób nasuwa się zatem sugestia zastosowania atrybutów rozmytych i wnioskowania rozmytego, co zostanie uwzględnione w dalszych badaniach.

Docelowo autor planuje zastosować dyskretyzację ewolucyjną, np. za pomocą metody proponowanej przez Czerniaka [8]. Wykorzystanie systemu ROSE2 w try-

bie wsadowym pozwoli na prostą integrację własnego mechanizmu dyskretyzacji z pozostałymi elementami systemu.

Przedziały dyskretyzacji i sposób zakodowania atrybutów warunkowych przedstawia tabela 2.

Tabela 2. Kodowanie atrybutów warunkowych

Atrybut	Granice przedziału	Wartość zakodowana
$q_1, q_2, q_3$	$(-\infty; -0.00010)$	1
	$\langle -0.00010; 0.00010 \rangle$	2
	$(0.00010; \infty)$	3
$q_4$	$(-\infty; 30)$	1
	$\langle 30; 70 \rangle$	2
	$(0; \infty)$	3

Atrybut decyzyjny  $D$  określony został jako dwuwartościowy, odpowiadający rekomendacji kupna ( $D = 1$ ) bądź sprzedaży ( $D = -1$ ). Rekomendacja ustalana jest na podstawie zmiany wartości ceny po osiągnięciu zadanego horyzontu czasowego – przyjęto horyzont równy 2 interwałom czasowym, zatem o rekomendacji decyduje zmiana ceny po upływie dwóch godzin od rozpatrywanego punktu:

$$\begin{cases} x_{t+2} - x_t > \Delta x_{min} & \Rightarrow D_t = 1 \\ x_{t+2} - x_t < -\Delta x_{min} & \Rightarrow D_t = -1 \end{cases} \quad (7)$$

Parametr  $\Delta x_{min}$  określa minimalny poziom zmiany ceny  $\Delta x$ , prowadzący do rekomendacji transakcji. Jeśli bezwzględny przyrost ceny będzie mniejszy niż  $\Delta x_{min}$ , system nie wygeneruje żadnej rekomendacji.

Tabela 3. Kodowanie atrybutu decyzyjnego

Atrybut	Zmiana ceny	Rekomendacja	Wartość zakodowana
$D$	$\Delta x > 0,0005$	Kup	1
	$\Delta x < -0,0005$	Sprzedaj	-1

Z podstawowego szeregu odrzucone zostały wszystkie przypadki, w których cena nie zmieniła się istotnie po upływie zadanego horyzontu czasowego ( $|\Delta x| < \Delta x_{min}$ ). Takie podejście proponowane jest przez licznych autorów dla różnych metod indukcyjnych (por. [9]) i zapobiegać ma zdominowaniu uczenia przez przypadki o niewielkim przyroście ceny. Dla częstych notowań, gdzie odstęp między kolejnymi próbkami jest rzędu kilku minut do godziny, tego rodzaju próbki mogą stanowić większość – wówczas algorytm uczenia może do nich właśnie nadmiernie dopasować system, prowadząc do sytuacji, którą można opisać następująco:

- wysoka ogólna skuteczność rozpoznawania;
- wysoka skuteczność rozpoznawania przypadków o  $|\Delta x| < \Delta x_{min}$  (przewidywanie niezmienności ceny, rekomendacja wstrzymania się od przeprowadzania transakcji);
- niska skuteczność rozpoznawania przypadków o  $|\Delta x| > \Delta x_{min}$  (najistotniejsza grupa przypadków – przewidywanie wzrostu lub spadku ceny, rekomendacja przeprowadzenia transakcji).

Szereg uczący po usunięciu opisanych przypadków liczył ostatecznie 613 próbek. Tabela 4 przedstawia oznaczenia konceptów decyzyjnych oraz ich liczebność. Tabela 5 przedstawia przykładową tablicę decyzyjną (dla fragmentu zbioru uczącego). Poniżej przedstawiono postać wygenerowanego pliku wejściowego dla systemu ROSE2.

Tabela 4. Liczebność konceptów decyzyjnych dla zbioru uczącego

Koncept decyzyjny	Wartość atrybutu D	Liczebność
$X_1$	-1	311
$X_2$	1	302

Tabela 5. Fragment tablicy informacyjnej

Obiekt (U)	MA (q1)	MA (q2)	MA (q3)	Stoch (q4)	Decyzja (D)
1	2	1	1	2	-1
2	2	2	1	2	1
3	2	2	1	1	1
4	3	3	1	3	1
5	3	3	2	3	1

## 5.2. Analiza aproksymacji zbiorów

Wygenerowany plik tekstowy, zawierający dane uczące, został następnie zaimportowany do systemu ROSE2 w celu wstępnej analizy zbioru danych. Wyznaczonych zostało 55 zbiorów elementarnych.

Wyniki analizy jakości przybliżenia zbiorów przedstawia rysunek 1. Jak widać dolne przybliżenia konceptów decyzyjnych są wyjątkowo mało liczne:

$$\begin{cases} \text{card } DP(X_1) = 22 \\ \text{card } DP(X_2) = 9 \end{cases} \quad (8)$$

Pozytywny obszar rodziny  $D^*$  zawiera jedynie 31 przykładów, co pociąga za sobą bardzo niską jakość przybliżenia:

$$\gamma(D^*) = 0,0506 \quad (9)$$



## Program 1. Wygenerowany plik isf dla systemu ROSE2

---

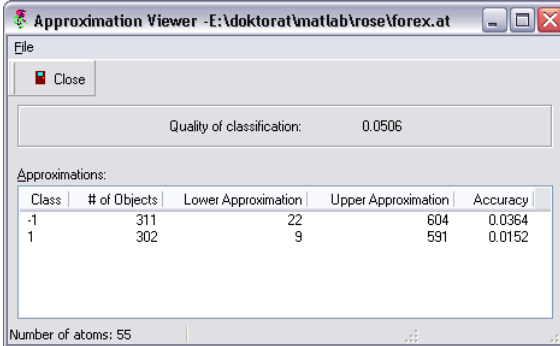
```

**ATTRIBUTES
A1: (numbercoded)
A2: (numbercoded)
A3: (numbercoded)
A4: (numbercoded)
Dec: [-1,1]
decision: Dec
**EXAMPLES
2 1 1 2 -1
2 2 1 2 1
2 2 1 1 1
3 3 1 3 1
3 3 2 3 1
**END

```

---

Powyzsza analiza prowadzi do wniosku, że z omawianego zbioru danych można wydobyc bardzo niewiele reguł deterministycznych. Ponadto przydatność tych reguł będzie znikoma, ze względu na niewielkie ich wsparcie. Należy zatem skupić się na poszukiwaniu reguł niedeterministycznych o jak największej pewności i wsparciu.



The screenshot shows a window titled "Approximation Viewer - E:\doktorat\matlab\rose\forex.at". It displays the "Quality of classification" as 0.0506. Below this is a table of approximations with the following data:

Class	# of Objects	Lower Approximation	Upper Approximation	Accuracy
-1	311	22	604	0.0364
1	302	9	591	0.0152

At the bottom of the window, it indicates "Number of atoms: 55".

Rysunek 2. Wynik analizy aproksymacji zbiorów w systemie ROSE2

### 5.3. Indukcja reguł

Indukcja reguł w systemie ROSE2 wykonywana jest za pomocą algorytmów LEM2, MODLEM lub EXPLORE [3, 10]. Oferowane są trzy tryby indukcji [11]:

- *Basic Minimal Covering* (LEM2) – generowanie kompletnego i spójnego zbioru reguł o jak najmniejszej liczbie i długości reguł;

- *Extended Minimal Covering* (MODLEM) – j.w., także dla systemów zawierających ciągle atrybuty bez potrzeby ich dyskretyzacji; generowanie reguł relacyjnych postaci  $a \leq \text{atrybut} \leq b$ ;
- *Satisfactory Description* (EXPLORE) – generowanie zbioru reguł spełniających zadane warunki.

W opisywanym przypadku zastosowano trzeci z wyżej wymienionych trybów indukcji, wymagający określenia następujących parametrów:

- maksymalna długość reguły,
- minimalna siła reguły (względna lub bezwzględna),
- minimalny poziom dyskryminacji.

Eksperymentalnie dobrano następujące parametry:

- maksymalna długość: 4,
- minimalna siła bezwzględna: 10,
- minimalny poziom dyskryminacji: 60%.

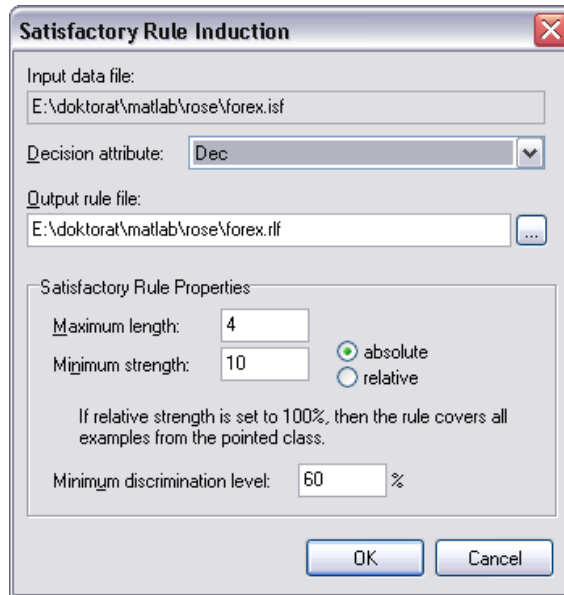
Przy powyższych parametrach w wyniku indukcji powstało 6 reguł niedeterministycznych o wsparciu od 15 do 134 i sile od 12 do 81. Uzyskane reguły przedstawia tabela 6.

Tabela 6. Reguły uzyskane z systemu ROSE2

Lp.	Reguła	Wsparcie	Siła	Poziom dysk.
1.	$(A1=2) \Rightarrow (Dec=-1)$	134	81	60,45%
2.	$(A2=2) \Rightarrow (Dec=-1)$	68	43	63,24%
3.	$(A3=2) \Rightarrow (Dec=-1)$	30	18	60,00%
4.	$(A3=1) \ \& \ (A4=3) \Rightarrow (Dec=-1)$	19	12	63,16%
5.	$(A1=1) \ \& \ (A2=3) \Rightarrow (Dec=1)$	15	12	80,00%
6.	$(A1=1) \ \& \ (A3=1) \ \& \ (A4=2) \Rightarrow (Dec=1)$	61	37	60,66%

Na tym etapie warto zastanowić się, jak siła i wsparcie reguł przekładają się na późniejsze zastosowanie systemu w praktyce. Ponieważ analizowany okres obejmował 8 tygodni, siła na poziomie kilkunastu oznacza średnio około 2 transakcji tygodniowo, co przy handlu krótkoterminowym można uznać za minimum, aby system był użyteczny. Wszystkie otrzymane reguły powinny wygenerować około 5 rekomendacji transakcji dziennie.

Liczbę oraz charakterystykę reguł uzyskanych przy odmiennych parametrach przedstawia Tabela 7. Zbyt wysokie wymagania co do pewności reguł powodują generowanie reguł o małym wsparciu, prowadząc do systemu o niskiej użyteczności i wiarygodności.



Rysunek 3. Parametry indukcji reguł w systemie ROSE2

#### 5.4. Ewaluacja zbioru reguł

Otrzymane z systemu ROSE2 reguły zostały następnie przetestowane na zbiorze obejmującym dane do końca roku 2006 (z pominięciem próbek należących do zbioru uczącego). Rozmiar zbioru testowego wyniósł 5215 próbek.

Tabele 8 i 9 przedstawiają statystyki dotyczące liczby oraz wyniku rekomendowanych w tym okresie transakcji. Wyniki transakcji podawane są w jednostkach zwanych pipsami, przy czym jeden pips oznacza zmianę na ostatnim miejscu znaczącym ceny. W opisywanym przypadku pary walutowej EUR/USD 1 pips = 0,0001 USD. Rzeczywiste wyniki wyrażone względnym lub bezwzględnym przyrostem kapitału zależą od wielkości stosowanej dźwigni finansowej oraz ewentualnych dodatkowych kosztów transakcji.

Tabela 7. Charakterystyka reguł przy różnych parametrach indukowania

Parametry			Rezultat indukcji		
Maks. długość	Min. siła	Min. poziom dyskryminacji	Liczba reguł	Min. wsparcie	Maks. wsparcie
3	30%	100%	0	–	–
3	30%	60%	0	–	–
4	10	75%	4	15	20
4	10	60%	6	15	134
4	20	60%	3	61	134

Tabela 8. Liczba rekomendowanych transakcji

Liczba transakcji	Kupno	Sprzedaż	Razem
Udanych	295	858	1153
Nieudanych	220	808	1028
<b>Razem</b>	515	1666	2181

Tabela 9. Wyniki przeprowadzonych transakcji

	Kupno	Sprzedaż	Razem
<b>Zysk z udanej transakcji</b>			
Minimalny	0	0	0
Średni	13	10	11
Maksymalny	111	111	111
<b>Strata z nieudanej transakcji</b>			
Minimalna	1	1	1
Średnia	12	12	12
Maksymalna	91	125	125
<b>Średni wynik pojedynczej transakcji</b>			
	2,13	-0,38	0,21

Przewaga transakcji udanych nad nieudanymi nie jest duża, rzędu 10%. Stąd średni wynik transakcji, choć dodatni, jest bliski zeru. Współczynnik Sharpe'a, definiowany jako stosunek średniego wyniku do odchylenia standardowego wyników [12]:

$$S = \frac{\bar{d}}{\sigma_d} \quad (10)$$

osiągnął niewielką dodatnią wartość:

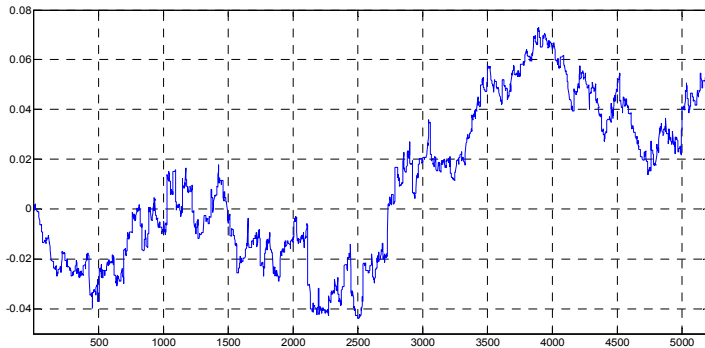
$$S = 0,0117 \quad (11)$$

Na rysunku 3 widzimy zmiany symulowanego stanu konta. Widać wyraźnie, że skuteczność rekomendacji znacznie zmieniała się w czasie – w pewnym okresie stan konta oscyluje wokół wartości początkowej, w późniejszym wyraźnie wzrasta. Sugerowałoby to wprowadzenie jednej z dwóch możliwych modyfikacji systemu:

- ponowne uczenie systemu co pewien okres (np. 8 tygodni),
- poszukiwanie wskaźnika charakteryzującego okresy, w których system sprawdza się lepiej i filtrowanie z jego użyciem generowanych rekomendacji.

## 6. Wnioski

Otrzymane rezultaty pozwalają stwierdzić, że opracowany system nie jest jeszcze gotowy do praktycznego stosowania, jednak kierunek badań wydaje się obie-



Rysunek 4. Wykres symulowanego stanu konta w okresie testowym

cujący. W handlu krótkoterminowym osiągnięcie wysokiego stosunku liczby transakcji korzystnych do niekorzystnych jest niezwykle trudne, dążyć należy raczej do zwiększania stabilności strategii – utrzymania nieznacznej, ale stałej przewagi transakcji przynoszących zysk. Wyniki wskazują wyraźnie na istnienie statystycznie uzasadnionej zależności między szybkimi wahaniami cen a wartościami dobraneo zbioru atrybutów. Dalsze badania należałoby prowadzić w trzech niżej opisanych kierunkach.

Okresowa wysoka skuteczność systemu sugerowałaby poszukiwanie wskaźnika opisującego chwilowy charakter rynku i wskazującego na zasadność zastosowania omawianego systemu. Być może uzupełnienie przestrzeni atrybutów o średnie kroczące i oscylatory o dłuższych okresach pozwoli na zwiększenie trafności rekomendacji.

Jak już wspomniano, konieczne jest zastosowanie automatycznej dyskretyzacji, jak również selekcji atrybutów warunkowych. W obu przypadkach obiecujące wydaje się być podejście ewolucyjne i niewątpliwie zostanie ono w przyszłości przetestowane. Ponadto ze względu na sposób interpretacji zastosowanych atrybutów sugeruje się wprowadzenie atrybutów rozmytych oraz rozmytego wnioskowania.

Osobną kwestią pozostaje optymalizacja strategii inwestycyjnej wykorzystującej generowane rekomendacje. Wyniki powinno znakomicie podnieść wprowadzenie zleceń zabezpieczających oraz utrzymywanie pozycji przez dłuższy okres w przypadku braku rekomendacji lub rekomendacji zgodnej z poprzednim krokiem.

## Bibliografia

- [1] Shen L., Loh H. T. *Applying rough sets to market timing decisions*. Decision Support Systems 37, 2004, s. 583-597
- [2] Pawlak Z., Skowron A. *Rudiments of rough sets*. Information Sciences 177, 2007, s. 3-27
- [3] Prędko B., Słowiński R., Stefanowski J., Susmaga R., Wilk Sz. *ROSE - Software Implementation of the Rough Set Theory*. W: *Rough Sets and Current Trends in Computing*, Polkowski L., Skowron A. (red.), *Lecture Notes in Artificial Intelligence*, vol. 1424. Springer-Verlag, 1998, s. 605-608

- 
- [4] Murphy J. J. *Analiza techniczna rynków finansowych*. WIG-PRESS, 1999
  - [5] Investopedia.com. *Basics of Moving Averages* [online],  
<http://investopedia.com/articles/technical/052201.asp> [dostęp: listopad 2007]
  - [6] Olson D. *Have trading rule profits in the currency markets declined over time?*. Journal of Banking & Finance 28, 2004, s. 85-105
  - [7] Rutkowski L. *Metody i techniki sztucznej inteligencji*. PWN, 2005
  - [8] Czerniak J. *Porównanie metody LDGen z wiodącymi metodami dyskretyzacji wstępnej stosowanymi w kontekście teorii zbiorów przybliżonych*. W: *Metody Informatyki Stosowanej w Zarządzaniu*, Wydział Informatyki Politechniki Szczecińskiej, 2004
  - [9] Deboeck G. J., Cader M. *Pre- and Postprocessing of Financial Data*. W: *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets*, Wiley 1994
  - [10] Grzymala-Busse J., Stefanowski J., Wilk Sz. *A comparison of two approaches to data mining from imbalanced data*. Journal of Intelligent Manufacturing, 16, 2005, s. 565–573
  - [11] *ROSE2 – Rough Set Data Explorer. User's Guide*. ProSoft, 1999
  - [12] Sharpe W. F. *The Sharpe Ratio*. The Journal of Portfolio Management Fall, 1994, s. 49-58
  - [13] Tay F. E. H., Shen L. *Economic and financial prediction using rough sets model*. European Journal of Operational Research 141, 2002, s. 641-659
  - [14] Qian B., Rasheed K. *Stock market prediction with multiple classifiers*. Applied Intelligence 26, 2007, s. 25-33
  - [15] Yao J., Tan C. L. *A case study on using neural networks to perform technical forecasting of forex*. Neurocomputing 34, 2000, s. 79-98

# Wbudowanie wiadomości w dane medialne na podstawie analizy dwóch najmniej znaczących bitów kontenera

Larisa Dobryakova, Evgeny Ochin

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*In this article is elaborated a steganography algorithm on the base on the partly distortion of container LSB with use the pair of LSB of containers.*

## Słowa kluczowe:

*steganography, container, embedding of dates, stego-message, Least Significant Bit (LSB)*

## 1. Wprowadzenie

Zadanie bezpieczeństwa informacji przed niesankcjonowanym dostępem było rozwiązywane przez cały okres historii ludzkości. Nauka steganografia była znana już wiele wieków temu. Rozwój technologii obliczeniowych i szerokie rozpowszechnienie technologii multimedialnych dał mocny impuls do rozwoju steganografii cyfrowej.

Steganografia – nauka o organizacji ukrytego przesyłania danych, przy której ukrywa się sam fakt przesyłania danych. W odróżnieniu od kryptografii, gdzie obserwator potrafi ustalić, czy jest przesyłana wiadomość zaszyfowanym tekstem, metody steganografii pozwalają na przesyłanie konfidennej informacji niepostrzeżenie dla osób obcych [1].

Obecnie w związku z rozwinięciem techniki obliczeniowej i kanałów prędkościowych przesyłania informacji pojawiły się nowe metody steganograficzne, na podstawie których leżą właściwości przedstawienia informacji w plikach komputerowych.

## 2. Algorytmy zamiany LSB danych medialnych

W steganografii komputerowej istnieją dwa główne rodzaje plików: stego-wiadomość (ukrywana wiadomość) i kontener – plik, który używa się do ukrycia w nim wiadomości. Początkowy stan kontenera, gdy jeszcze nie zawiera ukrytej informacji, nazywa się kontenerem-oryginałem, a konieczny stan kontenera, gdy już zawiera stego-wiadomość, nazywa się kontenerem-rezultatem. W charakterze kontenera-oryginału często wykorzystuje się różne pliki medialne, na przykład, pliki

obrazu, dźwięku, wideo i inne. Wbudowanie stego-wiadomości w kontener idzie w parze z pewnymi zniekształceniami danych medialnych, jednak charakter zniekształceń musi być minimalny, aby słuchacz (lub widz), nie zauważył pogorszenia jakości danych, zawartych w pliku medialnym. Wzrok i słuch człowieka nie odróżnia nieznacznych zmian w kolorach obrazu lub w jakości dźwięku [2, 3, 4].

Jedną ze znanych metod steganografii z wykorzystaniem plików medialnych jest zamiana najmniej znaczących bitów kontenera (LSB) odpowiednią dwójkową stego-wiadomością.

**Zamiana najmniej znaczących bitów jednowymiarowych danych medialnych.** Metodę tę można wyjaśnić następująco: niech kontener-oryginał przedstawia  $I$ -elementowy wektor liczb dwójkowych  $N$ -pozycyjnych, które reprezentowane są w kodzie uzupełniającym i opisują pewne dane audio:

$$D_i = -d_{i,N-1} \cdot 2^{N-1} + \sum_{n=0}^{N-2} d_{i,n} \cdot 2^n, \quad i = 0, 1, \dots, I-1 \quad (1)$$

gdzie  $d_{i,n} \in \{0, 1\}$  –  $n$ -ta cyfra dwójkowa  $i$ -tej próbki kontenera-oryginału,  $d_{i,N-1}$  – pozycja znaku  $i$ -tej próbki kontenera-oryginału ( $d_{i,N-1} = 0$  dla liczb nieujemnych,  $d_{i,N-1} = 1$  dla liczb ujemnych), przy czym:  $-2^{N-1} \leq D_i \leq (2^{N-1} - 1)$ . Na przykład, dla  $N = 8$ , zakres liczb reprezentowanych przez  $D_i$  wynosi  $-128 \leq D_i \leq 127$ .

Stego-wiadomość tekstową zawsze można przedstawić w postaci wiersza bitowego długości  $K$  bitów:

$$\{s_0 s_1 \cdots s_k \cdots s_{K-1}, \quad s_k \in \{0, 1\}\} \quad (2)$$

Oznaczmy kontener-rezultat jako:

$$D_i^+ = -d_{i,N-1}^+ \cdot 2^{N-1} + d_{i,0}^+ + \sum_{n=1}^{N-2} d_{i,n}^+ \cdot 2^n, \quad i = 0, 1, \dots, I-1 \quad (3)$$

gdzie  $d_{i,n}^+ \in \{0, 1\}$  –  $n$ -ta cyfra dwójkowa  $i$ -tej próbki kontenera-rezultatu,  $d_{i,N-1}^+$  – pozycja znaku  $i$ -tej próbki kontenera-rezultatu.

Algorytm wbudowania stego-wiadomości w kontener:

$$\left\{ \begin{array}{l} D_i^+ = -d_{i,N-1} \cdot 2^{N-1} + s_i + \sum_{n=1}^{N-2} d_{i,n} \cdot 2^n, \quad i = 0, 1, \dots, K-1 \\ D_i^+ = D_i, \quad i = K, \dots, I-1 \end{array} \right\}$$

Stego-wiadomość zamienia najmniej znaczące bity pierwszych  $K$  próbek kontenera-oryginału, co doprowadza do pewnych zniekształceń tego kontenera.

**Zamiana najmniej znaczących bitów dwuwymiarowych danych medialnych** [1, 5, 6, 7]. Większość badań zostało przeprowadzonych w kierunku wykorzystania obrazów w postaci stego-kontenerów. Istnieje ku temu kilka powodów [1, 8]:



- Niezbędność ochrony zdjęć cyfrowych, obrazów, plików wideo przed bezprawnym kopiowaniem i rozpowszechnianiem.
- Dostyc duży rozmiar cyfrowej reprezentacji obrazu.
- Znany z wczesnego rozmiar kontenera oraz brak ograniczeń, które nakładane są przez wymagania ukrywania wiadomości w czasie rzeczywistym.
- Słaba czułość oka człowieka na nieznaczne zmiany kolorów, jaskrawości i kontrastu obrazów, oraz na zniekształcenia w pobliżu konturów.
- Dobrze opracowane metody obróbki obrazów.

Niech kontener-oryginał reprezentuje obraz w postaci macierzy  $I \times J$   $N$ -pozycyjnych całkowitych liczb dwójkowych:

$$D_{i,j} = \sum_{n=0}^{N-1} d_{i,j,n} \cdot 2^n, \quad i = 0, 1, \dots, I-1, \quad j = 0, 1, \dots, J-1 \quad (4)$$

gdzie  $d_{i,j,n} \in \{0, 1\}$  –  $n$ -ta cyfra dwójkowa  $i, j$ -tej próbki kontenera-oryginału. Zakres reprezentowanych liczb wynosi:  $0 \leq D_{i,j} \leq (2^N - 1)$ . Na przykład, dla  $N = 8$ , zakres wynosi  $0 \leq D_{i,j} \leq 255$ .

Kontener-rezultat, jak i kontener-oryginał reprezentuje obraz w postaci macierzy  $I \times J$   $N$ -pozycyjnych całkowitych liczb dwójkowych:

$$D_{i,j}^+ = d_{i,j,0}^+ + \sum_{n=1}^{N-1} d_{i,j,n} \cdot 2^n, \quad i = 0, 1, \dots, I-1, \quad j = 0, 1, \dots, J-1 \quad (5)$$

gdzie  $d_{i,j,n} \in \{0, 1\}$  –  $n$ -ta ( $n = 1, 2, \dots, N$ ) cyfra dwójkowa  $i, j$ -tej próbki kontenera-rezultatu, która równa jest odpowiedniej cyfrze kontenera-oryginału,  $d_{i,j,0}^+ \in \{0, 1\}$  – najmniej znaczące bity  $i, j$ -tej próbki kontenera-rezultatu, które przedstawiają wynik wbudowania stego-wiadomości tekstowej. Algorytm wbudowania stego-wiadomości w kontener można opisać następująco:

$$\left. \begin{array}{l} k = 0 \quad /*k - \text{numer bitu stego-wiadomości} */ \\ \text{for } i = 0, 1, \dots, I-1 : \\ \quad \text{for } j = 0, 1, \dots, J-1 : \\ \quad \quad \text{if } k < K \text{ then } D_{i,j}^+ = s_k + \sum_{n=1}^{N-1} d_{i,j,n} \cdot 2^n \\ \quad \quad \quad \text{else } D_{i,j}^+ = D_{i,j}; \\ \quad \quad \quad k = k + 1; \\ \quad \quad \text{end;} \\ \quad \text{end;} \\ \text{if } k < K \text{ then Goto Error\_Krótki\_kontener} \end{array} \right\} \quad (6)$$

Stego-wiadomość zamienia najmniej znaczące bity  $d_{i,j,0}$  pierwszych  $K$  próbek kontenera-oryginału, co doprowadza do pewnych zniekształceń tego kontenera.

### 3. Wbudowanie wiadomości w dane medialne na podstawie częściowego zniekształcenia najmniej znaczących bitów kontenera z wykorzystaniem analizy dwóch najmniej znaczących bitów kontenera

W niniejszej pracy w celu zmniejszania stego-błędów rozpatrzone zostało wbudowanie wiadomości w dane medialne na podstawie analizy zawartości dwóch najmniej znaczących bitów  $d_{i,1}d_{i,0}$  próbek danych i modyfikacji najmniej znaczących bitów  $d_{i,0}$  próbek danych.

Dla danych jednowymiarowych algorytm ten można opisać następująco:

$$\left. \begin{array}{l}
 k = 0 \quad /* k - \text{numer bitu stego-wiadomości} */ \\
 \text{for } i = 0, 1, \dots, I - 1 : \\
 \quad \left[ \begin{array}{l}
 \text{if } s_k d_{i,1} d_{i,0} = 000 \text{ then } \{D_i^+ = D_i; k = k + 1;\} \\
 \text{if } s_k d_{i,1} d_{i,0} = 001 \text{ then } \{d_{i,0} = 0; k = k + 1;\} \\
 \text{if } s_k d_{i,1} d_{i,0} = 011 \text{ then } d_{i,0} = 0; \\
 \text{if } s_k d_{i,1} d_{i,0} = 111 \text{ then } \{D_i^+ = D_i; k = k + 1;\} \\
 \text{if } s_k d_{i,1} d_{i,0} = 110 \text{ then } \{d_{i,0} = 1; k = k + 1;\} \\
 \text{if } s_k d_{i,1} d_{i,0} = 100 \text{ then } d_{i,0} = 1; \\
 \text{else } D_i^+ = D_i; \\
 \text{else if } k = K \text{ then Goto End\_Wbudowanie\_skuteczne} \\
 \text{end;} \\
 \text{if } k < K \text{ then Goto Error\_Krótki\_kontener} \\
 \text{End\_Wbudowanie\_skuteczne: } F_{min} = k; \\
 /* F_{min} - \text{minimalna długość kontenera,} \\
 \quad \text{potrzebna do wbudowywania stego-wiadomości} */
 \end{array} \right\} \quad (7)$$

Algorytm odczytywania stego-wiadomości z kontenera-rezultatu:

$$\left. \begin{array}{l}
 k = 0 \\
 \text{for } i = 0, 1, \dots, I - 1 : \\
 \quad \left[ \begin{array}{l}
 \text{if } d_{i,1}^+ = d_{i,0}^+ \text{ then } \{s_k = d_{i,0}^+; k = k + 1;\}; \\
 \text{if } k \geq K \text{ then goto End\_of\_destego} \\
 \text{end;} \\
 \text{End\_of\_destego:}
 \end{array} \right\} \quad (8)$$

Dla danych dwuwymiarowych algorytm ten można opisać następująco:

$$\left. \begin{array}{l}
 k = 0 \quad /* k - \text{numer bitu stego-wiadomości} */ \\
 \text{for } i = 0, 1, \dots, I - 1 : \\
 \quad \text{for } j = 0, 1, \dots, J - 1 : \\
 \quad \quad \left[ \begin{array}{l}
 \text{if } s_k d_{i,j,1} d_{i,j,0} = 000 \text{ then } \{D_{i,j}^+ = D_{i,j}; k = k + 1;\} \\
 \text{if } s_k d_{i,j,1} d_{i,j,0} = 001 \text{ then } \{d_{i,j,0} = 0; k = k + 1;\} \\
 \text{if } s_k d_{i,j,1} d_{i,j,0} = 011 \text{ then } d_{i,j,0} = 0; \\
 \text{if } s_k d_{i,j,1} d_{i,j,0} = 111 \text{ then } \{D_{i,j}^+ = D_{i,j}; k = k + 1;\} \\
 \text{if } s_k d_{i,j,1} d_{i,j,0} = 110 \text{ then } \{d_{i,j,0} = 1; k = k + 1;\} \\
 \text{if } s_k d_{i,j,1} d_{i,j,0} = 100 \text{ then } d_{i,j,0} = 1; \\
 \text{else } D_{i,j}^+ = D_{i,j};
 \end{array} \right] \\
 \quad \text{else if } k = K \text{ then Goto End\_Wbudowanie\_skuteczne} \\
 \text{end; end;} \\
 \text{if } k < K \text{ then Goto Error\_Krótki\_kontener} \\
 \text{End\_Wbudowanie\_skuteczne: } F_{min} = i * I + j; \\
 \\
 /* F_{min} - \text{minimalna dugość (liczba próbek) kontenera,} \\
 \quad \text{potrzebna do wbudowywania stego-wiadomości} */
 \end{array} \right\} \quad (9)$$

Algorytm odczytywania stego-wiadomości z kontenera-rezultatu:

$$\left. \begin{array}{l}
 k = 0; \\
 \text{for } i = 0, 1, \dots, I - 1 : \\
 \quad \text{for } j = 0, 1, \dots, J - 1 : \\
 \quad \quad \text{if } d_{i,j,1}^+ = d_{i,j,0}^+ \text{ then } \{s_k = d_{i,j,0}^+; k = k + 1\}; \\
 \quad \quad \text{if } k \geq K \text{ then goto End\_of\_destego} \\
 \quad \text{end;} \\
 \text{end;} \\
 \text{End\_of\_destego:}
 \end{array} \right\} \quad (10)$$

Zaletą tych algorytmów jest to, że dla odzyskania stego-wiadomości z kontenera-rezultatu wykorzystana się dwa najmniej znaczące bity kontenera ( $d_{i,j,1}^+ d_{i,j,0}^+$  -

w przypadku 1D oraz  $d_{i,j,1}^+ d_{i,j,0}^+$  – w przypadku 2D), chociaż zniekształcenia wprowadza się tylko w bity  $d_{i,0}$  (1D) lub  $d_{i,j,0}$  (2D), co zmniejsza prawdopodobieństwo odnalezienia wiadomości przez osoby obce.

Realizacji działania algorytmu wbudowania wiadomości na podstawie częściowego zniekształcenia najmniej znaczących bitów kontenera z wykorzystaniem analizy dwóch najmniej znaczących bitów kontenera w 1D i 2D dane medialne opracowana została w środowisku MATLAB i potwierdziła prawidłowe działanie algorytmu. Eksperyment pokazał, że wbudowanie stego-wiadomości w pliki 1D i 2D nie doprowadza do zmian pliku audio zauważalnych dla słuchu człowieka i do zmian obrazu widocznych dla oka człowieka.

#### 4. Wartość zniekształcenia kontenera przy częściowym zniekształceniu najmniej znaczących bitów kontenera

Oszacowanie ilościowe odporności zabezpieczenia systemu steganograficznego przed zewnętrznymi oddziaływaniami przedstawia dosyć trudne zadanie, które w praktyce realizowane jest zazwyczaj za pomocą metod analizy systemowej, modelowania matematycznego lub badania eksperymentalnego [5, 8].

Z zasady, zawodowo opracowany stego-system zabezpiecza trzypoziomowy model ochrony danych, który rozwiązuje dwa główne zadania:

- ukrywanie faktu istnienia chronionych danych (pierwszy poziom ochrony);
- blokowanie niesankcjonowanego dostępu do danych, które realizuje się za pomocą wyboru odpowiedniej metody ukrywania danych (drugi poziom ochrony).

W charakterze trzeciego poziomu ochrony może występować wstępna ochrona kryptograficzna (szyfrowanie) ukrywanych danych.

Do porównawczego szacowania jakości środków steganograficznych stosuje się różne wskaźniki, dające oceny ilościowe.

Przy analizie poziomu zniekształceń najbardziej rozpowszechnioną miarą oceny zniekształceń wnoszonych w kontener jest obecnie wartość średniokwadratowa (*Mean Square Error*).

**Wartość średniokwadratowa dla 1D:**

$$MSE = \frac{1}{F_{min}} \sum_{i=0}^{F_{min}-1} (D_i - D_i^+)^2 = \frac{1}{F_{min}} \sum_{i=0}^{F_{min}-1} |d_{i,0} - d_{i,0}^+|, \quad (11)$$

$F_{min}$  – minimalna długość kontenera potrzebna do wbudowywania stego-wiadomości (wartość ta jest wartością zmienną, zależną od zawartości stego-wiadomości i kontenera).

**Wartość średniokwadratowa dla 2D:**

$$MSE = \frac{1}{F_{min}} \sum_{i=0}^{I_{end}-1} \sum_{j=0}^{J_{end}-1} (D_{i,j} - D_{i,j}^+)^2 = \frac{1}{F_{min}} \sum_{i=0}^{I_{end}-1} \sum_{j=0}^{J_{end}-1} |d_{i,j,0} - d_{i,j,0}^+|, \quad (12)$$

Tabela 1. Ocena wartości średniokwadratowej MSE

Metoda	MSE	$F_{\min}/K$
Algorytm zamiany LSB, 1D	0,50	1
Algorytm zamiany LSB, 2D	0,50	1
Algorytm na podstawie cząstkowych zniekształceń LSB, 1D	0,15	1,93
Algorytm na podstawie cząstkowych zniekształceń LSB, 2D	0,14	2,09

$F_{\min}$  – minimalna długość (liczba próbek) kontenera, potrzebna do wbudowywania stego-wiadomości,  $I_{end} = \lceil F_{\min}/I \rceil$ , gdzie operator  $\lceil X \rceil$  oznaczają część całkowitą od  $X$ ,  $J_{end} = F_{\min} - \lceil F_{\min}/I \rceil * I$ .

W tabeli 1 przedstawiona została ocena wartości średniokwadratowej MSE dla opracowanego algorytmu wbudowania wiadomości w dane medialne na podstawie częściowego zniekształcenia najmniej znaczących bitów kontenera oraz dla algorytmu zamiany najmniej znaczących bitów pierwszych  $K$  próbek kontenera. Modelowanie wykonano w środowisku MATLAB.

Także policzona została zależność pomiędzy minimalnie potrzebną długością kontenera a długością wiadomości  $F_{\min}/K$ . Jak widać z tabeli wartość ta dla opracowanych algorytmów jest dwa razy większa, niż długość wbudowanej wiadomości, jednakże wartość średniokwadratowa jest 3 razy mniejsza, niż dla algorytmów zamiany najmniej znaczących bitów pierwszych  $K$  próbek kontenera.

## 5. Wniosek

W rezultacie przeprowadzonych eksperymentów pokazano, że wartość średniokwadratowa przy wykorzystaniu metody wbudowania wiadomości na podstawie częściowego zniekształcenia najmniej znaczących bitów kontenera z wykorzystaniem analizy dwóch najmniej znaczących bitów kontenera w 1D i 2D dane medialne jest 3 razy mniejsza, niż przy wbudowywaniu stego-wiadomości w kontener-oryginał metodą zamiany pierwszych  $K$  próbek kontenera-oryginału. Oprócz tego, metoda ta jest bardziej odporna przed odczytywaniem wiadomości przez osoby obce, niż istniejące metody wbudowania stego-wiadomości w dane medialne.

## Bibliografia

- [1] Gribunun W. G., Okow I. N., Turincew I. W. *Steganografia cyfrowa*. SOLON-Press, Moskwa 2002 (w jęz. rosyjskim)
- [2] Bender W., Gruhl B., Morimoto N., Lu A. *Techniques for data hiding*. IBM systems journal, Vol. 35, No. 3, 1996
- [3] Petitcolas F., Anderson R., Kuhn M. *Information Hiding – A Survey*. Proceedings IEEE, Special Issue on Identification and Protection of Multimedia Information. Vol. 87, No. 7, 1999

- [4] Arnold M., Kanka S. *MP3 robust audio watermarking*. International Watermarking Workshop, 1999
- [5] Choroszko W. O., Azarow O. D., Szelest M. E., Jaremczuk Ju. E. *Podstawy steganografii komputerowej*. WDTU<sub>b</sub> 2003 (w jęz. rosyjskim)
- [6] Fridrich J., Du R., Long M. *Steganalysis of LSB Encoding In Color Images*. Proceedings of ICME 2000, New York City, July 31 – August 2, New York, USA
- [7] Fridrich J., Goljan M., Du Rui. *Reliable Detection of LSB Steganography in Color and Grayscale Images*. IEEE Multimedia, Vol. 8, 2001
- [8] Konachowicz G. F., Puzyrienko A. Ju. *Steganografia komputerowa. Teoria i praktyka*. MK-Press, Kijów 2006 (w jęz. rosyjskim)

# Moduł ekspertyz w multimedialnym atlasie wad laminatów

Sławomir Jaszczak, Krzysztof Gorący, Waldemar Pankanin

*Politechnika Szczecińska, Wydział Informatyki,  
Politechnika Szczecińska, Wydział Technologii i Inżynierii Chemicznej*

## Abstract:

*Modern database systems connected with an Internet administration panel in www browser are an excellent tool for collecting and ordering an expert knowledge. A knowledge acquisition process is the most time consuming stage in the procedure of creating a new rule expert system and usually demands many repetitions and arrangements between a knowledge engineer and an expert engineer from a specific area. Authors of this article are interested in an expert knowledge related to failures of composites materials, that are popular materials used in the production of many articles used in everyday life. Data about composites failures stored in the SQL database systems were converted into the txt file i.e. rules and facts database for the JESS reasoning machine. JESS is a modification of the CLIPS language and gives an knowledge engineer opportunity to do expert system with elements of JAVA language. In this article was presented a construction of the rules and facts generator and a way of integration it with a www application named the Multimedia Atlas of Composites Failures.*

## Słowa kluczowe:

*expert systems, knowledge representation, rule-based knowledge representation, JESS*

## 1. Wprowadzenie

Moduł ekspertyz multimedialnego atlasu wad laminatów umożliwia dokonywanie diagnoz wad kompozytów polimerowych poprzez Internet. Narzędzie tego typu powinno w znacznym stopniu ułatwić pracę ekspertów dziedzinowych, dokonujących oceny wad wyrobów gotowych, wytwarzanych w trakcie złożonego procesu technologicznego oraz spowodować skrócenie czasu podejmowania przez nich ostatecznych decyzji.

Jednym z najważniejszych wymogów postawionych w trakcie tworzenia omawianego modułu było zagwarantowanie wiarygodności i spójności wiedzy na podstawie, której będzie wnioskowanie. Z tego powodu źródłem wiedzy dziedzinowej byli wyłącznie eksperci z Instytutu Polimerów Politechniki Szczecińskiej.

Założono ponadto, że ekspertyzy mogą być dokonywane w trybie „online”. Wykorzystanie silnika JESS oraz języka Java umożliwiło zrealizowanie tego zadania.

W wyniku analizy opracowano listę szczegółowych wymagań funkcjonalnych, które powinien spełniać moduł ekspertyz:

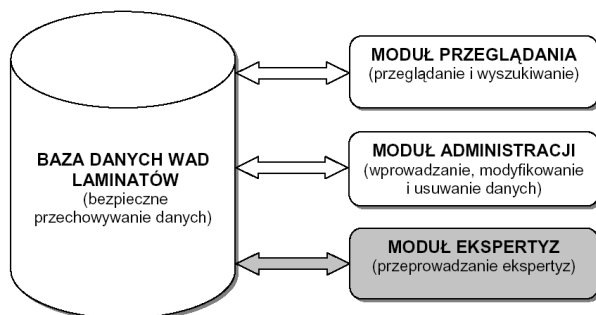
- niezbędnym i najważniejszym składnikiem modułu jest generator reguł i faktów dla maszyny wnioskującej JESS - stworzenie bazy wiedzy stanowi element niezbędny do rozpoczęcia procesu wnioskowania;
- możliwość pracy użytkownika z modułem ekspertyz w trybie dialogowym, który jest jednym z najpowszechniejszych sposobów pracy większości systemów ekspertowych działających na stronach WWW;
- interfejs użytkownika, powinien umożliwiać pracę w trybie dialogowym z systemem ekspertowym za pośrednictwem Internetu;
- możliwość wyboru przypuszczalnych wad dla procesu wnioskowania;
- efektem końcowym wnioskowania przy pomocy modułu powinna być wada lub lista wad zdiagnozowanych, lista przypuszczalnych przyczyn ich powstania oraz metody zapobiegania i metody napraw;
- przenośność między platformami sprzętowymi i programowymi dzięki zastosowaniu odpowiednich technologii;
- możliwość pracy wielu użytkowników w tym samym czasie.

## 2. Architektura multimedialnego atlasu wad laminatów z uwzględnieniem modułu ekspertyz

Multimedialny atlas wad laminatów, przedstawiony na rys.1 składa się z:

- modułu przeglądania zawartości,
- modułu administracji i zarządzania,
- modułu ekspertyz, stworzonego w zakresie niniejszej pracy.

Moduł ekspertyz stanowi integralną część całej aplikacji. Wszystkie moduły są powiązane z istniejącą bazą danych wad kompozytów polimerowych, która stanowi „fundament” całego systemu.



Rysunek 1. Schemat aplikacji z uwzględnieniem modułu ekspertyz



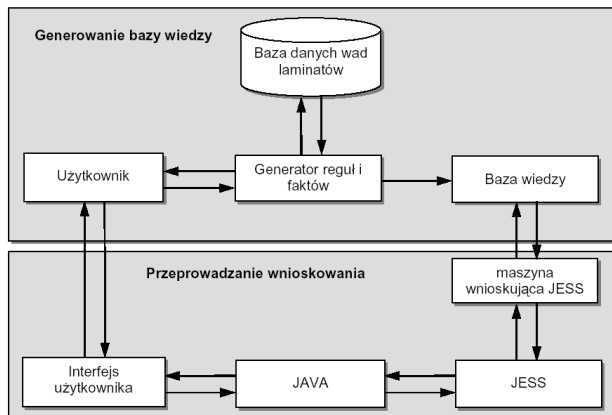
Jak wynika z rys. 1 baza danych o wadach laminatów połączona jest z wszystkimi modułami aplikacji. Przechowuje uporządkowane dane dotyczące wad kompozytów polimerowych i stanowi źródło wiedzy dla bazy wiedzy stworzonego modułu ekspertowego.

Moduł ekspertyz multimedialnego atlasu wad laminatów składa się z:

- generatora reguł i faktów dla maszyny wnioskującej JESS;
- apletu języka Java z interfejsem użytkownika, umożliwiającym przeprowadzanie ekspertyz wad laminatów.

Generator reguł i faktów odpowiada za utworzenie bazy wiedzy dla modułu ekspertyz na podstawie danych zawartych w bazie danych. Wiedza generowana jest w postaci pliku reguł i faktów o odpowiedniej, wcześniej opracowanej strukturze.

Aplet Javy z interfejsem użytkownika stanowi środowisko graficzne dla maszyny wnioskującej JESS i umożliwia prowadzenie dialogu systemu z użytkownikiem w celu ustalenia powstałej wady kompozytu polimerowego wraz z przyczynami jej powstania oraz metodami napraw i zapobiegania. Na rys. 2. przedstawiona jest struktura modułu ekspertyz. Moduł ekspertyz podzielony jest na dwie główne części: generowanie bazy wiedzy oraz przeprowadzanie wnioskowania. W pierwszej części zasadniczą rolę odgrywa blok „Generator reguł i faktów”, którego zadaniem jest utworzenie bazy wiedzy w postaci pliku reguł i faktów.



Rysunek 2. Schemat modułu ekspertyz multimedialnego atlasu wad laminatów

Wygenerowany plik ładowany jest do maszyny wnioskującej JESS i przy pomocy interfejsu użytkownika (apletu napisanego w języku Java) przeprowadzany jest dialog z systemem. Efektem końcowym, czyli wynikiem ekspertyzy są: lista zdiagnozowanych wad laminatów oraz dodatkowe dane odnośnie przyczyn powstania oraz metod naprawy i zapobiegania każdej z tych wad.

Procesem wnioskowania, polegającym na „odpalaniu” konkretnych reguł według określonej strategii, dodawaniu i usuwaniu faktów z pamięci roboczej zajmuje się maszyna wnioskująca JESS.

Kolejność wykonywania działań podczas pracy z modułem wygląda następująco:

- 1) użytkownik rozpoczyna pracę z modułem poprzez wygenerowanie pliku reguł i faktów stanowiącego bazę wiedzy dla maszyny wnioskującej JESS;
- 2) wygenerowany plik ładowany jest do maszyny wnioskującej JESS;
- 3) użytkownik przy pomocy apletu komunikującego się poprzez język Java z silnikiem JESS przeprowadza dialog z systemem w celu ustalenia diagnozy;
- 4) efektem końcowym pracy z modułem ekspertyz jest lista rozpoznanych wad laminatów.

Jednym z najważniejszych elementów zbudowanego modułu ekspertyz jest generator bazy wiedzy. Jego zadaniem jest utworzenie pliku reguł i faktów na podstawie danych zawartych w bazie danych wad laminatów.

Praca z generatorem stanowi pierwszy, wymagany etap pracy z modułem ekspertyz, gdyż bez pliku reguł i faktów nie możliwe byłoby przeprowadzenie drugiego etapu, czyli dokonania ekspertyzy.

Sam generator zaimplementowany został w języku PHP. Natomiast struktura generowanego pliku zawierającego wiedzę potrzebną przy wnioskowaniu została opracowana w języku JESS.

Poprawność merytoryczna oraz syntaktyczna wiedzy gromadzonej w pliku reguł i faktów uzależniona jest od poprawności danych zawartych w bazie danych na podstawie, których ta wiedza jest tworzona.

Poniżej przedstawiony został schemat generatora pliku reguł i faktów (Rys. 3).

Zgodnie z rys. 3. baza danych wad laminatów jest elementem kluczowym generatora pliku reguł i faktów. Odpowiada ona za dostarczenie odpowiednich danych do formularza wyboru wad do ekspertyzy i jest głównym źródłem wiedzy dla bazy wiedzy modułu diagnostycznego.

Do najważniejszych funkcji generatora reguł i faktów należą:

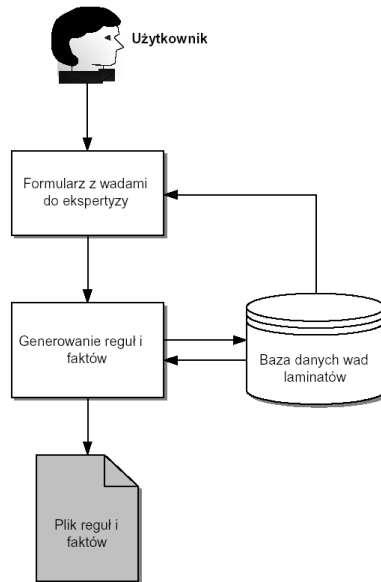
- wybór przypuszczalnych wad do ekspertyzy przy pomocy formularza WWW,
- generowanie spójnej i kompletnej bazy wiedzy w postaci pliku reguł i faktów na podstawie danych zawartych w bazie danych wad laminatów.

Utworzony plik oprócz zbioru faktów i reguł stanowiących bazę wiedzy dla modułu ekspertyz zawiera również dodatkowe funkcje oraz konstrukcje językowe umożliwiające posłużenie się nim jako samodzielnym programem, który wystarczy załadować do środowiska JESS, by natychmiastowo rozpocząć proces wnioskowania.

Poniżej znajduje się schemat ilustrujący zadania realizowane przez generator reguł i faktów (Rys. 4).

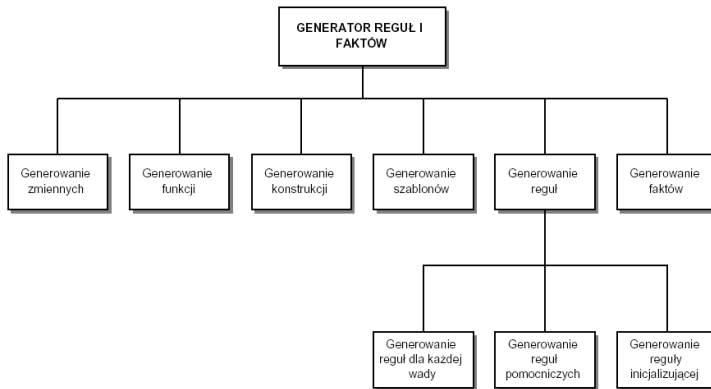
Zgodnie z rys. 4 zadania generatora reguł i faktów obejmują:

- **Generowanie zmiennych** – generowane są wszystkie zmienne globalne JESS, które używane są w trakcie wnioskowania. Są to głównie zmienne pomocnicze do obliczania procentowej oceny wystąpienia danej wady. Generowaniem zmiennych zajmuje się funkcja `genDefblobal()`.



Rysunek 3. Uogólniony schemat generatora reguł i faktów

- **Generowanie funkcji** – generowane są wszystkie funkcje używane w trakcie wykonywania programu JESS. Służą one głównie do wyświetlania tekstu na ekranie, np. wyniku wnioskowania czy pytań pojawiających się w trakcie dialogu systemu z użytkownikiem. Generowane są przez funkcję `genDeffunction()`.
- **Generowanie konstrukcji** – generowane są konstrukcje językowe JESS używane przez program w trakcie wnioskowania. Do takich konstrukcji należy np. (`Defquery`) opisana w rozdziale poświęconym technologiom wykonania modułu ekspertyz. Generowaniem konstrukcji zajmuje się funkcja `genConstruct()`.
- **Generowanie szablonów** – generowane są konstrukcje, zwane szablonami, używane do zdefiniowania struktury bazy wiedzy w module ekspertyz. Szczegółowo zostały one omówione w podrozdziale pod tytułem „Struktura bazy wiedzy”. Generowaniem szablonów zajmuje się funkcja `genDeftemplate()`.
- **Generowanie reguł** – generowane są wszystkie reguły używane w trakcie wnioskowania. Na podstawie relacji między konkretnymi tabelami w bazie danych wad laminatów tworzone są: reguły dla każdej wady oraz reguła inicjująca proces wnioskowania. Generowane są również reguły pomocne przy obliczaniu procentowej oceny wystąpienia wady oraz wyświetlania wyniku wnioskowania. Generowaniem reguł zajmują się funkcje: `genDefrule_pomocn()`, `genDefruleStart()`, `genDefrule_wada()`.
- **Generowanie faktów** – generowane są wszystkie fakty, które mogą być wprowadzone w trakcie przeprowadzania ekspertyzy na listę faktów. Dodawanie faktów dotyczących wad laminatów realizowane jest przez funkcję `genDefruleStart()`.



Rysunek 4. Schemat zadań realizowanych przez generator reguł i faktów

System zbudowany przy pomocy środowiska JESS zawiera wiedzę eksperta zakodowaną w postaci reguł i faktów. Fakty stanowią pewne struktury danych, które umożliwiają przechowywanie wiedzy w określonej, spójnej formie. JESS udostępnia trzy główne metody reprezentacji faktów. Są to [5]:

- **prosta lista faktów** – tworzona jest poleceniami (**assert**) lub (**defacts**);
- **szablon faktów** – stanowi grupę powiązanych ze sobą danych, może być traktowany jako złożony fakt mający wiele pól o ściśle określonych typach [4,5].
- **klasa (rama)** – to rozbudowany szablon, który opisuje wspólne charakterystyki albo atrybuty obiektów. Można powiedzieć, że klasa jest abstrakcyjnym, uproszczonym modelem pewnego obiektu rzeczywistego świata [4,5]. Klasy w JESS oraz ich instancje umożliwiają programowanie aplikacji zorientowane obiektowo.

W module ekspertyz multimedialnego atlasu wad laminatów wykorzystano szablon jako sposób reprezentacji wiedzy. Szablon w języku JESS tworzony jest przy pomocy konstrukcji (**deftemplate**). Każda taka struktura składa się z pól zwanych także slotami. Posiadają one swoje nazwy, określane również mianem atrybutów oraz opcjonalnie typ przechowywanych w nich wartości. Szablon z określonymi wartościami pól nazywany jest instancją [4,5].

Poniżej znajduje się kod języka JESS przedstawiający ogólną deklarację szablonu wykorzystanego w module ekspertyz:

```

(deftemplate wada
  (slot nazwa_wady)
  (slot kategoria)
  (multislot symptom)
  (multislot przyczyna (default "Brak przyczyn powstawania"))
  (multislot naprawa (default "Brak metod napraw"))
  (multislot zapobieganie (default "Brak metod zapobiegania"))
  (multislot skutki (default "Brak skutków powstania"))
)
  
```

Każda instancja tego szablonu dotyczy pojedynczej wady określonej następującymi wartościami slotów:

- nazwa wady,
- kategoria wady,
- objawy (symptomy) powstania wady,
- przyczyny powstania wady,
- metody zapobiegania powstawania wady,
- skutki powstania wady,
- pole stanowiące licznik wskazań na daną wadę podczas procesu wnioskowania.

Niektóre ze slotów przedstawionego szablonu deklarowane za pomocą polecenia (`multislot`) umożliwiają przechowywanie wartości wielopolowych.

Istotnym slotem w szablonie wady jest „symptom”, gdyż to względem wartości występujących w tym polu przeprowadzany jest proces wnioskowania. Dlatego też slot ten musi być wypełniony wartościami w każdej stworzonej instancji szablonu.

Instancje szablonu, czyli konkretne zbiory faktów dotyczące wad tworzone są przy pomocy polecenia (`assert`) w trakcie trwania procesu wnioskowania. O tym, jakie fakty zostaną wprowadzone do pamięci roboczej JESS decydują odpowiedzi na pytania dotyczące objawów wystąpienia wad laminatów udzielone przez użytkownika w trakcie prowadzenia dialogu z systemem.

Zbiór samych faktów nie jest wystarczający do opisania jakiegokolwiek dziedziny wiedzy. Dlatego też dodatkowo stosuje się reguły, które reprezentowane są najczęściej w postaci operandów logicznych IF, THEN, ELSE.

W bazie wiedzy modułu ekspertyz reguły tworzone są dla każdej wady biorącej udział w procesie wnioskowania. Poniżej znajduje się kod języka JESS przedstawiający przykładową regułę utworzoną dla danej wady.

```
(defrule Pecherze "Reguła opisująca wadę 'Pęcherze' "
?wada-adres <- (wada(nazwa_wady ?nazwa_wady)
(kategoria ?kategoria)
(przyczyna $?przyczyna)
(naprawa $?naprawa)
(skutki $?skutki)
(zapobieganie $?zapobieganie )
(symptom $?symptom&:(member$ "Obłe wypukłości na powierzchni" $?symptom))
)
?ocena-adres <- (ocena(nazwa_w ?nazwa_wady) (licznik ?licznik))
=>
(modify ?wada-adres(symptom "zatwierdzony"))
(modify ?ocena-adres (licznik (+ ?licznik 1))))
```

Wszystkie reguły dotyczące wad laminatów posiadają podobną strukturę do przedstawionej powyżej. Jedyna zmiana może dotyczyć tylko części kodu, gdzie występuje warunek wystąpienia symptomów (podkreślony fragment), który może być różny w zależności od danej wady.

Oprócz typowych struktur budujących bazę wiedzy modułu ekspertyz, plik reguł i faktów zawiera również dodatkowe, pomocne konstrukcje języka JESS. Do takich zaliczyć można np. reguły: `oblicz-procenty1` i `oblicz-procenty2`. Ich zadaniem jest obliczenie procentowego prawdopodobieństwa wystąpienia danej wady.

The screenshot shows a web browser window with a URL starting with 'kresowol.pl'. The page title is 'Strona główna | Szukaj | Ekspertyza | Administrator | Demos | Forum'. The main content area is titled 'Moduł diagnostyczny wad laminatów' and contains a section 'Wybierz wady dla których ma być przeprowadzona ekspertyza'. Below this is a table with two columns: 'Nr katalog.' and 'Nazwa wady'. Each row has a checkbox in the first column. At the bottom of the table are two buttons: 'Zamknij wszystkie' and 'Odmknij wszystkie'. Below the table are two buttons: 'Generuj plik clp' and 'Reset'.

Nr katalog.	Nazwa wady
<input type="checkbox"/>	1 Za małą grubość laminatu
<input type="checkbox"/>	2 Miejscowo za małą grubość laminatu
<input type="checkbox"/>	3 Za dużą grubość laminatu na całej powierzchni
<input type="checkbox"/>	4 Miejscowo za dużą grubość laminatu
<input type="checkbox"/>	5 Za małą wytrzymałość laminatu
<input type="checkbox"/>	6 Za małą zawartość wzmocnienia
<input type="checkbox"/>	7 Za dużą zawartość wzmocnienia
<input type="checkbox"/>	8 Wypaczenie laminatu
<input type="checkbox"/>	9 Brak dokładności wymiarowej
<input type="checkbox"/>	10 Nieoczekiwane zapachy wyrobu podczas eksploatacji
<input type="checkbox"/>	11 Nie oddanie kształtu formy
<input type="checkbox"/>	12 Niedokształtowanie
<input type="checkbox"/>	13 Za małą twardość
<input type="checkbox"/>	14 Mikropełnięta matryca
<input type="checkbox"/>	15 Chropowatość formy
<input type="checkbox"/>	16 Chropowatość
<input type="checkbox"/>	17 Fałszywość
<input type="checkbox"/>	18 Pęcherze
<input type="checkbox"/>	19 Pęknięcia
<input type="checkbox"/>	20 Pęknięcia włóknowate
<input type="checkbox"/>	21 Płamy i przebarwienia
<input type="checkbox"/>	22 Rozwarstwianie żeloutu od laminatu
<input type="checkbox"/>	23 Wtrącenia oboje
<input type="checkbox"/>	24 Włócznie zarzu włókien wzmocnienia
<input type="checkbox"/>	25 Zmarszczki żeloutu
<input type="checkbox"/>	26 Zmiana barwy żeloutu (określenie)
<input type="checkbox"/>	27 Wykaskanie włókna wzmacniającego
<input type="checkbox"/>	28 Sfałdowanie warstwy wzmocnienia

Rysunek 5. Formularz wyboru wad do ekspertyzy

Użytkownik rozpoczynając pracę z generatorem ma możliwość zaznaczenia w formularzu tych wad, dla których ma być generowany plik z bazą wiedzy. W ten sposób już przed rozpoczęciem ekspertyzy można określić przypuszczalne wady, które mogły powstać w badanym kompozycie polimerowym. Jednak takie podejście wcale nie oznacza, że wady, które nie zostały wybrane w formularzu nie będą brane pod uwagę w procesie wnioskowania. Użytkownik może tylko w pewnym stopniu zawęzić zakres poszukiwania powstałej wady, gdyż proces tworzenia bazy wiedzy uzależniony jest od relacji występujących w bazie danych wad laminatów i to jest decydujący czynnik wpływający na ostateczną zawartość generowanego pliku oraz na wynik końcowy ekspertyzy.

Poniżej znajduje się widok okna aplikacji z formularzem przedstawiającym wady, dla których możliwe jest przeprowadzenie ekspertyzy (Rys. 5).

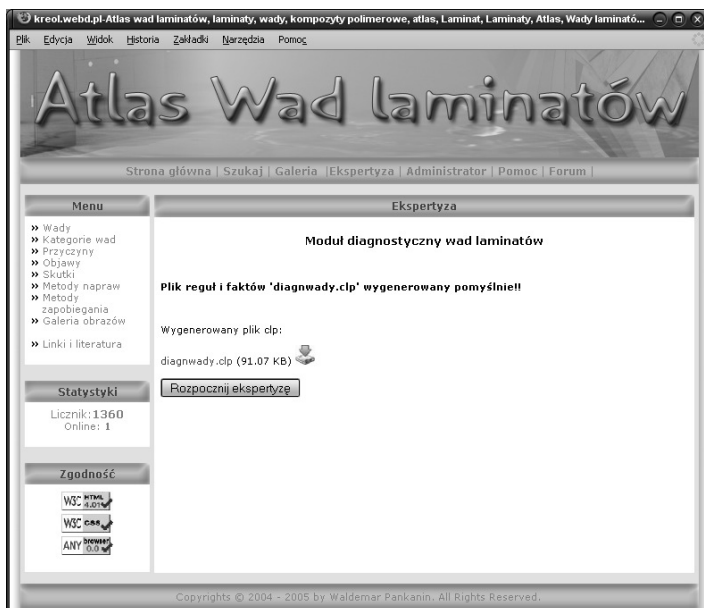
Użytkownik rozpoczynając pracę z modulem ekspertyz musi wybrać przy pomocy formularza przypuszczalne wady, dla których ma być przeprowadzona diagnoza.

Po zaakceptowaniu formularza przyciskiem „Generuj plik clp” użytkownik zostanie przeniesiony na stronę zawierającą odnośnik do utworzonego pliku bazy wiedzy (Rys. 6). Istnieje możliwość obejrzenia zawartości tego pliku w przeglądarce internetowej lub ściągnięcia go na dysk twardy.

Wygenerowany plik zawiera również zestaw zmiennych globalnych używanych w trakcie działania programu.

Po wygenerowaniu pliku z bazą wiedzy użytkownik może przejść do następnego etapu pracy z modulem diagnostycznym, czyli przeprowadzenia ekspertyzy w trybie dialogowym, wykorzystując maszynę wnioskującą JESS lub aplet Javy. JESS umożliwia, bowiem pełną komunikację oraz integrację ze środowiskiem Java. Pozwala to tworzyć rozbudowane aplikacje, które w różnym stopniu korzystają z cech zarówno JESS jak i Javy.

Aplet Javy umożliwia przeprowadzanie diagnoz wad kompozytów polimerowych przy pomocy prostego interfejsu użytkownika, który został stworzony tak, aby stawianie diagnoz odnośnie wad laminatów odbywało się na zasadzie prowadzenia dialogu użytkownika



Rysunek 6. Widok okna aplikacji po wygenerowaniu pliku z bazą wiedzy

z systemem i na podstawie tego wyciąganiu wniosków w postaci listy zdiagnozowanych wad.

Poniżej znajduje się widok okna apletu Javy w trakcie przeprowadzania ekspertyzy (Rys.7) oraz przykładowy scenariusz ekspertyzy.

Pytanie nr 1 – „Czy występuje objaw: Laminat ma większą grubość niż zaplanowano”?

Odpowiedź udzielona na pytanie nr 1 – „TAK”.

Pytanie nr 2 – „Czy istnieją jeszcze jakieś objawy”?

Odpowiedź udzielona na pytanie nr 2 – „TAK”.

Pytanie nr 3 – „Czy występuje objaw: Wyrób ma widoczne zgrubienia na dużych powierzchniach”?

Odpowiedź udzielona na pytanie nr 3 – „NIE”.

Pytanie nr 4 – „Czy występuje objaw: Wykonany laminat ma za małą wytrzymałość mechaniczną”?

Odpowiedź udzielona na pytanie nr 4 – „NIE”.

Pytanie nr 5 – „Czy występuje objaw: Wykonany laminat ma za małą wytrzymałość mechaniczną”?

Odpowiedź udzielona na pytanie nr 5 – „NIE”.

Pytanie nr 6 – „Czy występuje objaw: Laminat ma większą grubość niż zaplanowano”?

Odpowiedź udzielona na pytanie nr 6 – „TAK”.

Pytanie nr 7 – „Czy istnieją jeszcze jakieś objawy”?

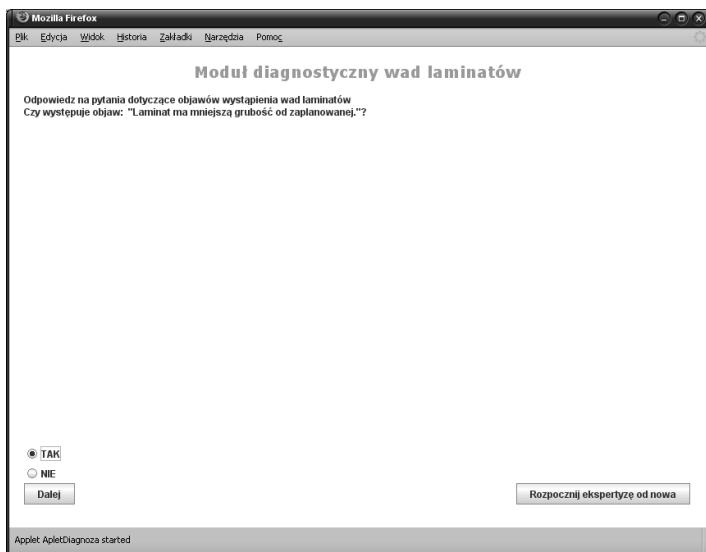
Odpowiedź udzielona na pytanie nr 7 – „TAK”.

Pytanie nr 8 – „Czy występuje objaw: Za duża grubość laminatu”?

Odpowiedź udzielona na pytanie nr 8 – „TAK”.

Pytanie nr 9 – „Czy istnieją jeszcze jakieś objawy”?

Odpowiedź udzielona na pytanie nr 9 – „TAK”.



Rysunek 7. Okno apletu w trakcie prowadzenia dialogu użytkownika z systemem

Pytanie nr 10 – „Czy występuje objaw: Za mały ciężar właściwy?”

Odpowiedź udzielona na pytanie nr 10 – „NIE”.

Pytanie nr 11 – „Czy występuje objaw: Za mała wytrzymałość?”

Odpowiedź udzielona na pytanie nr 11 – „TAK”.

Pytanie nr 12 – „Czy istnieją jeszcze jakieś objawy?”

Odpowiedź udzielona na pytanie nr 12 – „NIE”.

Zgodnie z rys. 7 prowadzony jest dialog systemu z użytkownikiem, który odpowiada na pytania dotyczące objawów wystąpienia wady laminatu. Udzielanie odpowiedzi przebiega na zasadzie zaznaczenia odpowiedniego pola („TAK” lub „NIE”) w oknie apletu i naciśnięciu przycisku „Dalej”. W przypadku wyczerpania wszystkich pytań lub udzieleniu odpowiedzi negatywnej na pytanie „Czy istnieją jeszcze jakieś objawy?” kończy się proces wnioskowania i przedstawiona zostaje diagnoza w postaci listy wad wraz z dodatkowymi danymi dotyczącymi przyczyn powstania oraz metod napraw i zapobiegania każdej z tych wad.

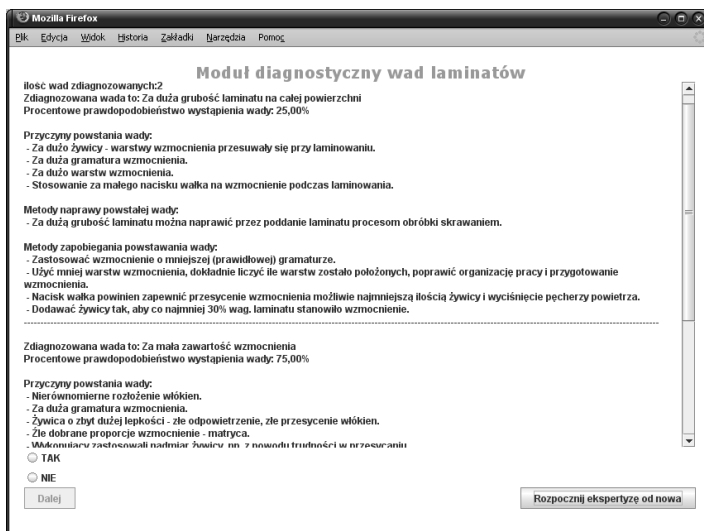
Na rys. 8 przedstawiono przykładowy wynik ekspertyzy po zakończeniu dialogu.

Zostały zdiagnozowane dwie wady kompozytów polimerowych. System ocenił prawdopodobieństwo wystąpienia każdej z nich oraz podał szczegółowe dane odnośnie przyczyn powstania, metod naprawy oraz metod zapobiegania powstawania tych wad. Zdiagnozowane przez system wady to:

- „Za duża grubość laminatu na całej powierzchni”  
Procentowe prawdopodobieństwo wystąpienia tej wady – 25%.
- „Za mała zawartość wzmocnienia”  
Procentowe prawdopodobieństwo wystąpienia tej wady – 75%.

System ocenił, że bardziej prawdopodobną wadą, jaka mogła powstać w badanym laminacie jest „Za mała zawartość wzmocnienia”.





Rysunek 8. Okno apletu przedstawiające wynik ekspertyzy

### 3. Podsumowanie

Moduł ekspertyz wad laminatów jest adresowany do firm produkujących te tworzywa, gdzie jego stosowanie może przyczynić się do zwiększenia jakości stawianych diagnoz, zmniejszenia czasu podejmowania decyzji i co najistotniejsze do zmniejszenia kosztów związanych z przeprowadzaniem drogiej, tradycyjnych ekspertyz. Zaprojektowany system nie wymaga dodatkowego oprogramowania i umożliwia dostęp do zasobów poprzez przeglądarkę internetową. Istotną cechą jest ponadto otwartość systemu i możliwość rozbudowy bazy wiedzy o nowe przypadki wad w oparciu o grupę autoryzowanych ekspertów dziedzinowych. Z punktu widzenia inżynierii wiedzy interesującym aspektem zaprojektowanego systemu jest możliwość swobodnej integracji mechanizmu wnioskowania JESS z aplikacją bazodanową, działającą w dynamicznym środowisku internetowym.

### Bibliografia

- [1] ASTM D 2563 – 70 Standard Practice for Classifying Visual Defects in Glass-Reinforced Plastics Laminate Parts
- [2] ASTM D 2562 – 70 Standard Practice for Classifying Visual Defects in Parts Molded from Reinforced Thermosetting Plastics
- [3] ASTM D 4385 – 84a Standard Practice for Classifying Visual Defects in Thermosetting Reinforced Plastics Pultruded Products
- [4] Białko M., Musiał K. *Programowanie w języku CLIPS 6.0*. Politechnika Koszalińska 2001.
- [5] Friedman-Hill E. *Jess in action*. Manning Publications Co., Greenwich 2003
- [6] Gorący K., Jaszczak S. *Analiza możliwości zastosowania systemów ekspertowych w opracowaniu technologii, ocenie jakości i diagnostyce kompozytów polimerowych*. 9 Sesja Naukowa Wydziału Informatyki Politechniki Szczecińskiej 2004

- [7] Knosala R. *Zastosowania metod sztucznej inteligencji w inżynierii produkcji*. WNT, Warszawa 2002
- [8] Królikowski W., Kłosowska-Wońkiewicz Z., Penczek. P. *Żywyce i laminaty poliestrowe*. WNT, Warszawa 1986
- [9] Luger G.F., Stubblefield W.A. *Artificial Intelligence and the Design of Expert Systems*, The Benjamin/Cummings Publishing Company Inc., Redwood 1989
- [10] Michaeli W. *Einführung in die Kunststoffverarbeitung*. Carl Hanser Verlag, Wien 1992
- [11] Mulawka J. J. *Systemy ekspertowe*. WNT, Warszawa 1996
- [12] Niederliński A. *Regulowe systemy ekspertowe*. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2000

# Recognition of Psychological Characteristics from Face

Ekaterina Kamenskaya<sup>1</sup>, Georgy Kukharev<sup>2</sup>

<sup>1</sup>*Department of Computer Software Environment,  
St.-Petersburg Electrotechnical University*

<sup>2</sup>*Faculty of Computer Science and Information Technology,  
Szczecin University of Technology*

## Abstract:

*The nature of computer vision causes the fact that not only computer science researchers are interested in it, but neuroscientists and psychologists, too. One of the main interests for psychology is identification of person's psychological traits and personality types which can be accomplished by different means of psychological testing: questionnaires, interviews, direct observations, etc. Though that is a general tendency of people to read character into a person's physical form, especially face. In relation to psychological characteristics recognition, face provides researchers and psychologists with instrument of obtaining information about personality and psychological traits that would be much more objective than questionnaires and neuropsychological tests and could be obtained remotely using person's facial portrait, with no need for personal involvement. The paper describes approaches to psychological characteristics recognition from facial image such as physiognomy, phase facial portrait, ophthalmogeometry, and explains the need in automating it.*

## Keywords:

*psychometrics, psychological characteristics, personality, Myers-Briggs typology, ophthalmogeometry, physiognomy, pattern recognition, face recognition, facial expression recognition*

## 1. Introduction

A contemporary definition for personality is offered by Carver and Scheier: "Personality is a dynamic organization, inside the person, of psychophysical systems that create a person's characteristic patterns of behavior, thoughts, and feelings" [1]. In other words, personality is a complex combination of traits and characteristics that determines our expectations, self-perceptions, values and attitudes, and predicts our reactions to people, subjects and events.

According to the Diagnostic and Statistical Manual of the American Psychiatric Association [2], personality traits are "enduring patterns of perceiving, relating to, and thinking about the environment and oneself that are exhibited in a wide range of social and personal contexts." Traits and characteristics are the same

thing, as a trait is a distinguishing characteristic, feature or quality. Theorists generally assume that a) traits are relatively stable over time, b) traits differ among individuals, and c) traits influence behavior. Many psychologists have studied personality traits, and have attempted to identify or define all of the human traits. Some examples of personality traits identified and studied by psychologists are: warmth, emotional stability, independence, dominance, impulsivity, sensitivity, imagination, introversion-extroversion, suspiciousness, etc.

There is continuing debate about how many different personality traits exist. The most popular model of psychological traits is a five-dimension personality model named as the “Big Five” and proposed by Lewis Goldberg: Extraversion, Neuroticism, Agreeableness, Conscientiousness, Openness to experience.

The difference between personality type and personality trait should be considered. Personality types are distinguished from personality traits, which come in different levels or degrees. According to type theories, for example, there are two types of people, introverts and extraverts. According to trait theories, introversion and extraversion are part of a continuous dimension, with many people in the middle.

Identification of psychological characteristics is the task widely used in theoretical and practical psychological research, education, coaching, career guidance and hiring process, business and political affairs, psychotherapeutic diagnostics, self-exploration and awareness, etc. Teachers evaluate student’s personality traits and cognitive abilities to know how to represent material better and how to establish communication and learning process in more efficient way. Lawyers are beginning to use personality testing for criminal behavior analysis, litigation profiling, witness examination and jury selection. Medical staff analyses personality characteristics and observes patient’s psychological state in regard of its influence on medical treatment process. Even websites design and software interfaces are now developed based on knowledge of user’s personalities and their preferences.

It’s hard to underestimate the role of identification of psychological characteristics in modern society, as everyone needs to know people’s psychological traits to understand or predict their reactions to various subjects and situations.

## **2. Face analysis and other methods for psychological characteristics recognition**

The question of methods and instruments which may be used for effective psychological characteristics identification exists. To measure human characteristics or identify personality types psychological researchers apply psychometrics which is the field of study including the theory and technique of psychological measurement primarily concerned with the study of differences between individuals. Psychometrics involves such research tasks as the construction of instruments and procedures for measurement; and the development of theoretical approaches to measurement. For instance, the study states that a useful psychological measure must be both valid (i.e., actually measures what it claims to measure) and reliable (i.e., internally consistent or give consistent results over time).

The most accepted means of measuring personality using psychometrics study is psychological questionnaires and tasks. There are two major types of such personality tests. Projective tests assume that personality is primarily unconscious and assess an individual by how he or she responds to an ambiguous stimulus. The idea is that unconscious needs will come out in the person's response. Objective tests assume that personality is consciously accessible and measure it by self-report questionnaires. Research on psychological assessment has generally found that objective tests are more valid and reliable than projective tests. Examples of testing questionnaires and tasks are Holland Codes, Rorschach test, Minnesota Multiphasic Personality Inventory, Myers-Briggs Type Indicator, Enneagram Type Indicator, NEO PI-R, Thematic Apperception Test, Kelly's Repertory Grid, Online Depression Screening Test, Online Screening for Anxiety and so on [3].

Except psychological questionnaires and tasks, interviews and direct observation tests are applied. Interviews often contain some observation and questions about person's past, abilities and attitudes. Direct observation of people as they complete activities may be with clinical purpose, such as to observe the nature of a parent-child interaction in order to understand a relational disorder. The Parent-Child Interaction Assessment-II [4] is an example of a direct observation procedure that is used with school-age children and parents: the parents and children are videotaped playing at a make-believe zoo. Direct observation procedures are also used in research, for example to explore sequences of behavioral interaction.

There is a range of approaches to measuring personality and psychological characteristics which are based on human physical appearance. They are described below:

### **Evaluation based on face**

The face plays a crucial role in human social cognition. Facial expressions are important signals of internal states – emotions and intentions. Humans also see in the face signals of internal qualities that are more stable over time, like attractiveness as a mate [5,6,7] or dominance [8,9,10], etc. According to Liggett [11], “There can be little doubt that the face plays a crucial part in our everyday assessment of our fellows. Not only does it enable us to identify transient emotions – flashes of pleasure and rage, disappointment and hatred – it can also help us to make useful judgments about more durable and lasting qualities of personality and character”. Judging personality from face comes from ancient times [12,13] to nowadays. The belief that the face reveals information about underlying character cuts across national, cultural and geographical boundaries. Several classical Greek and Roman scholars, including Plato and Aristotle, argued that faces contain clues about people's underlying personalities and dispositions. For over 2500 years many notable philosophers, historians and physicians openly supported the idea that a person's disposition is reflected in their face. Widespread interest in physiognomy – the study of the face and its relationship to human ability, potential and character – peaked at the end of the eighteenth century when a physician and pastor named




Johann Kaspar Lavater produced a formal classification system and set of rules specifying the relationship between the face and the mind [14]. Lavater believed that character is assessed just as well from the shape and size of the forehead, for example, as from observations of behavior. Indeed, eighteenth and nineteenth century courts commonly used physiognomy to assess immoral tendencies in suspected criminals.

Among contemporary applications of face analysis not only psychological interpretations are accomplished, but also medical results are obtained. For example, more than 700 genetic issues influencing facial structure and facial features are known, and special software for disease identification from face had been developed [15,16].

### Body types

William Sheldon classified personality according to body type [17]. He called this a person's somatotype and identified three main somatotypes shown in Table 1.

Table 1. Sheldon's somatotypes and character interpretations

Sheldon's Somatotype	Character	Shape	Picture
Endomorph [viscerotonic]	Relaxed, sociable, tolerant, comfort-loving, peaceful	Plump, buxom, developed visceral structure	
Mesomorph [somatotonic]	Active, assertive, vigorous, combative	Muscular	
Ectomorph [cerebrotonic]	Quiet, fragile, restrained, non-assertive, sensitive	Lean, delicate, poor muscles	



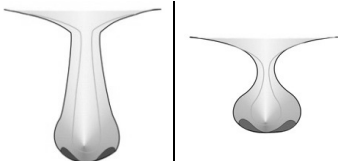

Person is rated on each of these three dimensions using a scale from 1 (low) to 7 (high) with a mean of 4 (average). Therefore, for example, a person who is a pure mesomorph would have a score of 1-7-1.

In Ayurvedic medicine (used in India since ~3000 BC) there are three main metabolic body types (doshas) – Vata, Pita, & Kapha – which in some way correspond to Sheldon's somatotypes. Body types have been criticized for very weak empirical methodology and are not generally used in Western psychology (they are used more often in alternative therapies and Eastern psychology and spirituality).

**Complex physical appearance evaluation**

This is approach of evaluation of face and body parts in complex, and it is considered to be physiognomy too. Physical appearance characteristics such as appearance of some facial features, of the skull, shoulders, hands, fingers, legs, type of mimics and voice may define personality traits. For example, it’s used in socionics (see Table 2) that is a branch of psychology based on Carl Jung’s work on Psychological Types. Moreover, many socionics experts use the visual method of personality characteristics identification as a main method for personality traits and types recognition.

Table 2. Example of some outer appearance characteristics and their interpretation

OUTER APPEARANCE					
No.	Physical character	Sensing		Intuitive	
01	The form of bones and muscles	Short and thick, muscles are pronounced 		Lengthy and thin, muscles aren't pronounced 	
02	Form of the nose	Sensing + Logical	Sensing + Ethical	Intuitive + Ethical	Intuitive + Logical
		«triangle with peak on the top» Horizontal line in the nose bridge. 		«triangle with peak on the top» 	

**Neuropsychological tests**

Around the 1990s, neuroscience entered the domain of personality psychology. It introduced powerful brain analysis tools like Electroencephalography (EEG), Positron Emission Tomography (PET), Functional Magnetic Resonance Imaging (fMRI) and structural MRI including diffusion tensor imaging (DTI) to this study. One of the founders of this area of brain research is Richard Davidson of the University of Wisconsin-Madison [18]. Davidson’s research lab has focused on the role of the prefrontal cortex and amygdala in manifesting human personality. In particular, this research has looked at hemispheric asymmetry of activity in these

regions. Neuropsychological studies have illustrated how hemispheric asymmetry can affect an individual's personality.

In contemporary psychological research there should be an instrument which would provide a maximum amount and type of objective/unbiased information about personality in as short a time as possible, preferably with no participation of person whose characteristics are identified. Comparison of approaches to identification of psychological characteristics described above is represented in Table 3.

Table 3. Some comparison of approaches to identification of psychological characteristics

Criterion	Approaches			
	Psychological questionnaires	Interview, direct observation	Face, body evaluation	Neuropsychological tests
Easy and not time-consuming for person who is tested	-	-	+	-
Person may not participate in testing process	-	-	+	-
High validity and reliability	+ [19]	-	?	-
Practically no possibility for respondent faking	-	-	+	-
No need in expensive hi-tech hardware	+	+	+	-

In psychological testing there is considerable problem that respondents are often able to distort their responses. This is particularly problematic in employment contexts and other contexts where important decisions are being made and there is an incentive to present oneself in a favorable manner. Social desirability is a tendency to portray self in a positive light, and faking bad also happens, that is purposely saying 'no' or looking bad if there's a 'reward' (e.g. attention, compensation, social welfare, etc.). Work in experimental settings [20,21] has shown that when student samples have been asked to deliberately fake on a personality test, they demonstrated that they are capable of doing this.

Though several strategies have been adopted for reducing respondent faking, this is still a problem for such traditional psychological testing instruments like questionnaires, interviews, direct observations. Surprisingly, neuropsychological tests are prone to respondent faking, too [22,23]. Faking response styles include faking bad (malingering), faking good (defensiveness), attempts at invalidation, mixed responding (faking good and bad), and a fluctuating, changing style that occurs within one evaluation session. These response styles lead to getting incorrect results.

Concerning face and facial features, faking becomes much more complicated: it's impossible to change the shape of a nose or cheekbones just when person wants. Besides, it is often unknown to a holder what his/her face reveals exactly. Theoretically people can "fake" facial features intentionally changing their shape,



color, texture, for instance, using plastic surgery, and identifying personal psychological characteristics becomes much harder in this case, though it may be also accomplished.

Face is the first subject that is unique for people and used for people recognition. Thus, face is the most available means of evaluation among other instruments based on questionnaires, interviews, neuropsychological tests. People in general may not participate in testing process, identification of personality characteristics may be done remotely, even by exterior parties.

Summarizing, face provides researchers and psychologists with instrument of obtaining information about personality and psychological traits that would be much more objective than questionnaires and neuropsychological tests (as we can't change facial features just when such desire appears) and could be obtained remotely using person's facial portrait, with no need for personal involvement.

If such instrument is working automatically (system gets facial portrait, processes it and in result gives out information about personality characteristics) and has straight-forward layout, then: 1) psychological testing becomes more accurate, fast, objective and available for different kinds of research and applications; 2) deep knowledge in interpretation of facial features, which is rather rare in modern society, isn't needed to administer and use the instrument. Methods and algorithms originally developed for face detection, face recognition and facial expression recognition research fields as well as contemporary trends (applying standard face images, multimodality, three-dimensionality) should be applied and adjusted to so-called Automatic Psychological Characteristics Recognition from Face. From its side, Automatic Recognition of Psychological Characteristics from Face is believed to bring scientific benefits to face recognition, facial expression recognition, face animation, face retrieval, etc., and finally contribute to development of human-computer interaction on higher level. Thus, the relations between such research areas as face recognition, facial expression recognition and psychological characteristics recognition are mutually beneficial.

### **3. Approaches to psychological characteristics recognition from face**

There are three main approaches to psychological characteristics recognition from face: physiognomy, phase facial portrait and ophthalmogeometry, see Fig.1. The first originally interprets different facial features, the second works with angles of facial features and facial asymmetry, and the third extracts and interprets eye region parameters. Methods developed for these approaches are described below.

**Physiognomy** is a theory based upon the idea that the assessment of the person's outer appearance, primarily the face, facial features, skin texture and quality, may give insights into one's character or personality. Physiognomy has flourished since the time of the Greeks (Empedocles, Socrates, Hippocrates and Aristotle), amongst the Chinese and Indians, with the Romans (Polemon and Adamantius), in the Arab world (including Avicenna), and during the European renaissance (Gerolamo Cardano and Giovanni Battista della Porta). It faded in

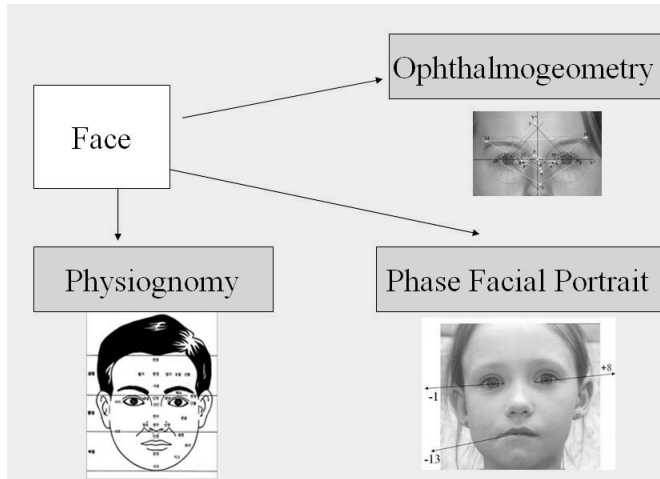


Figure 1. Approaches to psychological characteristics recognition from facial portrait

popularity during the 18th century, was eclipsed by phrenology in the 19th and has been refreshed by personologists in the 20th century.

During 20th century attempts had been made to perform scientific experiments concerning validity of different facial features interpretations and high accuracy results had been claimed [24], though they are mostly aren't accepted by official science [25]. At the same time, science step by steps proves some physiognomy beliefs. For instance, correlations have been established between IQ and cranial volume [26,27,28,29]. Testosterone levels, which are known to correlate with aggressiveness, are also strongly correlated with features such as finger-length ratios and square jaws [30,31].

Interpretation of facial features based on physiognomy has been implemented into psychological characteristics diagnosis tools such as “Visage” Project [32] developed by Dr. Paul Ekman and “Digital physiognomy“ software [33] developed by Uniphiz Lab.

“Visage” is a project for collecting and organizing information about relatively permanent facial features. It includes methods for storing, retrieving, and inspecting the data. Visage is a unique database schema for representing physiognomy and the interpretation of physiognomic signs. The Visage demonstration application illustrates limited variations of some facial features in the following categories: forehead and eyebrows (see the Fig.2), eyes and eyelids, nose, mouth and jaw, cheeks, chin, ears. User should select features that are distinctive about the face that is going to be interpreted and then click the “Get...” button. The application retrieves information from the database relevant to description of physiognomy, including an estimation of the accuracy of the sources of information.

“Digital physiognomy“ software determines a person’s psychological characteristics based on temperament types, intellect, optimism – pessimism, conformism – adventurism, egoism – altruism, philanthropy – hostility, laziness, honesty, etc.,

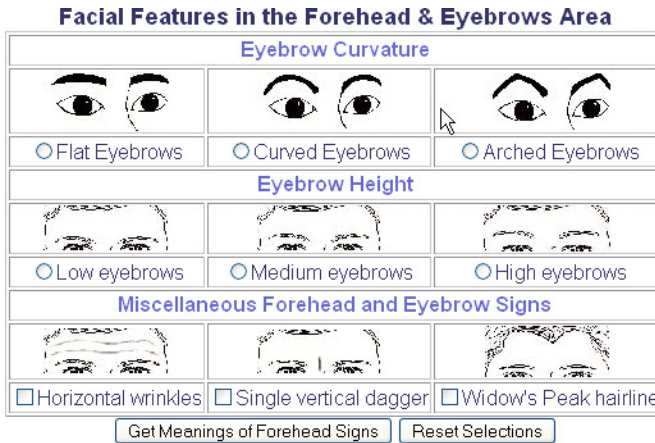


Figure 2. Example of the table and interface of Visage demonstration application: facial features in the forehead and eyebrow area [34]

and then presents a detailed person's character analysis in a graphic format. The tool works like a police sketch (photo robot), so user has to select different parts of the person's face, and doesn't need to have a person's photograph, see Fig. 3. It's claimed that only the facial features that can be interpreted with high accuracy were used, and the confidence factor is calculated for each interpretation by the tool. It should be noted that "Digital physiognomy" tool also uses visual systematic classification of 16 personality types based upon Myers-Briggs typology, see Fig. 4.

"Visage" and "Digital Physiognomy" projects are some of the first attempts to develop physiognomic database and use modern technology for physiognomic interpretations. In spite of having value for psychological diagnosis based on physiognomy, both projects use manual selection of facial features, and thus, can't be used extensively and applied in scientific research.

Phase facial portrait approach to psychological characteristics recognition from facial portrait is primarily based on calculating of angles of facial features lines directions, as shown on Fig.5. Video-computer psychological diagnosis and correction method [35] had been invented by professor Avtandil Anuashvili. It's remote method for personal psycho-type identification, method of one of the brain hemispheres dominance identification, method of psycho-diagnostics and psycho-correction. The method is based on the thesis that face is an informational background projecting in biological and psychological terms the results of joint brain hemispheres functioning. The system of 49 types of people in coordinates "left-side thinker – right-side thinker", "psychologically stable personality – instable personality" had been created and correlated to other personal typologies, see Fig. 6.

Software developed to apply video-computer psychological diagnosis and correction method accepts facial image and determines: 1) dominance of one of the brain hemispheres on the basis of a difference of amplitudes of oscillatory processes occurring in right and left hemispheres and 2) degrees of a coordination of

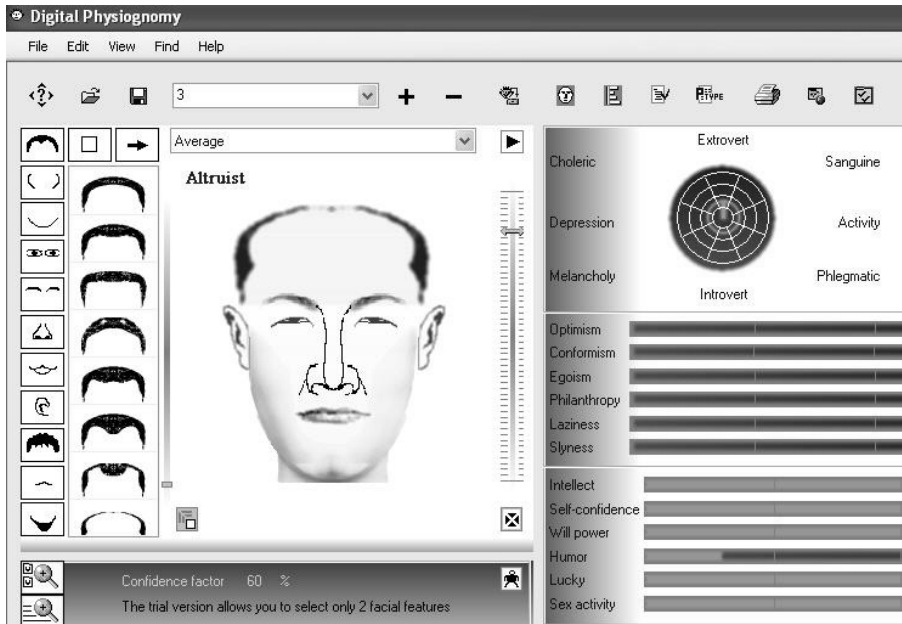


Figure 3. Fragment of “Digital physiognomy“ software

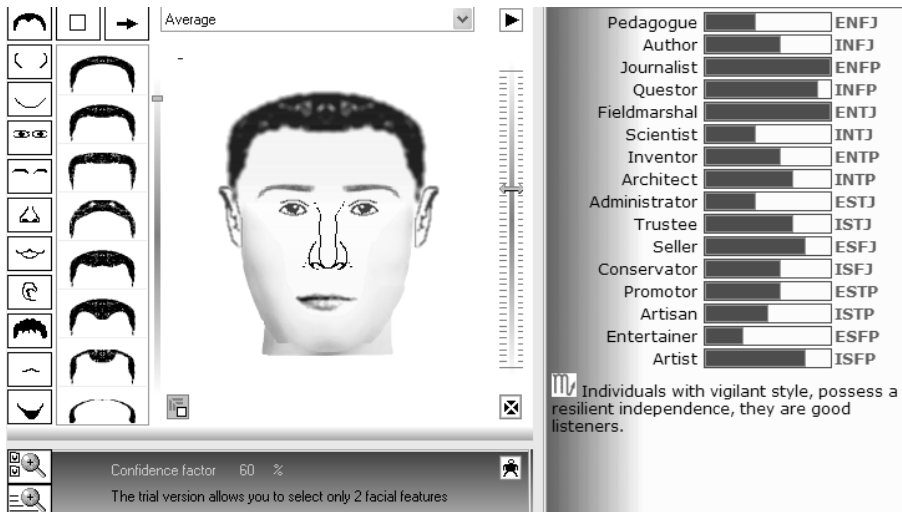


Figure 4. Socionics visual interpretation by “Digital physiognomy“ based upon Myers-Briggs typology

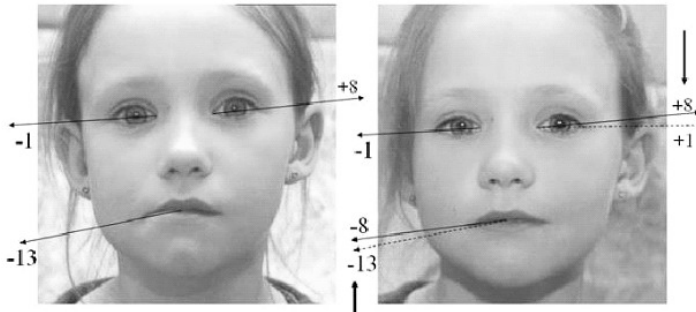


Figure 5. The representation of phase facial portrait concept [37]

these oscillatory processes among themselves [36]. Based on these meanings, the software classifies a given person to one of 49 psychological types and gives out complete personal characteristic, professional characteristic and recommendations on harmonization, effective interaction with other people and environment.

Video-computer psychological diagnosis and correction method is one of the primary instrumental psychological methods concerning examination and usage

*TYPOLOGY on ANUASHVILI*  
49 types (usual people)  
The numbers mean, - %

H - Harmony of a person  
L - Logic  
I - Intuition  
S - Stability  
D - Destability

		Stable							
		Air Future, creation ↑ S							
		S-Stability of relation between hemispheres (Stability of mentality)							
Gladness								Calmness	
Logical stable		L80S80	L70S80	L60S80	LIS80	I60S80	I70S80	I80S80	Intuitive stable
LS Sanguine "Tactician"		H=65 (ESTJ)	H=70	H=75 (ENTJ)	H=80	H=75 (ISTJ)	H=70	H=65 (INTJ)	IS Phlegmatic "Strategist"
		L80S70	L70S70	L60S70	LIS70	I60S70	I70S70	I80S70	
		H=60	H=60	H=65	H=70	H=65	H=60	H=60	
		L80S60	L70S60	L60S60	LIS60	I60S60	I70S60	I80S60	
		H=55 (ESTP)	H=55	H=55 (ENTP)	H=60	H=55 (ISTP)	H=55	H=55 (INTP)	
Fire Present								Water Past	
Logical L		L80SD	L70SD	L60SD	LISD	I60SD	I70SD	I80SD	Intuitive
L-Domination of the left hemisphere (logic, mind)		H=50	H=50	H=50	H=50	H=50	H=50	H=50	I-Domination of the right hemisphere (Intuition, Spirit)
		L80D60	L70D60	L60D60	LID60	I60D60	I70D60	I80D60	
		H=45 (ESFJ)	H=45	H=45 (ENFJ)	H=40	H=45 (ISFJ)	H=45	H=45 (INFJ)	
		L80D70	L70D70	L60D70	LID70	I60D70	I70D70	I80D70	
		H=40	H=40	H=35	H=30	H=35	H=40	H=40	
LD Anger		L80D80	L70D80	L60D80	LID80	I60D80	I70D80	I80D80	Melancholy ID
Logical destable		H=35 (ESFP)	H=30	H=25 (ENFP)	H=20	H=25 (ISFP)	H=30	H=35 (INFP)	Melancholic "Dreamer"
		Ground Chaos, destruction ↓ D							
		D-destability of relation between hemispheres (Destability of mentality)							
		Destable							

Figure 6. Table of 49 psychological types system invented used for video-computer psychological diagnosis and correction; here L – logic (practical mind), I – intuition (spirit, principles), S – stability, D – destability [38]

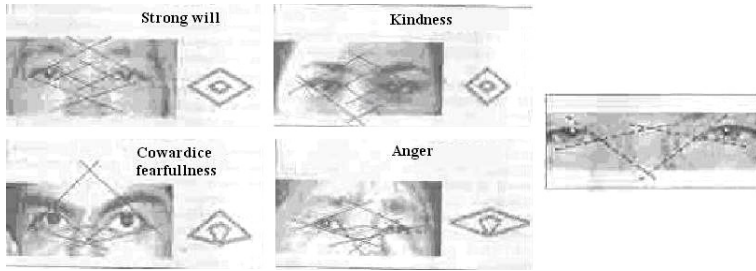


Figure 7. Translated picture from Muldashev's book [39]: here two parameters of facial eye region are used for recognition of some basic psychological traits, e.g. strong will and fearfulness, etc.

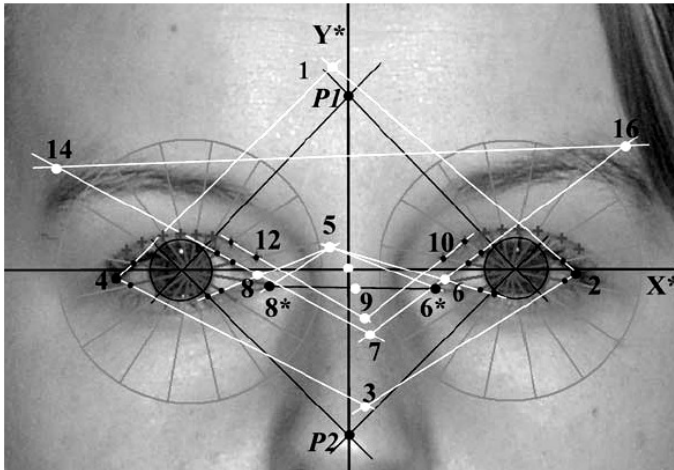


Figure 8. Ophthalmogeometrical pattern extraction [40]

of brain asymmetry phenomena and face asymmetry. Although Anuashvili claims that application developed for video-computer psychological diagnosis and correction method is entirely automated, practically it may be considered to be semi-automated as manual selection of facial points on image is required. This limits usage of such application for extensive research and other purposes.

Concerning ophthalmogeometry approach, it is based on idea that person's emotional, physical and psychological states can be recognized by 22 parameters of an eyes part of the face [39], see Fig. 7. Ophthalmogeometry phenomenon has been discovered by prof. Ernst Muldashev. Apart from other interesting facts, E. Muldashev has found that in 4-5 years after birth the only practically constant parameter of human body is the diameter of the transparent part of cornea which equals  $10 \pm 0,56$  mm. He also represented an idea that ophthalmogeometrical pattern is unique for people. The procedure of this pattern identification and calculation is described by Leonid Kompanets [40], see Fig. 8.

Ophthalmogeometry is based on interesting ideas and may be applied to psychological, medical research as well as to biometrics, though this is not very deeply investigated area of facial analysis which primarily needs automation of ophthalmogeometric pattern extraction and further investigation.

## 4. Conclusion

The paper represents general idea that face provides researchers and psychologists with objective instrument of obtaining information about personality and psychological traits. An up-to-date survey of approaches and methods in psychological characteristics recognition from facial image is provided.

In perspective new research task of automating procedures in applications of psychological characteristics recognition from face should be explored. Various approaches and methods developed within face recognition, facial expression recognition, face retrieval, face modeling and animation may be applied and adjusted for recognition of psychological characteristics from face. Undeniably, such automated system of psychological characteristics recognition from face will get countless psychological, educational, business applications. It may be used also as part of medical systems: 1) patient's psychological state and traits influence the process of medical treatment, and it should be taken into consideration and researched; 2) patient's psychological characteristics should be taken into account to reflect and construct the psychosomatic model of disease in the environment, which includes biological, psychological, and social factors.

## References

- [1] Carver C. S., Scheier M. F. *Perspectives on personality (4th ed.)* Boston: Allyn and Bacon, 2000, page 5.
- [2] DSM, *Diagnostic and Statistical Manual of Mental Disorders*, can be found at <http://www.psych.org/research/>.
- [3] Hampson S. E. *Advances in Personality Psychology*. Psychology Press, 2000.
- [4] Holigrocki R. J., Kaminski P. L., Frieswyk S. H. (2002). *PCIA-II: Parent-Child Interaction Assessment Version II*. Unpublished manuscript, University of Indianapolis. (Update of PCIA Tech. Rep. No. 99-1046. Topeka, KS: Child and Family Center, The Menninger Clinic.) (Available from Dr. Richard J. Holigrocki or Dr. Patricia L. Kaminski).
- [5] Nigel Barber. *The evolutionary psychology of physical attractiveness: Sexual selection and human morphology*. Ethology and Sociobiology, Volume 16, Issue 5, September 1995, pages 395-424.
- [6] John P. Swaddle, Innes C. Cuthill. *Asymmetry and Human Facial Attractiveness: Symmetry May not Always be Beautiful*. Proceedings: Biological Sciences, Vol. 261, No. 1360 (Jul. 22, 1995), pages 111-116.
- [7] Thomas R. Alley, Michael R. Cunningham. *Averaged faces are attractive, but very attractive faces are not average*. Psychological Science 2 (2), 1991, pages 123-125.
- [8] Leslie A. Zebrowitz, Gillian Rhodes. *Sensitivity to "Bad Genes" and the Anomalous Face Overgeneralization Effect: Cue Validity, Cue Utilization, and Accuracy in Jud-*

- ging Intelligence and Health. Journal of Nonverbal Behavior Volume 28, Number 3 / September, 2004, pages 167-185.
- [9] Caroline F. Keating. *Gender and the Physiognomy of Dominance and Attractiveness*, Social Psychology Quarterly, Vol. 48, No. 1 (Mar., 1985), pages 61-70.
- [10] Ulrich Mueller, Allan Mazur. *Facial Dominance of West Point Cadets as a Predictor of Later Military Rank*. Social Forces, Vol. 74, No. 3 (Mar., 1996), pages 823-850.
- [11] J. Liggett, *The human face*. New York: Stein and Day, 1974, page 276.
- [12] *Physiognomics*, attributed to Aristotle. Cited in J.Wechsler (1982), *A human comedy: Physiognomy and caricature in 19th century Paris* (p.15). Chicago: University of Chicago Press.
- [13] A. Brandt. *Face reading: The persistence of physiognomy*. Journal Psychology Today, 1980, December, page 93.
- [14] Sibylle Erle. *Face to Face with Johann Caspar Lavater*, Literature Compass 2 (2005) RO 131, pages 1 -4.
- [15] Stefan Boehringer, Tobias Vollmar, Christiane Tasse, Rolf P Wurtz, Gabriele Gillessen-Kaesbach, Bernhard Horsthemke and Dagmar Wiczorek. *Syndrome identification based on 2D analysis software*. European Journal of Human Genetics (2006), pages 1-8.
- [16] Hartmut S Loos, Dagmar Wiczorek, Rolf P Würtz, *Christoph von der Malsburg and Bernhard Horsthemke*. Computer-based recognition of dysmorphic faces, European Journal of Human Genetics (2003) 11, pages 555-560.
- [17] Irvin L. Child, William H. Sheldon. *The correlation between components of physique and scores on certain psychological tests*. Journal of Personality, Vol. 10, Issue 1, September 1941, page 23.
- [18] Richard Davidson, Ph.D. Vilas Professor of Psychology and Psychiatry. Can be found at <https://psychiatry.wisc.edu/faculty/FacultyPages/Davidson.htm>.
- [19] *The Validity of Graphology in Personnel Assessment*. Psychological Testing Centre. Found at [www.psychtesting.org.uk](http://www.psychtesting.org.uk), November 1993 reviewed April 2002.
- [20] Chockalingam Viswesvaran, Deniz S. Ones. *Meta-Analyses of Fakability Estimates: Implications for Personality Measurement*, Educational and Psychological Measurement, Vol. 59, No. 2, 1999, pages 197-210.
- [21] Deniz S. Ones, Chockalingam Viswesvaran, Angelika D. Reiss. *Role of Social Desirability in Personality Testing for Personnel Selection: The Red Herring*. Journal of Applied Psychology, 1996. Vol. 81, No. 6, pages 660-679.
- [22] Hall, Harold V.; Poirier, Joseph G.; Thompson, Jane S. *Detecting deception in neuropsychological cases: toward an applied model*. From: The Forensic Examiner, 9/22/2007.
- [23] Allyson G. Harrison, Melanie J. Edwards and Kevin C.H. Parker. *Identifying students faking ADHD: Preliminary findings and strategies for detection*. Archives of Clinical Neuropsychology. Volume 22, Issue 5, June 2007, pages 577-588
- [24] Naomi Tickle. *You Can Read a Face Like a Book: How Reading Faces Helps You Succeed in Business and Relationships*, Daniels Publishing, 2003.
- [25] Robert Todd Carroll. *The Skeptic's Dictionary: A Collection of Strange Beliefs, Amusing Deceptions, and Dangerous Delusions*. Wiley; 1st edition (August 15, 2003)
- [26] J. Philippe Rushton, C. Davison Ankney. *Brain size and cognitive ability: Correlations with age, sex, social class, and race*. Psychonomic Bulletin & Review, 1996, 3 (1), pages 21-36.
- [27] Michael A. McDaniel. *Big-brained people are smarter: A meta-analysis of the re-*



- lationship between in vivo brain volume and intelligence. *Intelligence*, Volume 33, Issue 4, July-August 2005, pages 337-346.
- [28] J. Philippe Rushton. *Cranial size and IQ in Asian Americans from birth to age seven*. *Intelligence*, Volume 25, Issue 1, 1997, pages 7-20.
- [29] John C. Wickett, Philip A. Vernon, Donald H. Lee. *Relationships between factors of intelligence and brain volume*. *Personality and Individual Differences*, Volume 29, Issue 6, December 2000, pages 1095-1122.
- [30] John T. Manning. *Digit Ratio: A Pointer to Fertility, Behavior, and Health*. Rutgers University Press, 2002.
- [31] Bernhard Fink, Karl Grammer, Philipp Mitteroecker, Philipp Gunz, Katrin Schaefer, Fred L. Bookstein, John T. Manning. *Second to fourth digit ratio and face shape*. *Proceedings of the royal society*, Volume 272, Number 1576 / October 07, 2005, pages 1995-2001.
- [32] *Visage Project for Physiognomy Data*, can be found at [http://face-and-emotion.com/dataface/visage/about\\_visage.jsp](http://face-and-emotion.com/dataface/visage/about_visage.jsp). Date: 20 January 2008.
- [33] *Digital Physiognomy Software: Match person's face to his or her character*. Can be found at <http://www.uniphiz.com/physiognomy.htm>. Date: 20 January 2008.
- [34] *Facial Features in the Forehead & Eyebrows Area*, can be found at [http://face-and-emotion.com/dataface/visage/visage\\_forehead.jsp](http://face-and-emotion.com/dataface/visage/visage_forehead.jsp). Date: 20 January 2008.
- [35] Anuashvili Avtandil. *Fundamentals of Psychology. Scientific, Philosophic and Spiritual Fundamentals of Psychology*. The Institute for Control Problems Press, Moscow, 2001 (In Russian).
- [36] *Video-computer system for psychodiagnosis and psychocorrection is the first objective tool for psychotherapist and psychologist*, can be found at <http://www.anuashvili.ru/Video-computer%20psychodiagnostics/video-compute.htm>. Date: 20 January 2008.
- [37] *The audio-presentation for video-computer psychological diagnosis and correction*, can be found at <http://www.tsarvogolove.ru> (in Russian). Date: 20 January 2008.
- [38] Avtandil Anuashvili's website, can be found at <http://www.anuashvili.ru/>. Date: 20 January 2008.
- [39] Muldashev Ernst R. *Whom did we descend from?*, "OLMA-PRESS", Moscow 2002 (In Russian)
- [40] Leonid Kompanets. *Biometrics of asymmetrical face*. From Biometric Authentication: First International Conference, By David Zhang, Anil K. Jain, ICBA 2004 , pages 67-73.



# Opracowanie rozproszonego klucza steganograficznego

Jerzy Korostil, Łukasz Nozdrzykowski

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*This paper presents a review of a stego key based algorithm used for hiding information into a digital image. We also present a scheme of a stego key used to threshold classification of image blocks and selection of coefficients of DCT transform within which information is hidden.*

## Słowa kluczowe:

*steganography, stego keys, information hiding*

## 1. Wprowadzenie

Steganografia jest systemem ochrony informacji ukrywającym wiadomość w innej wiadomości, zwanej informacją nośną. Tutaj stosowanie kluczy dotyczy głównie steganografii silnej (Robust), czyli takiej, gdzie ukryta wiadomość jest ciężka do wykrycia, odczytania oraz zniszczenia [2]. W artykule przedstawiono systemy steganograficzne operujące na obrazach cyfrowych.

## 2. Użytkowanie kluczy steganograficznych

Główną składową większości systemów ochrony informacji są klucze. W kryptografii klucz definiuje się następująco: Każdy element  $e \in K$ , gdzie  $K$  to zbiór kluczy, zwany jest kluczem szyfrującym. Określa on jednoznacznie bijekcję z  $M$  (wiadomość) do  $C$  (szyfrogram). Bijekcję tą oznaczoną jako  $E_e$  określa się mianem funkcji szyfrującej. Natomiast funkcja deszyfrująca  $D_d$  jest bijekcją z  $C$  na  $M$ . Określona jest ona jednoznacznie przez klucz deszyfrujący  $d \in K$  [1]. W pierwszych algorytmach steganograficznych nie wykorzystywano kluczy. Były to proste algorytmy operujące w dziedzinie przestrzennej takie, jak LSB (najmniej znaczącego bitu). Następnie zaczęto stosować algorytmy szyfrujące wiadomość przed jej ukryciem z użyciem klucza kryptograficznego [3].

Zgodnie z pracą [4] obecnie opracowywane metody steganograficzne stosują klucze w następujących rolach:

1. Określają pozycję ukrycia wiadomości poprzez:
  - sekwencję klucza;
  - sekwencję pseudolosową wygenerowaną na podstawie klucza.
2. Rozrzucanie ukrywanej wiadomości.

## 2.1. Wykorzystanie klucza do wyznaczania pozycji ukrycia wiadomości

Klucz steganograficzny może służyć do wyznaczania miejsc ukrycia wiadomości w informacji nośnej. Mogą to być konkretne bity informacji nośnej lub wybrane współczynniki określonej transformaty (DCT, DWT). W przypadku dziedziny przestrzennej bity klucza mogą wyznaczać przykładowo wiersz i kolumnę pewnego bloku obrazu stanowiącego informację nośną.

Miejsca ukrycia wiadomości może określać tablica (maska), której wartości z określonego wiersza i kolumny porównane z blokiem informacji nośnej o tym samym rozmiarze, będą określać czy ukrywać bit wiadomości, czy nie. W tym wypadku maska jest jednocześnie kluczem steganograficznym. Taką metodę zaprezentowano w pracy [5], gdzie wiadomość ukrywana jest we współczynnikach transformaty DWT wyznaczonych przez użytą maskę.

Inną metodą jest zastosowanie funkcji wyznaczającej miejsca ukrycia wiadomości na podstawie podanego klucza. Przykładem może być zastosowanie generatora pseudolosowego, którego ciąg wyjściowy wyznaczy takie miejsca, a odpowiedni algorytm przedstawiono w pracy [6]. W proponowanej metodzie zastosowano generator oparty o rejestr LFSR inicjowany kluczem. Dodatkowo dla zwiększenia bezpieczeństwa do wyznaczania miejsc ukrycia wiadomości wykorzystuje się bity wiadomości ukrywanej i nośnej, które wyznaczają pojedyncze bity następnego indeksu (do ukrywania stosuje się LSB).

Wadą tego typu metod jest fakt, że wiadomość jest ukrywana we wszystkich obrazach w tych samych miejscach (przy założeniu, że klucz jest niezmienny). Klucz taki stosunkowo łatwo jest wyliczyć. Ponadto nie brane są pod uwagę lokalne własności obrazu, takie jak zróżnicowane teksturowanie. Jeżeli stosowana metoda mocno zniekształca obraz, to jeżeli użyto jako informacji nośnej obrazu o jednolitej teksturze, to fakt ukrycia wiadomości jest dość łatwy do wykrycia.

## 2.2. Klucze steganograficzne stosowane do rozrzucania ukrywanej wiadomości

Przed ukryciem wiadomość jest rozrzucana z wykorzystaniem klucza steganograficznego. W artykule [7] przedstawiono metodę wykorzystującą 3 klucze i rozpraszanie widma do ukrywania wiadomości. Pierwszy klucz służy do opcjonalnego szyfrowania wiadomości przed jej ukryciem. Drugi klucz wykorzystywany jest do generowania pseudolosowego szumu. Wiadomość ukrywana łączona jest z szumem dając wiadomość tajną. Następnie wykonywane jest właściwe ukrycie wiadomości z wykorzystaniem klucza trzeciego. Użycie rozpraszania widma zabezpiecza przed wykryciem wizualnym faktu użycia metody steganograficznej oraz odczytania ukrytej wiadomości. Sama wiadomość jest rozpraszana we wszystkich częstotliwościach silnego sygnału, jakim jest informacja nośna podczas, gdy właściwa wiadomość ukrywana jest poprzez dodawanie wartości na poziomie niższym niż poziom szumu [8].

W tych metodach trudniej jest stwierdzić fakt ukrycia wiadomości, lecz nadal wiadomość ukrywana jest w tych samych miejscach informacji nośnej. Ponadto nadal nie jest brana pod uwagę informacja o informacji nośnej podczas ukrywania.

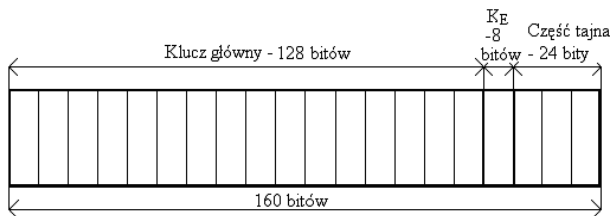
### 2.3. Klucze w metodach analizujących miejsca ukrycia wiadomości

W redukcji powstających zniekształceń na skutek ukrywania informacji metodami steganograficznymi może pomóc stosowanie badania lokalnych własności informacji nośnej. Ukrywanie wiadomości w obszarach obrazu o słabej teksturze może spowodować widoczne zniekształcenia.

W artykule [9] przedstawiono metodę opartą o transformatę kosinusową DCT, badającą energię kwantyzowanych współczynników transformaty. Dwie wartości progowe  $T_1$  oraz  $T_2$  stanowią klucz steganograficzny. Jeżeli współczynnik transformaty jest większy od wartości pierwszego progu to może on zostać użyty do ukrycia wiadomości. Dopiero wartość jego amplitudy większa od drugiego z progów pozwala na pozytywne wybranie tego współczynnika do procesu ukrywania wiadomości. W tej metodzie badane są własności obrazu, w którym ukrywana jest wiadomość, lecz progi (klucz) nie jest zmieniany w trakcie ukrywania informacji, więc dość łatwo jest go wyznaczyć.

## 3. Proponowany schemat klucza steganograficznego

W pracy [10] przedstawiono algorytm steganograficzny ukrywania informacji we współczynnikach transformaty DCT cyfrowego obrazu kolorowego. W algorytmie tym można użyć dwóch kluczy lub jednego większego. Obraz jest wstępnie dzielony na bloki o rozmiarze 32x32 piksele. Następnie dokonywana jest klasyfikacja przydatności danego bloku pod kątem ukrycia w nim wiadomości. Selekcji dokonuje się z użyciem pierwszego klucza (lub pewnej części większego klucza) i pewnego progu. Jeżeli blok jest sklasyfikowany jako przydatny, dokonuje się kolejnego podziału na bloki 8x8 pikseli. Następnie po dokonaniu na każdym z podbloków wykonywana jest transformata DCT. W każdym podbloku można ukryć jeden bit wiadomości, poprzez modyfikację dwóch wyznaczonych współczynników. Współczynniki te wyznacza się na podstawie reszty większego klucza lub drugiego mniejszego. Schemat całego klucza przedstawiono na rysunku 1.



Rysunek 1. Format klucza

Cały klucz składa się z trzech części: głównej  $K_E$  (klasyfikacji) oraz tajnej. Część klucza  $K_E$  służy do klasyfikacji bloków obrazu. Klucz główny składa się z 16 części. Każda z kolejnych części służy do wyznaczenia współczynników transformaty DCT kolejnych podbloków obrazu. Przed pierwszym użyciem klucza zaproponowano jego przetworzenie przez odpowiednią dla jego rozmiaru funkcję skrótu (tzn. dającej ciąg wyjściowy większy lub równy całej długości klucza). Ponadto przed każdorazowym użyciem całego klucza (czyli dla każdego kolejnego bloku) można go dodatkowo przetworzyć przez funkcję skrótu. Może to zapewnić większe bezpieczeństwo przed poznaniem przez adwersarza używanego klucza. Dzięki bezpiecznej funkcji skrótu i części tajnej klucza, nie jest możliwe wyznaczenie (z wyjątkiem ataku brutalnego) całego klucza na podstawie danych uzyskanych w dowolnej fazie ukrywania wiadomości. Część tajna klucza nie może zostać wyliczona na podstawie informacji wyznaczonych z obrazu oryginalnego oraz obrazu z ukrytą wiadomością. Powoduje to, że atakujący musi wyznaczać klucz dla każdego bloku obrazu, gdyż odpowiednio długa część tajna uniemożliwi wyliczenie klucza zastosowanego dla następnego bloku.

Złożoność obliczeniowa ataku brutalnego zależy od długości części klucza  $K_E$  oraz długości podkluczy stosowanych dla poszczególnych podbloków oraz ilości podbloków. W prezentowanym w pracy [2, 10] algorytmie klucz  $K_E$  ma długość 8 bitów, podobnie każdy z 16 podkluczy klucza głównego. Ponieważ ideą selekcji bloków obrazu jest zabezpieczenie przed wizualnym wykryciem wiadomości więc złożoność obliczeniowa ataku brute force jest znikoma, niemniej należałoby przetestować  $2^8$  wartości klucza  $K_E$ . Dla wyliczenia bitu wiadomości, przy założeniu silnego algorytmu steganograficznego, należy przetestować  $2^8$  podkluczy. Dla wszystkich podbloków jest to  $16 \cdot 2^8 = 2^{12}$  obliczeń do wyznaczenia 16 bitów ukrytej wiadomości. Dla zwiększenia złożoności ataku brutalnego należy zwiększyć długości podkluczy oraz zastosować inną bezpieczną metodę doboru bloków. Niemniej atak ten należy powtórzyć dla wszystkich bloków obrazu, gdyż klucz jest zmieniany dla każdego z bloków obrazu.

Na rysunku 2a) przedstawiono schemat rozproszonego zastosowania klucza steganograficznego w trakcie ukrywania informacji. Klucz ten jest podzielony na trzy części, z czego pierwsza oznaczona jako klucz główny podzielona jest na 16 części (dla każdego podbloku obrazu osobna część). Stanowi on przeciwieństwo kluczy stosowanych w algorytmach przedstawionych we wstępie (rysunek 2b), gdzie klucz był raz wprowadzany na wejściu algorytmu i w jednoznaczny sposób określał miejsce ukrycia wiadomości (dla każdego bloku obrazu ten sam klucz).

Na rysunku 3 przedstawiono przykład zastosowania klucza rozproszonego w algorytmie zaprezentowanym w pracy [2, 10]. Odtworzenie wiadomości możliwe jest z wykorzystaniem tego samego algorytmu, z tym, że zamiast ukrywania stosuje się odczytywanie wiadomości. Stosuje się następującą zasadę:

- ukryto bit 1, gdy:  $t_1 \geq t_2$ ;
- ukryto bit 0, gdy:  $t_1 < t_2$ .

Cały klucz można w prosty sposób adaptować do innych algorytmów steganograficznych odpowiednio wydłużając lub skracając część główną i  $K_E$  klucza.



Część tajna może pozostawać taka sama. Należy także dobrać odpowiednią bezpieczną funkcję skrótu dającą ciąg wyjściowy o odpowiednim rozmiarze. Ponadto w przypadku algorytmów wykorzystujących wstępną selekcję obszarów obrazu, trzeba dostosować progi pozwalające odpowiednio sklasyfikować przydatność danego bloku.

Dla algorytmu przedstawionego w pracy [2, 10] zaproponowano zastosowanie progów dla wykonania selekcji bloku obrazu. Do selekcji bloku obrazu wykorzystywany jest klucz  $K_E$  i wyliczone miary jakościowe danego bloku. Miarą tą mogą być miary statystyczne (dla dowolnej barwy składowej obrazu, w której ukrywana jest wiadomość) lub ilość punktów tworzących krawędzie w danym bloku (tu wykorzystywana jest luminancja obrazu).

Przedstawione progi zostały tak dobrane, aby wyznaczyć równowagę pomiędzy jak ilością ukrywanych danych, a widocznymi zniekształceniami w informacji nośnej spowodowanymi steganograficznym ukrywaniem danych. Odpowiednie progi dla trzech histogramów (kolorów, kontrastów i korelacji) i czterech miar statystycznych (średniej arytmetycznej, mody oraz odchyień standardowego i przeciętnego) zaprezentowanego algorytmu przedstawiono w tabeli 1. Ponadto zdefiniowano próg, który pozwala na selekcję bloku obrazu, przez zliczenie punktów tworzących krawędź i porównanie jej z wartością klucza. Blok jest pozytywnie sklasyfikowany, gdy spełniona jest zależność:

$$255 * S/1024 > 64 + K_E/255 * 128,$$

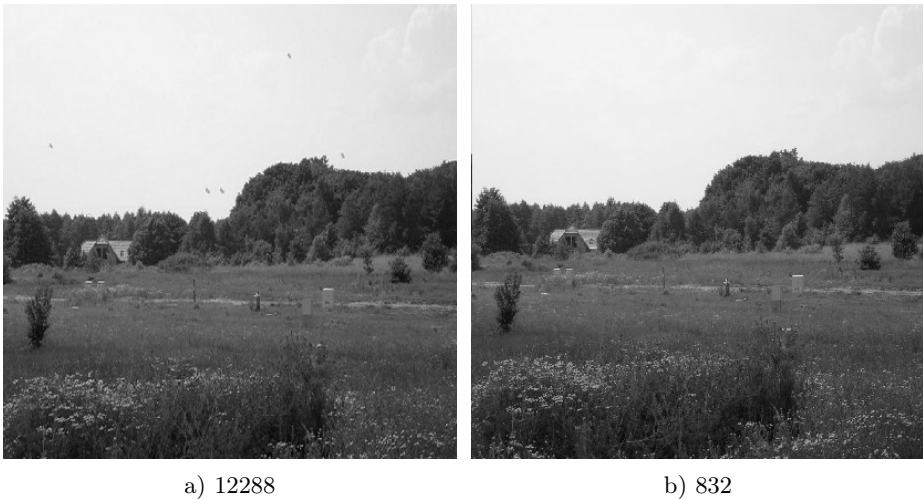
gdzie  $K_E$  – klucz,  $S$ -liczba punktów krawędziowych.

W pracy [10] przedstawiono stosowanie poszczególnych miar statystycznych dla budowania progów. Najlepiej jest stosować histogramy korelacji i kontrastów oraz progi oparte o odchylenie standardowe i przeciętne. Poniżej przedstawiono obrazy, w których ukryto wiadomość przedstawionym wyżej algorytmem steganograficznym z i bez zastosowania progów. Dla przykładu wykorzystano próg oparty o odchylenie standardowe histogramu kontrastów. Obrazy wynikowe przedstawia rysunku 4.

Tabela 1. Propozycja progów dla miar statystycznych trzech rodzajów histogramów

	Histogram kolorów	Histogram kontrastów	Histogram korelacji
<b>Średnia arytmetyczna</b>	$X_{SR} + 32 > K_E/255 * 128$	$X_{SR} < K_E/255 * 8$	$X_{SR} < K_E/255 * 64$
<b>Moda</b>	$m_0 \geq K_E$	$(m_0 > 127 - K_E/255)$ && $(m_0 < 127 + K_E/255)$	$m_0 < K_E$
<b>Odch. standardowe</b>	$\sigma < K_E/255 * 24$	$\sigma < K_E/255 * 16$	$\sigma < K_E/255 * 64$
<b>Odch. przeciętne</b>	$d < K_E/255 * 40$	$d < K_E/255 * 64$	$d/2048 * 128 < K_E/2$





Rysunek 4. Przykład zastosowania algorytmu: a) bez zastosowania progów, b) z progiem

Wiadomość była ukrywana w obrazie o rozdzielczości 1024x768 pikseli (łącznie 12288 bloków 8x8 pikseli). Na pierwszym obrazie pojawiły się widoczne zniekształcenia w miejscach o jednolitych obszarach (niebo). Na drugim obrazie, gdy włączony był próg tych zniekształceń nie ma. Oczywiście jest, że w drugim przypadku zmalała ilość ukrywanych bitów wiadomości (mniejsza ilość podbloków jest klasyfikowana). Dla obydwu obrazów uzyskano następujące współczynniki (MSE – błąd średniokwadratowy, PSNR – szczytowy stosunek sygnału do szumu):

- a) obraz uzyskany bez stosowania progów:  
MSE=4.089  
PSNR=49.48
- b) obraz uzyskany ze stosowaniem progów:  
MSE=0.3526  
PSNR=73.99.

Także współczynniki uzyskiwane pokazują, że znacznie mniejsze są różnice między obrazem z ukrytą wiadomością, gdy stosowano próg doboru bloków niż, gdy go nie stosowano.

#### 4. Wnioski

Podsumowując, zastosowanie kluczy w podwójnej roli, tj. selekcji bloków obrazu oraz w funkcji ukrywającej może znacznie zwiększyć bezpieczeństwo ukrytej wiadomości przed jej ujawnieniem i odczytaniem, dzięki zastosowaniu klucza połączonego z badaniem lokalnych własności informacji nośnej. Przedstawiony schemat formowania klucza zmniejsza szanse wizualnego wykrycia faktu zaszywania wiadomości. Ponadto zastosowanie funkcji skrótu nie pozwala na jakiegokolwiek wnioskowanie na temat klucza na żadnym etapie ukrywania wiadomości (zapewnia to część

tajna klucza i bezpieczna funkcja skrótu, przy jednoczesnym założeniu zastosowania maksymalnie bezpiecznej metody steganograficznej). Dodatkowo klucz można w łatwy sposób dostosować do innych metod steganograficznych.

## Bibliografia

- [1] Menezes A. J., P. C. Van Oorschot, Vanstone S. A. *Kryptografia Stosowana*. WNT, Warszawa 2005
- [2] Nozdrzykowski Ł. *Wbudowywanie wiadomości metodami steganograficznymi w środowisku graficznym na podstawie wykorzystania specjalnych parametrów obrazu. Praca magisterska*. Politechnika Szczecińska, Wydział Informatyki, Szczecin 2006
- [3] Ross Anderson. *Stretching the Limits of Steganography* [online]. [www.cl.cam.ac.uk/~rja14/Papers/stegan.pdf](http://www.cl.cam.ac.uk/~rja14/Papers/stegan.pdf) [dostęp: 2007]
- [4] Abdelkader H., Ouda and Mahmoud R., El-Sakka, Senior Member, IEEE. *A Step Towards Practical Steganography Systems* [online]. [www.csd.uwo.ca/~kader/Publications/Ouda\\_ICIAR2005.pdf](http://www.csd.uwo.ca/~kader/Publications/Ouda_ICIAR2005.pdf) [dostęp: 2007]
- [5] Areepongsa S., Syed Y.F., Kaewkammed N., Rao K. *Steganography For Low Bit-Rate Based Image Coder* [online]. [www-ee.uta.edu/dip/paper/icip\\_2000.pdf](http://www-ee.uta.edu/dip/paper/icip_2000.pdf) [dostęp: 2007]
- [6] Sharp T. *An Implementation of Key-Based Digital Signal Steganography* [online]. <http://citeseer.ist.psu.edu/742986.html> [dostęp: 2007]
- [7] Marvel L. M., Boncelet C. G., Jr., Retter C. T. *Spread Spectrum Image Steganography*. IEEE Transactions on Image Processing, Vol. 8, No. 8, 1999
- [8] Levický D., Šurin S. *Codebook Code Division Multiple Access Image Steganography*. Radioengineering, Vol. 14, No. 1, 2005 [online]. [www.urel.feec.vutbr.cz/RADIOENG/2005/05\01\41\45.pdf](http://www.urel.feec.vutbr.cz/RADIOENG/2005/05\01\41\45.pdf) [dostęp: 2007]
- [9] Wong P. H. W., Au O. C., Wong J. W. C. *A Data Hiding Technique in JPEG Compressed Domain* [online]. <http://citeseer.ist.psu.edu/609998.html> [dostęp: 2007]
- [10] Korostil J., Nozdrzykowski Ł. *Sposób budowania klucza steganograficznego w oparciu o progowe metody wyboru bloków obrazu*. Metody Informatyki Stosowanej. Roczniki Informatyki Stosowanej Wydziału Informatyki PS nr 10, Szczecin 2006

# Porównanie własności heurystyk do wyznaczania reduktów minimalnych

Marcin Korzeń

Politechnika Szczecińska, Wydział Informatyki

## Abstract:

*The minimal reduct problem is NP-hard, therefore it can be solved exactly only for relatively small datasets. The paper presents comparison of different heuristic approaches to attribute selection for building rough set reducts, based on either discernibility matrix or information theoretical measures like entropy and Gini index. It has been shown theoretically and experimentally that entropy and Gini index work better if the reduct is later used for prediction of previously unseen cases, and the criterion based on the discernibility matrix tends to work better for learning functional relationships where generalisation is not an issue.*

## Słowa kluczowe:

*minimal reduct, entropy, Gini index, information gain*

## 1. Wprowadzenie

System informacyjny [10] jest to para  $(U, A)$ , gdzie  $U = \{u_1, \dots, u_n\}$  jest zbiorem obiektów (rekordów, przykładów), a  $A = \{a_1, \dots, a_m\}$  jest zbiorem atrybutów. Atrybut  $a \in A$  jest funkcją  $a : U \rightarrow D_a$  przyporządkowującą każdemu obiektowi wartość z dziedziny atrybutu  $D_a$ . Bez straty ogólności możemy założyć, że dziedzina jest zbiorem indeksów  $D_a = \{1, \dots, k\}$  dla każdego  $a \in A$ . System decyzyjny jest to system informacyjny  $(U, A \cup \{d\})$  z wyróżnionym atrybutem decyzyjnym  $d$ . W dalszej części zakłada się, że dziedzina atrybutu  $d$  jest równa  $D_d = \{1, \dots, l\}$ .

Reduktem warunkowym nazywamy minimalny (w sensie inkluzji) zbiór atrybutów warunkowych, który pozwala wyznaczyć wartość atrybutu decyzyjnego  $d$  z taką samą dokładnością, jak pełny zbiór atrybutów warunkowych, formalną definicję można znaleźć przykładowo w [10]. Z systemem decyzyjnym  $(U, A \cup \{d\})$  można związać następującą macierz nierozróżnialności

$$M_{ij}^d = \{a \in A : a(u_i) \neq a(u_j) \text{ i } d(u_i) \neq d(u_j)\}.$$

Macierz rozróżnialności systemu decyzyjnego przechowuje w komórce o indeksach  $(i, j)$  atrybuty, które rozróżniają obiekty o numerach  $i$  oraz  $j$ , o ile decyzje dla tych obiektów są różne, w przeciwnym razie dana komórka macierzy jest pusta.

Analogicznie rozważa się macierze rozróżnialności systemów informacyjnych (nie bierze się pod uwagę decyzji), jednak w analizie danych większe znaczenie

```

1: function  $P = \text{MINIMALREDUCTWITHDISCERNIBILITYMATRIX}((U, A))$ 
2:    $P := \emptyset, i := 0$ 
3:   Wyznacz macierz/listę rozróżnialności  $M^d$ 
4:   while  $M \neq \emptyset$  do
5:     Oblicz częstości atrybutów w macierzy  $M^d$ ,
6:     Wybierz najczęstszy atrybut  $a_{*i}$  i dodaj do  $P$ 
7:     Usuń wszystkie komórki zawierające  $a_{*i}$ 
8:      $i := i + 1$ 
9:   end while
10:  Usuń kolejno zbędne elementy ze zbioru  $P$ 
11: end function

```

Rysunek 1. Algorytm heurystyczny poszukiwania reduktów z wykorzystaniem macierzy rozróżnialności (por. [1], [6] )

mają systemy decyzyjne<sup>4</sup>. Problem wyznaczania reduktów bezwzględnych (związanych z systemami informacyjnymi) można sprowadzić do problemu wyznaczania reduktów względnych, dodając do systemu informacyjnego atrybut decyzyjny numerujący klasy obiektów nierozróżnialnych.

Macierz rozróżnialności ma silny związek z reduktami systemu informacyjnego. W terminach macierzy rozróżnialności *redukt* jest minimalnym (w sensie inkluzji) zbiorem atrybutów, który ma niepusty przekrój z każdą komórką macierzy nierozróżnialności. Przez *redukt minimalny* rozumie się redukt o minimalnej długości, takich reduktów w systemie informacyjnym może być kilka. Ponadto atrybut  $a$  jest *niesuwalnym* w systemie decyzyjnym, jeżeli w macierzy rozróżnialności istnieje komórka dokładnie równa  $\{a\}$ . Zbiór atrybutów niesuwalnych tworzy *rdzeń*. Rozmiar macierzy jest rzędu  $O(n^2m)$ ,  $n$  jest liczbą obiektów, a  $m$  atrybutów. Jest to dość dużo nawet dla niezbyt dużych zbiorów danych.

Wiele algorytmów wyznaczania reduktów (zarówno wszystkich jak i minimalnych), (por. [1], [12]) bazuje na konstrukcji macierzy rozróżnialności. Cała informacja o systemie decyzyjnym jest zawarta w macierzy rozróżnialności. Informacje te wykorzystuje się następująco: te atrybuty, które sprzyjają rozróżnianiu decyzji występują częściej w macierzy rozróżnialności oraz atrybut objaśnia taki fragment systemu informacyjnego jak często występuje on w macierzy rozróżnialności. Jeżeli wybierzemy taki częsty atrybut do potencjalnego reduktu to możemy usunąć z macierzy komórki zawierające taki atrybut – ta część macierzy może być odtworzona przy użyciu tego atrybutu.

Standardowe podejście do wyznaczania reduktu o minimalnej długości przedstawia algorytm na rysunku 1. Algorytm ten w zachłanny sposób wybiera kolejno atrybuty, które rozróżniają największą liczbą par obiektów pod warunkiem, że pewną liczbę atrybutów już wybrano.

W dalszej części używane będą następujące oznaczenia: dla częstości atrybutów  $n_{.i} = |\{u \in U : a(u) = i\}|$ ,  $n_{.j} = |\{u \in U : d(u) = j\}|$ ,  $n_{ij} = |\{u \in U : a(u) =$

<sup>4</sup> Z tego względu w pracy będą rozważane systemy decyzyjne i redukty warunkowe

$i \wedge d(u = j)$ ],  $a = i$  oznacza że atrybut  $a$  przyjmuje wartość  $i$ ,  $a = i \wedge d = j$  oznacza zdarzenia łączne  $a = i$  i  $d = j$ , ponadto  $p_{i.} = \frac{n_{i.}}{n}$ ,  $p_{.j} = \frac{n_{.j}}{n}$ ,  $p_{ij} = \frac{n_{ij}}{n}$  są oszacowaniami prawdopodobieństw  $a = i$ ,  $d = j$  i  $a = i \wedge d = j$ . Dla potrzeb analizy zakłada się, że każdy atrybut warunkowy przyjmuje dokładnie  $k$  wartości, natomiast atrybut decyzyjny  $l$ .

## 2. Teorio-informacyjne miary oceny atrybutów

W pracach [8], [5], [9] pokazano, że najważniejsze informacje o macierzy różnorodności można odtworzyć nie konstruując tej macierzy, a jedynie bazując na częstościach atrybutów. Takie podejście daje znaczny zysk obliczeniowy, gdyż rozkład pojedynczego atrybutu można znaleźć w czasie proporcjonalnym do  $n$ .

Zasadnicze pytanie przy wyborze atrybutu jest następujące: mając atrybut o danych częstościach należy obliczyć liczbę par obiektów, które ten atrybut różni – tj. liczbę komórek macierzy nierozróżnialności, w których jest on obecny. Główna idea zliczania jest następująca. Rozważmy atrybut  $a$ , który przyjmuje dwie wartości '0' z częstością  $n_1$  oraz '1' z częstością  $n_2$ , zatem liczba par obiektów, które mają różne wartości z jednej strony  $S(a) = 2n_1n_2$  (wybieramy parę (obiekt z klasy '0', obiekt z klasy '1') oraz odwrotnie), natomiast z drugiej strony  $S(a) = n^2 - n_1^2 - n_2^2$ , gdzie  $n = n_1 + n_2$  jest liczbą obiektów (od liczby wszystkich par obiektów odejmujemy pary obiektów z klasy '0' i pary z klasy '1'). Powyższe rozumowanie w łatwy sposób przenosi się na atrybuty o skończonej liczbie wartości, również w przypadku zliczania par przy uwzględnieniu atrybutu decyzyjnego.

### Przykład

W tabeli 1 przedstawiono przykład systemu decyzyjnego  $(U, A \cup \{d\})$ ,  $U = \{1, 2, 3, 4, 5, 6, 7\}$ ,  $A = \{a_1, \dots, a_5\}$  oraz jego macierz  $[M_{ij}^d]$  nierozróżnialności (tak naprawdę przedstawiono jedynie połowę – górno trójkątny fragment w postaci listy). Można sprawdzić używając wzorów zebranych w tab. 2, że licznosci w macierzy rozróżnialności można wyliczyć bazując na licznosciach łącznych atrybutów  $(a_i, d)$ , które są przedstawione w tab. 1 poniżej tabeli informacyjnej. Przykładowo:

$$\begin{aligned} I_S(a_1, d) &= S(d) - S(d|a_1) = ((7^2 - 1^2 - 2^2 - 1^2 - 1^2 - 2^2) - \\ &\quad (1^2 - 1^2) - (2^2 - 1^2 - 1^2) - (4^2 - 1^2 - 2^2 - 1^2)) \\ &= (38 - 0 - 2 - 10) = 26 = 2 \cdot 13, \end{aligned}$$

odpowiada to licznosci atrybutu  $a_1$  w macierzy rozróżnialności. Gdy dodamy do potencjalnego reduktu najczęściej występujący atrybut  $a_3$  oraz następnie usuniemy wszystkie pola macierzy, które zawierają ten atrybut, dostaniemy zredukowaną macierz nierozróżnialności względem decyzji, pod warunkiem, że wybrany został atrybut  $a_3$ . Zredukowaną macierz przedstawiono również w tabeli 1. Proces wyboru atrybutów kontynuujemy do momentu, gdy w wyniku kolejnych redukcji dostaniemy macierz pustą.  $\square$

Tabela 1. Przykładowy system decyzyjny ( $U, A \cup d$ ) (poniżej podano częstości atrybutów) oraz jego macierz nierozróżnialności.

System decyzyjny							Macierz rozróżnialności						
$u_i$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$d$	$i \ j$	$M(i, j)$	$i \ j$	$M(i, j)$			
1	1	1	1	2	3	3	1 2	1 4	2 7	1 2 3 4			
2	2	1	1	1	3	4	1 3	1 3 4 5	3 5	1 4 5			
3	2	1	2	1	1	5	1 4	1 4 5	3 6	1 2 3 4 5			
4	3	1	1	3	2	5	1 5	1 3 5	3 7	1 2 3 4 5			
5	3	1	2	2	2	1	1 6	1 2 3 4 5	4 5	3 4			
6	3	3	3	3	2	2	1 7	1 2 3 4	4 6	2 3			
7	3	3	3	3	3	2	2 3	3 5	4 7	2 3 5			
							2 4	1 4 5	5 6	2 3 4			
							2 5	1 3 4 5	5 7	2 3 4 5			
							2 6	1 2 3 4 5					
	$a_1 d$	$a_2 d$	$a_3 d$	$a_4 d$	$a_5 d$	$d$	Macierz po redukcji						
$d$	$n_1$	$n_2$	$n_3$	$n_1$	$n_2$	$n_3$	$n_1$	$n_2$	$n_3$	$n_1$	$n_2$	$n_3$	$n_d$
1	0	0	1	1	0	0	0	1	0	0	1	0	1
2	0	0	2	0	0	2	0	0	2	0	1	1	2
3	1	0	0	1	0	0	0	1	0	0	0	1	1
4	0	1	0	1	0	0	0	0	0	1	1	1	1
5	0	1	1	2	0	0	1	1	0	1	1	0	2
$S_{a_i}^d$	13	10	15	15	13	19							
$i \ j$	$M_{ij}$												
1 2	1 4												
1 4	1 4 5												
2 4	1 4 5												
6 7	5												

Wybór atrybutu na podstawie częstości występowania w macierzy nierozróżnialności w naturalny sposób prowadzi do pewnej miary  $S$  atrakcyjności atrybutu, zbliżonej wzorem do indeksu Gini'ego (por. tab. 2). Ciekawe jest pytanie: czy zmieniając miarę (gain)  $S$  można poprawić własności atrybutów tworzących redukt.

Problem znajdowania minimalnego reduktu jest NP-zupełny [6]. Natomiast przedstawiony algorytm zachłanny (alg. 1), niezależnie od wyboru miary przyrostu informacji działa w czasie wielomianowym. Zatem trudno oczekiwać, aby znalazła się miara – będąca funkcją częstości atrybutów, która zawsze wskazywałaby takie atrybuty, aby uzyskany redukt był minimalny w sensie długości (oznaczałoby to w szczególności, że  $P=NP$ ). Warto zaznaczyć, że dla poszukiwania reduktów minimalnych stosuje się bardziej zaawansowane podejścia, jak np. algorytmy genetyczne zaimplementowane w systemie RSES [2]. Eksperymenty pokazują, że takie rozwiązania może poprawić jakość rozwiązania, jednak dla dużych zbiorów danych takie podejścia są często wolne (por. [8]).

W tabeli 2<sup>5</sup> przedstawiono zestawienie postawowych miar atrakcyjności atrybutów. Miary  $G_2$  i  $S$  różnią się jedynie tym, że pierwsza jest napisana w terminach częstości, a druga w terminach liczości. Miary te odpowiadają wybieraniu atrybutów na podstawie macierzy nierozróżnialności.

Różnica pomiędzy miarą  $G_2$ , a indeksem Gini'ego  $G$  mimo podobieństwa wzorów jest istotna. Na entropii  $H$  opierają się liczne algorytmy konstrukcji drzew klasyfikacyjnych, jak ID3 czy C4.5 [11]. Miara  $G$  jest wykorzystana przez algo-

<sup>5</sup> Zanieczyszczenie - ang. *impurity* - terminologia stosowana w drzewach klasyfikacyjnych

Tabela 2. Podstawowe miary informacyjności atrybutów:  $H$  – entropia Shannona,  $G = G_{CART}$  – index Gini’ego,  $G_2$  – zmodyfikowany index Gini’ego opowiadający macierzy rozróżn.,  $S$  – liczności jak w macierzy rozróżnialności (liczba par obiektów, które atrybut rozróżnia),  $M$  prawdopodobieństwo błędnej klasyfikacji (ang. *misclassification*).

miara •	ogólnie •( $d$ )	warunkowa •( $d a = i$ )	zanieczyszczenie •( $d a$ )	przyrost informacji( $a,d$ )
$H$	$\sum_j p_j \log(p_j)$	$\sum_{j=1}^l \frac{p_{ij}}{p_i} \log\left(\frac{p_{ij}}{p_i}\right)$	$\sum_{i=1}^k p_i H(d a = i)$	$I_H = H(d) - H(d a)$
$G$	$1 - \sum_{j=1}^l p_j^2$	$1 - \sum_{j=1}^l \left(\frac{p_{ij}}{p_i}\right)^2$	$\sum_{i=1}^k p_i G(d a = i)$	$I_G = G(d) - G(d a)$
$G_2$	$1 - \sum_{j=1}^l p_j^2$	$1 - \sum_{j=1}^l \left(\frac{p_{ij}}{p_i}\right)^2$	$\sum_{i=1}^k p_i^2 G(d a = i)$	$I_{G_2} = G(d) - G_2(d a)$
$S$	$n^2 - \sum_{j=1}^l n_j^2$	$n_i^2 - \sum_{j=1}^l n_{ij}^2$	$\sum_{i=1}^k S(d a = i)$	$I_S = S(d) - S(d a)$
$M$	$1 - \max_{j=1}^l p_j^2$	$1 - \max_{j=1}^l \left(\frac{p_{ij}}{p_i}\right)^2$	$\sum_{i=1}^k p_i M(d a = i)$	$I_M = M(d) - M(d a)$

rytm Breimana CART [3]. Powszechnie uważa się, że miara  $M$  ma znacznie słabsze własności do wyboru atrybutów niż entropia czy indeks Gini’ego (por. [7, str. 130, 131]), z tego względu te właśnie miary znajdują zastosowanie przy konstrukcji drzew decyzyjnych. W dalszej części analizowane będzie zachowanie zachłannego sposobu wyboru atrybutów (analogicznie jak w alg. 1) przy różnym wyborze miar przyrostu informacji (ang. *gain*), nie tylko  $I_S$ .

Miary  $G_2$  i  $I_{G_2}$  pochodzące tak naprawdę od macierzy rozróżnialności mają szereg ciekawych własności analogicznych do entropii  $H$  i informacji wzajemnej Shannona  $I_H$ . przykładowo prawdziwe są zależności:

$$I_{G_2}(a, d) = G(a) + G(d) - G(a, d), \quad (1)$$

$$G(a, d) = G(a) + G_2(d|a) = G(d) + G_2(a|d), \quad (2)$$

$$G(a, d) = G(d, a). \quad (3)$$

Te zależności nie są prawdziwe dla przyrostu informacji Gini’ego  $I_G$  używanego przez algorytm CART.

Podstawową własnością, która odróżnia miary  $I_H$  i  $I_G$  od  $I_{G_2}$  jest zachowanie w przypadku atrybutów niezależnych. Zarówno  $I_H$  jak i  $I_G$  nie wybierze atrybutu, który jest statystycznie niezależny od decyzji ponieważ w obu przypadkach mamy: jeżeli  $a$  i  $d$  są niezależne, to  $I(a, d) = 0$ ,  $H(d|a) = H(a)$  oraz  $G(d|a) = G(d)$ . Natomiast w przypadku miary  $I_{G_2}$  mamy: jeżeli  $a$  i  $d$  są niezależne mamy<sup>6</sup>:

$$I_{G_2}(a, d) = G(a)G(d) > 0 \quad (4)$$

$$G(d|a) = G(d)(1 - G(a)) \neq G(d). \quad (5)$$

Zależności te mają kluczowe znaczenie przy wyborze atrybutów zwłaszcza w problemach uczenia maszynowego, gdzie rzadziej zależy nam na modelowaniu nieznannej zależności funkcyjnej pomiędzy atrybutami warunkowymi zapisanej w konkretnej tabeli decyzyjnej, a raczej na przewidywaniu/klasyfikacji obiektów

<sup>6</sup> Pierwsze wyrażenie jest niezerowe, pomijając trywialne przypadki atrybutów stałych

pochodzących z sytemu rzeczywistego, dla którego tabela informacyjna jest jedynie próbą. Przewidywanie takie powinno być wsparte od strony statystycznej – czyli przykładowo podobnie zachowywać się na niezależnej tabeli walidującej oraz posiadać własności uogólniające. Różnice w zachowaniu się miary  $I_{G_2}$  z jednej strony a  $I_H$  i  $I_G$  z drugiej uwypukla następujący przykład.

### Przykład

Rozważmy następującą rodzinę systemów informacyjnych (na podstawie [5]). Każdy system zawiera  $2^n$  obiektów. Atrybuty warunkowe składają się z dwóch grup  $A$  i  $B$ . Grupa  $A$  zawiera  $n$  atrybutów przyjmujące wszystkie  $2^n$  wartości kodu binarnego długości  $n$ . decyzja jest określona jako funkcja tylko atrybutów z grupy  $A$ :  $d = \sum_{i=1}^n a_i \bmod 2$ . Natomiast grupa  $B$  zawiera  $2^{n-1}$  atrybutów. Każdy z nich jest singletonem, tzn. przyjmuje w jednej tylko komórce wartość 1 – jeżeli decyzja jest równa 1. Przykład dla  $n = 3$  jest przedstawiony w tabeli 3. Nietrudno sprawdzić, że w systemie tym atrybuty  $a_i$  są statystycznie niezależne od decyzji, natomiast atrybuty  $b_j$  są zależne z decyzją, jednak bardzo słabo. Jak można się przekonać z tabeli 3 (część dolna) miary  $I_H$  i  $I_G$  nie zauważają atrybutów  $a_i$ , a nawet układu atrybutów:  $a_{i_1}a_{i_2}$ . Natomiast z tego względu że zależność pomiędzy  $b_j$  a  $d$  jest słaba miara  $I_{G_2}$  w pierwszej kolejności wybiera właśnie atrybut z  $a_i$  mimo, że jest on niezależny od decyzji.  $\square$

Tabela 3. Przykład opisanego systemu informacyjnego dla  $n=3$ :  $2^3$  obiektów,  $2^{3-1}$  atrybutów typu  $B$ , 3 atrybuty typu  $A$ . Poniżej: zmiana wartości  $I_G$ ,  $I_{G_2}$  i  $I_H$  wraz z dodaniem kolejnych atrybutów.

no.	$a_1$	$a_2$	$a_3$	$b_1$	$b_2$	$b_3$	$b_4$	$d = \sum a_i \pmod{2}$
1	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	1
3	0	1	0	0	1	0	0	1
4	0	1	1	0	0	0	0	0
5	1	0	0	0	0	1	0	1
6	1	0	1	0	0	0	0	0
7	1	1	0	0	0	0	0	0
8	1	1	1	0	0	0	1	1

	$(d a_1)$	$(d a_1a_2)$	$(d a_1a_2a_3)$	$(d b_1)$	$(d b_1b_2)$	$(d b_1b_2b_3)$	$(d b_1b_2b_3b_4)$
$I_G$	0.250	0.375	0.500	0.125	0.250	0.375	0.500
$I_H$	0.000	0.000	1.000	0.138	0.311	0.549	1.000
$I_{G_{CART}}$	0.000	0.000	0.500	0.071	0.167	0.300	0.500

W tym szczególnym przykładzie dość złośliwie dobrano zależność funkcyjną pomiędzy atrybutami warunkowymi a decyzją. Jednak to pokazuje, że redukt znalezionej z użyciem entropii będzie zawierał  $2^{n-1}$  atrybutów, natomiast redukt



Tabela 4. Porównanie jakości reduktów uzyskanych przy pomocy heurystyki  $S/G2$  z algorytmem genetycznym RSES [2]. Podano czas działania i w nawiasach długości uzyskanych reduktów.

	TIC 43	TIC 85	SATELLITE	SHUTTLE
rozmiar danych	(5822, 43)	(5822, 85)	(4435, 36)	(43500,9)
heurystyka $G2$	36 s (7)	274 s (23)	22 s (7)	36 s (4)
RSES	97 s (7)	466 s (22)	57 s (6)	272 s (4)

znaleziony na podstawie ilości par rozróżnień będzie zawierał znacznie mniej atrybutów bo tylko  $n$ . Z tego względu w ogólności nie można oczekiwać aproksymacyjnych<sup>7</sup> własności przy poszukiwaniu reduktów minimalnych z użyciem miar  $I_H$  czy  $I_G$ . Natomiast „aproksymacyjność” rozwiązania znalezionego z użyciem miary  $I_{G2}$  jest w kontekście tego przykładu otwarta, niemniej wyjaśnia on dlaczego wybór atrybutów na podstawie miary  $I_{G2}$  może dawać krótsze atrybuty.

### 3. Eksperymenty

Przy poszukiwaniu reduktów zasadniczo interesują nas dwa zagadnienia: 1) znalezienie reduktu minimalnego, 2) znalezienie minimalnego podzbioru atrybutów, który posiadałby własności uogólniające do klasyfikacji. W praktyce rzadko kiedy te dwa zagadnienia się pokrywają.

Pierwsze z tych zagadnień jest NP-zupełnym problemem optymalizacji dyskretnej zbliżonym treścią do zagadnienia pokrycia zbiorami (ang. *set covering*), zwłaszcza jeżeli weźmiemy pod uwagę kontekst macierzy nierozróżnialności. Natomiast drugie jest zagadnieniem uczenia maszynowego w pewnym sensie podobne do konstrukcji drzew decyzyjnych, zarówno jeśli chodzi o wybór atrybutów, jak i sposób przycinania. Dalej w części eksperymentalnej analizowane będą oba zagadnienia.

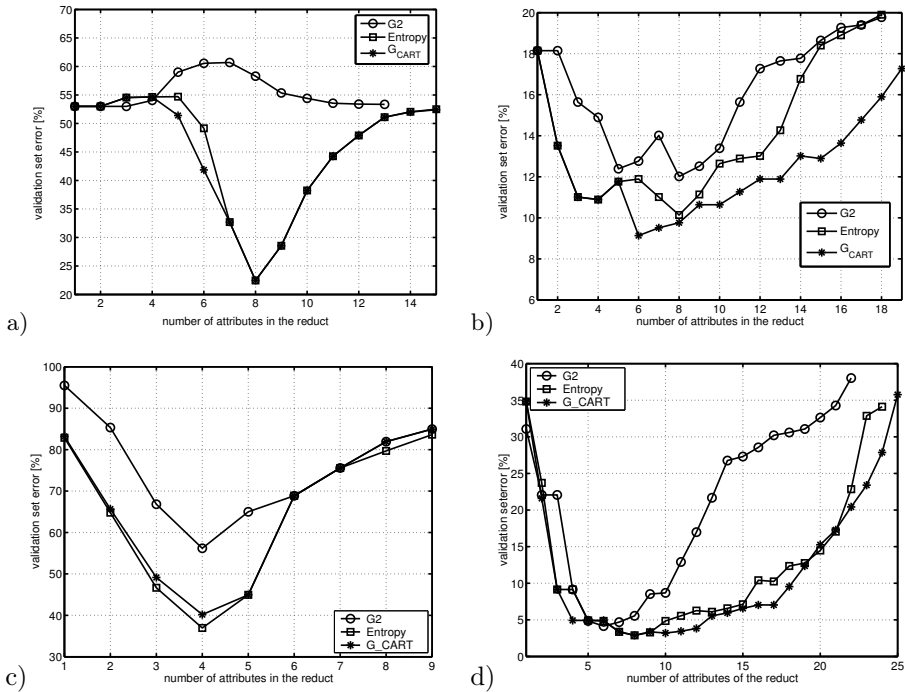
#### Poszukiwanie reduktów minimalnych

W tabeli 4 przedstawiono wyniki porównania działania zachłannego sposobu wyboru atrybutów opierając się na zliczaniu par (miara  $I_S$ ) z algorytmem genetycznym zaimplementowanym w systemie RSES [2]. Jak można się przekonać, czas działania jest zawsze znacznie krótszy, natomiast w dwóch przypadkach algorytm genetyczny znalazł rozwiązanie krótsze o jeden atrybut. Optymalność globalna rozwiązań w każdym przypadku pozostaje otwarta.

#### Własności uogólniające

W drugim eksperymencie analizowane są uogólniające własności znalezionych zbiorów atrybutów (reduktów). Każdy ze zbiorów danych został podzielony na

<sup>7</sup> Własność aproksymacji rozwiązania jest tu rozumiana jako gwarancja, że uzyskany redukt  $r$  nie różni się (nie jest dłuższy) od minimalnego  $r^*$  o więcej niż  $\alpha$  krotnie, ( $\alpha > 1$ ) tj.  $|r| \leq \alpha|r^*$



Rysunek 2. Porównanie jakości selekcji atrybutów na zbiorze walidującym dla różnych zbiorów danych: sztuczne dane (a), klasyfikacja spamu (1600 maili, 536 słów wziętych pod uwagę) (b), zbiór *letter* (c), przewidywanie końcówki szachowej zbior *chessKRvsKP* (d)

dwie części, uczącą  $U$  i walidującą  $V$ . Za każdym razem wybieranie atrybutów do reduktu odbywa się na zbiorze uczącym. Natomiast na zbiorze walidującym określany jest procent błędnych klasyfikacji jeżeli weźmiemy pod uwagę  $i$  pierwszych atrybutów z reduktu. Wyniki eksperymentu przedstawione są na rysunku 2.

Pierwszy rysunek (2a) przedstawia wyniki dla sztucznego zbioru danych. Zawiera on 4000 obiektów i 50 wygenerowanych niezależnie atrybutów, zbiór wartości każdego z atrybutów był 2 lub 3 elementowy. Atrybut decyzyjny został wybrany jako:  $d = \sum_{i=11}^{18} a_i^2 \bmod 5$ . Ponadto dla 25% obiektów zmieniono wartość atrybutu decyzyjnego. Jak można zauważyć zarówno miary  $I_H$  jak  $I_{G_{CART}}$  prawidłowo odgadły 8 atrybutów, w których była zawarta pewna informacja o decyzji, natomiast miara  $I_{G2}$  wybrała co prawda krótszy o dwa atrybuty redukt, który jednak nie posiada żadnych własności uogólniających – dobrze dopasował się on jedynie do zbioru uczącego.

Kolejne przykłady są rzeczywistymi zbiorami danych: 2b to filtracja spamu – wzięto pod uwagę 536 najczęściej występujących słów/ atrybutów – jak można zaobserwować stosunkowo dobry klasyfikator można uzyskać biorąc pod uwagę 6-8 słów wybranych w pierwszej kolejności, 2c to klasyczne benchmarki z repozytorium UCI/KDD [4]. Jak można zaobserwować kryterium  $G2$  zawsze odnajdowało

najkrótsze redukty, natomiast zdolność do uogólniania jest zwykle gorsza, znacznie lepiej pod tym względem działają miary  $G_{CART}$  i  $H$ .

#### 4. Podsumowanie

W pracy analizowano zachowanie różnych heurystyk do poszukiwania reduktów minimalnych. Z przeprowadzonych eksperymentów wynika, że heurystyka bazująca na macierzy rozróżnialności dobrze nada się do znajdowania krótkich reduktów, dla ustalonego systemu decyzyjnego, natomiast jeżeli chodzi o własności uogólniające to znacznie lepiej zachowującą się miary  $H$  i  $G_{CART}$ . Słabsze własności uogólniające miary  $I_{G2}$  spowodowane są najprawdopodobniej tym, że miara ta może dodawać do reduktu atrybuty statystycznie niezależne od decyzji. Należy zaznaczyć że w problemach analizy danych większe znaczenie ma znalezienie krótkiego reduktu przybliżonego, który posiada zdolność generalizacji na nieznanymi danych. Taki redukt przybliżony może być wykorzystany przy wstępnej selekcji atrybutów jak i do klasyfikacji.

#### Bibliografia

- [1] Bazan J. *Metody wnioskowań aproksymacyjnych dla syntezy algorytmów decyzyjnych*. Praca doktorska, Uniwersytet Warszawski 1999
- [2] Bazan J., Szczuka M., Wróblewski J. *A new version of rough set exploration system*. Third International Conference on Rough Sets and Current Trends in Computing RSCTC. Lecture Notes in Artificial Intelligence, Vol. 2475, s. 397-404, Springer-Verlag 2002
- [3] Breiman L., Olshen R. A., Friedman J. H., Stone C. J. *Classification and Regression Trees*. CRC Press 1984
- [4] Hettich S., Bay S. D. *The UCI KDD Archive* <http://kdd.ics.uci.edu>. Irvine, CA: University of California, Department of Information and Computer Science, 1999
- [5] Jaroszewicz S., Korzeń M. *Comparison of Information Theoretical Measures for Reduct Finding*. W: Rutkowski L., Tadeusiewicz R., Zadeh L. A., Zurada J. (red.) *ICAISC*. Lecture Notes in Computer Science, Vol. 4029, s. 518-527, Springer 2006
- [6] Komorowski J., Polkowski L., Skowron A. *Rough sets: a tutorial*, W: Pal S. K., Skowron A. (red.), *Rough-Fuzzy Hybridization: A New Method for Decision Making*. Springer-Verlag 1998
- [7] Koronacki J., Ćwik J. *Statystyczne systemy uczące*. WNT 2005
- [8] Korzeń M., Jaroszewicz S. *Finding Reducts Without Building the Discernibility Matrix*. ISDA, s. 450-455. IEEE Computer Society 2005
- [9] Nguyen S. H., Nguyen H. S. *Some efficient algorithms for rough set methods*. Proceedings of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96, s. 1451-1456. Granada, Spain, 1996
- [10] Pawlak Z. *Rough sets: Theoretical aspects of reasoning about data*. Dordrecht: Kluwer 1991
- [11] Quinlan R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann 1993
- [12] Zhang J., Wang J., Li D., He H., Sun J. *A New Heuristic Reduct Algorithm Base on Rough Sets Theory*. Lecture Notes in Computer Sciences, Vol. 2762, s. 247-253, Springer 2003



# Algorytmy wielopoziomowej dekompozycji i rekonstrukcji falkowopodobnej danych 2D

Dorota Majorkowska-Mech, Alexandr Țariov

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*In this article the vector-matrix computational procedures describing original algorithms of multilevel decomposition and reconstruction of two-dimensional data arrays are presented. The difference is in a data filtration approach. In described method data filtration is realized in discrete orthogonal bases domain instead of the traditional one using in wavelet method. The proposed procedures of multilevel decomposition and reconstruction, if large arsenal of fast algorithms of transforms computation in orthogonal bases is used, allows reducing number of arithmetic operations comparing with traditional wavelet method approach and accuracy of data representation is kept.*

## Słowa kluczowe:

*Two-dimensional Discrete Wavelet Transform 2D-DWT, discrete orthogonal bases, wavelet-like decomposition for images*

## 1. Wprowadzenie

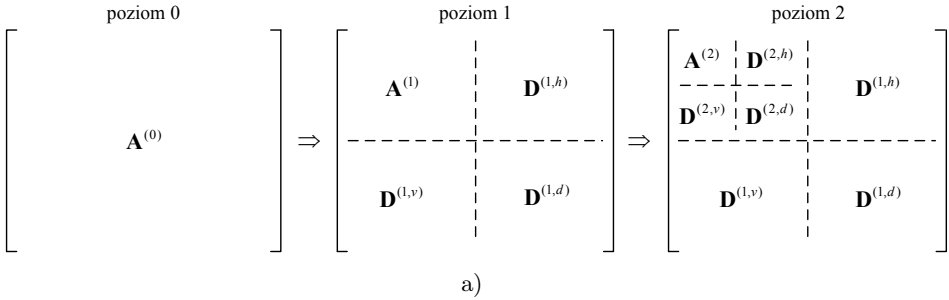
Dwuwymiarowa dyskretna transformata falkowa (ang. *Two-dimensional Discrete Wavelet Transform* – 2D-DWT) jest najbardziej rozpowszechnioną obecnie metodą cyfrowego przetwarzania obrazów. Wystarczy nadmienić, iż falki zostały wykorzystane w archiwatorze DjVu, standardach kompresji danych multimedialnych MPEG-4, JPEG2000 [1].

Istotą dekompozycji wielorozdzielczej jest podział sygnału na komponenty o mniejszej rozdzielczości za pomocą pary filtrów dolno- i górnoprzepustowego, i następnie hierarchiczna dekompozycja składowej niskoczęstotliwościowej w analogiczny sposób.

Pierwszy krok dekompozycji falkowej obrazu cyfrowego reprezentowanego przez macierz wartości amplitud o rozmiarze  $N \times M$ , polega na wyznaczeniu  $N$  jednowymiarowych transformat falkowych dla każdego wiersza macierzy reprezentującej ten obraz, a następnie wyznaczeniu  $M$  jednowymiarowych transformat nad kolumnami poprzednio wyznaczonych wyników. Jako wynik otrzymujemy cztery macierze współczynników: aproksymacji –  $\mathbf{A}^{(1)}$  oraz detali –  $\mathbf{D}^{(1,h)}$ ,  $\mathbf{D}^{(1,v)}$ ,  $\mathbf{D}^{(1,d)}$ . Pierwszy indeks przy macierzach oznacza poziom reprezentacji, zaś drugi indeks,

przy macierzach współczynników detali – nazwę detalu. Macierze te reprezentują dane otrzymane w wyniku filtracji dolnoprzepustowej oraz górnoprzepustowej oraz dwukrotnej decymacji (ang. *down-sampling*). Macierz  $\mathbf{A}^{(1)}$  zawierająca współczynniki aproksymacji, powstaje jako wynik filtracji dolnoprzepustowej zarówno według wierszy jak i kolumn. Macierz  $\mathbf{D}^{(1,h)}$  zawiera współczynniki tzw. detalu horyzontalnego, powstaje jako wynik filtracji dolnoprzepustowej według wierszy i górnoprzepustowej według kolumn. Macierz  $\mathbf{D}^{(1,v)}$  zawiera współczynniki detalu wertykalnego, powstaje jako wynik filtracji górnoprzepustowej według wierszy i dolnoprzepustowej według kolumn, natomiast  $\mathbf{D}^{(1,d)}$  to współczynniki detalu diagonalnego, który jest wynikiem filtracji górnoprzepustowej zarówno według wierszy jak i kolumn.

W dwuwymiarowej transformacie falkowej obrazu w kolejnym kroku  $k$  dekomponowana jest tylko macierz współczynników aproksymacji z poprzedniego kroku, tzn.  $\mathbf{A}^{(k-1)}$ . Wynikiem dekompozycji obrazu na poziomie  $k \geq 1$  jest  $1 + 3k$  macierzy oznaczonych następująco:  $\mathbf{A}^{(k)}$  – macierz współczynników aproksymacji na poziomie  $k$ , oraz macierze współczynników detali z poziomu  $k$  i poprzednich  $\mathbf{D}^{(k,h)}$ ,  $\mathbf{D}^{(k,v)}$ ,  $\mathbf{D}^{(k,d)}$ ,  $\mathbf{D}^{(k-1,h)}$ ,  $\mathbf{D}^{(k-1,v)}$ ,  $\mathbf{D}^{(k-1,d)}$ , ...,  $\mathbf{D}^{(1,h)}$ ,  $\mathbf{D}^{(1,v)}$ ,  $\mathbf{D}^{(1,d)}$ . Sposób dekompozycji falkowej obrazu został przedstawiony na rysunku 1.



b)

Rysunek 1. a) Sposób segmentacji danych 2D w dwóch krokach dekompozycji falkowej. b) Dekompozycja falkowa obrazu „Lichtenstein” [2] za pomocą filtrów Daubechies o długości 8. W celu wizualizacji współczynniki poszczególnych składowych zdekomponowanego obrazu zostały przeskalowane tak, aby wykorzystać pełny zakres odcieni szarości.

$\mathbf{A}^{(0)}$  jest obrazem wejściowym, czyli aproksymacją na poziomie zerowym;  $\mathbf{A}^{(1)}$ ,  $\mathbf{A}^{(2)}$ , itd. to współczynniki kolejnych aproksymacji, reprezentujących obraz wejściowy z coraz mniejszą rozdzielczością.

Wielopoziomową reprezentację sygnału można uzyskać również wykorzystując klasyczne dyskretne bazy ortogonalne. Ogólne podejście do tego problemu dla sygnałów jednowymiarowych zostało zaprezentowane w pracach [3,4]. W pracy [5] przedstawiono algorytmy dekompozycji i rekonstrukcji falkowopodobnej sygnałów jednowymiarowych wykorzystując bazę funkcji kosinusowych DCT-II. Metoda tam zaprezentowana ma tę przewagę nad metodą falkową, że dekompozycja może odbywać się bezpośrednio na wybrany poziom, a nie krok po kroku jak w przypadku dekompozycji falkowej. Algorytmy te można uogólnić na przypadek dwuwymiarowy, tzn. zastosować je do obrazów. Jeśli do wyznaczania transformat w bazach ortogonalnych zastosujemy szybkie algorytmy [6,7], to liczba operacji arytmetycznych przy wyznaczaniu falkowopodobnej dekompozycji obrazu może być mniejsza niż dla dekompozycji falkowej, zwłaszcza gdy poziom dekompozycji jest rzędu 4-6, a do takiego właśnie poziomu zwykle dekomponuje się obrazy w celu ich kompresji lub odszumienia.

## 2. Synteza procedur obliczeniowych dla dwuwymiarowej transformacji falkowopodobnej

Zacniemy od procedur jednowymiarowej falkowopodobnej dekompozycji i rekonstrukcji sygnału, ponieważ dekompozycja obrazu będzie polegała na wykonaniu transformacji jednowymiarowych dla wierszy, a następnie kolumn macierzy reprezentującej obraz cyfrowy.

Niech  $\mathbf{x}_{N \times 1} = [x_0, x_1, \dots, x_{N-1}]^T$  oznacza wektor kolumnowy danych wejściowych, gdzie  $N$  jest liczbą próbek sygnału wejściowego, a symbol  $[\cdot]^T$  oznacza operację transpozycji.

Przez  $\mathbf{B}_N = [b_{mn}]$  oznaczymy kwadratową macierz ortonormalną stopnia  $N$ ; wektory bazowe będące wierszami tej macierzy są ortogonalne i unormowane.

Wprowadźmy także transponowaną macierz przekształcenia ortogonalnego  $\mathbf{B}_{\frac{N}{2^k}}^T$  stopnia  $N/2^k$ , gdzie  $k$  oznacza poziom reprezentacji sygnału. Założyć należy, że liczba próbek sygnału wejściowego  $N$  jest podzielna przez  $2^k$ .

Filtracja dolno- i górnoprzepustowa jednowymiarowego sygnału cyfrowego  $\mathbf{x}_{N \times 1}$  może być realizowana w następujący sposób:

$$\mathbf{y}_{N \times 1}^{(1)} = \left( \mathbf{B}_{\frac{N}{2}}^T \oplus \mathbf{B}_{\frac{N}{2}}^T \right) \mathbf{B}_N \mathbf{x}_{N \times 1}, \quad (1)$$

gdzie symbol  $\oplus$  oznacza sumę prostą macierzy,  $\mathbf{y}_{N \times 1}$  jest wektorem współczynników dekompozycji o następującej strukturze  $\left[ \mathbf{a}_{1 \times \frac{N}{2}}, \mathbf{d}_{1 \times \frac{N}{2}} \right]^T$ , gdzie  $\mathbf{a}_{\frac{N}{2} \times 1}$  jest wektorem współczynników aproksymacji (dolnoprzepustowej składowej sygnału), a  $\mathbf{d}_{\frac{N}{2} \times 1}$  jest wektorem współczynników detali (górnoprzepustowej składowej sygnału). Wektor  $\mathbf{y}_{N \times 1}$  będzie reprezentacją sygnału wejściowego na pierwszym poziomie dekompozycji i będziemy go oznaczali  $\mathbf{y}_{N \times 1}^{(1)}$ .

W następnym kroku, podobnie jak przy dekompozycji falkowej, dekomponowany jest wektor współczynników aproksymacji z poziomu 1, a wektor współczynników detali z poziomu 1 pozostaje bez zmian, itd.

Dekompozycję sygnału wejściowego można wykonać bezpośrednio na wybrany poziom  $k \geq 1$ . Procedura obliczeniowa dekompozycji jest następująca [3]:

$$\mathbf{y}_{N \times 1}^{(k)} = \left[ \mathbf{B}_{\frac{N}{2^k}}^T \oplus \left( \bigoplus_{i=0}^{k-1} \mathbf{B}_{\frac{N}{2^{k-i}}}^T \right) \right] \mathbf{B}_N \mathbf{x}_{N \times 1}. \quad (2)$$

$N/2^k$  początkowych elementów wektora  $\mathbf{y}_{N \times 1}^{(k)}$  to współczynniki aproksymacji sygnału na poziomie  $k$ , następne  $N/2^k$  to współczynniki detali z poziomu  $k$ , kolejne  $N/2^{k-1}$  – współczynniki detali z poziomu  $k-1$ , itd.;  $N/2$  końcowych elementów wektora  $\mathbf{y}_{N \times 1}^{(k)}$  to współczynniki detali z poziomu 1.

Wprowadźmy oznaczenie:

$$\mathbf{F}_N^{(k)} = \left[ \mathbf{B}_{\frac{N}{2^k}}^T \oplus \left( \bigoplus_{i=0}^{k-1} \mathbf{B}_{\frac{N}{2^{k-i}}}^T \right) \right] \mathbf{B}_N. \quad (3)$$

Macierz  $\mathbf{F}_N^{(k)}$  odpowiedzialną za dekompozycję sygnału  $\mathbf{x}_{N \times 1}$  na poziom  $k$  będziemy nazywali macierzą filtracji na poziom  $k$ . Jest ona ortogonalna, tzn.

$$\left( \mathbf{F}_N^{(k)} \right)^{-1} = \left( \mathbf{F}_N^{(k)} \right)^T.$$

Procedura rekonstrukcji sygnału ze współczynników jego dekompozycji na poziomie  $k$  jest następująca:

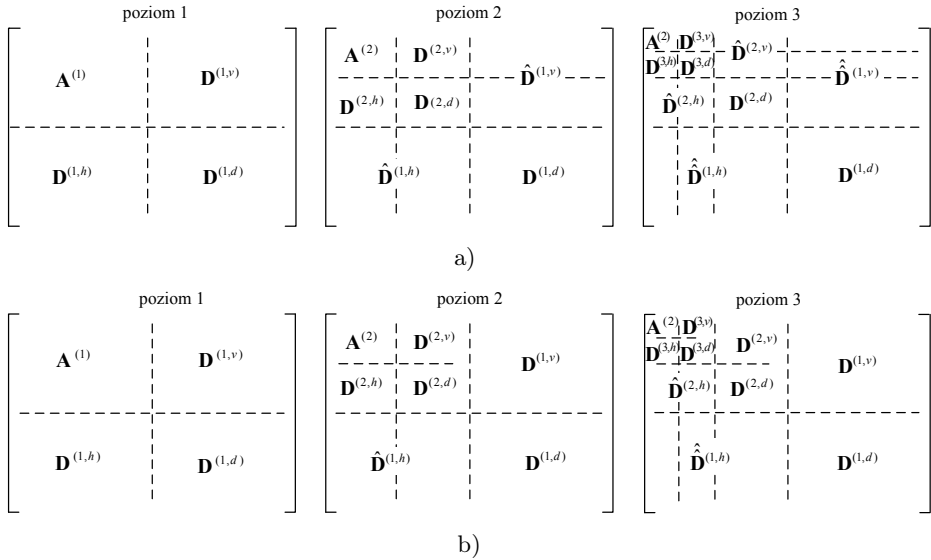
$$\mathbf{x}_{N \times 1} = \left( \mathbf{F}_N^{(k)} \right)^T \mathbf{y}_{N \times 1}^{(k)}, \quad (4)$$

przy czym macierz transformacji odwrotnej ma postać:

$$\left( \mathbf{F}_N^{(k)} \right)^T = \mathbf{B}_N^T \left[ \mathbf{B}_{\frac{N}{2^k}} \oplus \left( \bigoplus_{i=0}^{k-1} \mathbf{B}_{\frac{N}{2^{k-i}}} \right) \right]. \quad (5)$$

Dekompozycja falkowopodobna obrazu, reprezentowanego przez macierz  $\mathbf{X}_{N \times M}$  o  $N$  wierszach i  $M$  kolumnach, może być wykonywana, podobnie jak falkowa, z danego poziomu na poziom następny, przy czym w każdym kroku dokonywana jest falkowopodobna filtracja dolno- i górnoprzepustowa, najpierw wierszy a następnie kolumn macierzy współczynników aproksymacji z poziomu poprzedniego. Analogicznie jak dla sygnałów jednowymiarowych, aby można było wykonać dekompozycję falkowopodobną obrazu na poziom  $k$ , liczby  $N$  i  $M$  określające rozmiar macierzy  $\mathbf{X}_{N \times M}$  muszą być podzielne przez  $2^k$ . Taki, wykonywany krok po kroku, sposób dekompozycji jest jednak mało efektywny, ze względu na konieczność wykonania dużej liczby operacji arytmetycznych. Dla sygnałów jednowymiarowych można dokonać dekompozycji bezpośrednio na wybrany poziom. Czy taki bezpośredni sposób dekompozycji na wybrany poziom jest także możliwy dla obrazów?





Rysunek 2. a) Sposób segmentacji danych 2D po zastosowaniu bezpośredniej dekompozycji falkowopodobnej na poziom 1, poziom 2 i poziom 3 najpierw względem wierszy, a następnie względem kolumn macierzy; b) Sposób segmentacji danych 2D po zastosowaniu bezpośredniej dekompozycji falkowopodobnej na poziom 1, poziom 2 i poziom 3 najpierw względem wierszy, a następnie dekompozycji odpowiednich kolumn macierzy na poziom 1, poziom 2 lub 3

Zastosowanie jednowymiarowej dekompozycji na poziom  $k$  do wierszy a następnie kolumn macierzy reprezentującej obraz cyfrowy daje nieco inny od oczekiwanego, przedstawiony na rysunku 2 a) sposób segmentacji obrazu. Współczynniki detali wertykalnych, oprócz detalu z ostatniego poziomu  $k$ , zostały niepotrzebnie przefiltrowane względem kolumn, zaś współczynniki detali horyzontalnych, oprócz detalu z poziomu  $k$ , zostały niepotrzebnie przefiltrowane względem wierszy. Aby uniknąć tej zbędnej filtracji współczynników detali wertykalnych można, zamiast dekompozycji wszystkich kolumn na poziom  $k$ , wykonać dekompozycję na poziom  $k$  tylko  $M/2^{k-1}$  początkowych kolumn macierzy  $\mathbf{X}_{N \times M}$ , zaś kolejnych  $M/2^{k-1}$  kolumn na poziom  $k - 1$ , następnych  $M/2^{k-2}$  kolumn na poziom  $k - 2$ , itd.  $M/2$  końcowych kolumn powinno być dekomponowane na poziom 1. Otrzymamy wtedy sposób segmentacji obrazu przedstawiony na rysunku 2 b).

Pozostała jeszcze do usunięcia nadmiarowa filtracja współczynników detali horyzontalnych, oprócz detalu z ostatniego poziomu  $k$ , względem wierszy. Wydaje się, że najlepszym sposobem jest w tym wypadku dokonanie filtracji odwrotnej współczynników tych detali:  $\mathbf{D}^{(k-1,h)}$  z poziomu 1,  $\mathbf{D}^{(k-2,h)}$  z poziomu 2, itd.; detal  $\mathbf{D}^{(1,h)}$  z poziomu  $k - 1$ . Współczynniki otrzymane w ten sposób dekompozycji falkowopodobnej obrazu są identyczne jak współczynniki dekompozycji wykonywanej krok po kroku, a liczba operacji arytmetycznych niezbędnych do obliczenia tych współczynników jest znacznie mniejsza.

Aby uzyskać pełną analogię dekompozycji falkowej obrazu i falkowopodobnej należy jeszcze dla wszystkich poziomów pozamieniać miejscami macierze współczynników detali horyzontalnego i wertykalnego z danego poziomu (por. rys. 1a).

Jak już wspomniano dwuwymiarowa dyskretna transformacja falkowa polega na wykonaniu dekompozycji wierszy macierzy  $\mathbf{X}_{N \times M}$  reprezentującej obraz cyfrowy, i wstawieniu współczynników dekompozycji na miejsce owych wierszy, a następnie dekompozycji otrzymanej macierzy względem kolumn i zastąpienie tych kolumn wyznaczonymi współczynnikami. W celu przyspieszenia obliczeń dekompozycja względem poszczególnych wierszy może być wykonywana równoległe, podobnie jak i dekompozycja względem kolumn. Taki równoległy algorytm dekompozycji falkowej w notacji wektorowo-macierzowej został przedstawiony w [8]. Podobnie dwuwymiarowa falkowopodobna dekompozycja macierzy  $\mathbf{X}_{N \times M}$  reprezentującej obraz cyfrowy, jak i rekonstrukcja obrazu ze współczynników jego dekompozycji może być realizowana w sposób równoległy. Rozpatrzmy to zagadnienie bardziej szczegółowo.

Niech  $\mathbf{X}_{N \times M} = [x_{n,m}]$ , gdzie  $n = 0, 1, \dots, N-1$  oraz  $m = 0, 1, \dots, M-1$ , będzie macierzą danych wejściowych reprezentującą obraz cyfrowy. Na podstawie elementów tej macierzy zdefiniujemy wektor

$$\mathbf{X}_{NM \times 1} = [\mathbf{X}_{1 \times M}^{(0)}, \mathbf{X}_{1 \times M}^{(1)}, \dots, \mathbf{X}_{1 \times M}^{(N-1)}]^T,$$

gdzie  $\mathbf{X}_{1 \times M}^{(n)} = [x_{n,0}, x_{n,1}, \dots, x_{n,M-1}]$  – jest podwektorem zawierającym elementy  $n$ -tego wiersza macierzy wejściowej. Wektor  $\mathbf{X}_{NM \times 1}$  jest zwektoryzowaną postacią macierzy  $\mathbf{X}_{N \times M}$ . Wektoryzacja macierzy umożliwia równoległe wykonanie operacji na wszystkich wierszach, a następnie na kolumnach macierzy wejściowej.

Macierz współczynników dekompozycji na  $k$ -tym poziomie reprezentacji  $\mathbf{Y}_{N \times M}^{(k)}$  w postaci zwektoryzowanej oznaczmy  $\mathbf{Y}_{NM \times 1}^{(k)}$ , przy czym macierz ta jest wektoryzowana w ten sposób, że  $N/4^k$  początkowych elementów to ustawione wierszami współczynniki aproksymacji z  $k$ -tego poziomu, a następnie ustawione także wierszami współczynniki detali horyzontalnego, wertykalnego i diagonalnego z poziomu  $k$ , a następnie w analogicznej kolejności, współczynniki detali z poziomów  $k-1, k-2, \dots, 1$ . Procedura obliczeniowa falkowopodobnej dekompozycji obrazu na  $k$ -ty poziom reprezentacji będzie miała postać:

$$\mathbf{Y}_{NM \times 1}^{(k)} = \overline{\overline{\overline{\Psi}}}_{NM}^{(k)} \overline{\overline{\mathbf{P}}}_{NM}^{(k)} \overline{\overline{\Psi}}_{NM}^{(k)} \mathbf{P}_{NM} \Psi_{NM}^{(k)} \mathbf{X}_{NM \times 1}. \quad (6)$$

Poszczególne komponenty macierzowe występujące w (6) są określone w następujący sposób.

$\Psi_{NM}^{(k)} = \mathbf{I}_N \otimes \mathbf{F}_M^{(k)}$  – macierz dekompozycji wierszy na poziom  $k$ , gdzie symbol  $\otimes$  oznacza iloczyn tensorowy Kroneckera [9], zaś  $\mathbf{I}_N$  jest macierzą jednostkową stopnia  $N$ . Na głównej przekątnej, macierz ta ma bloki  $\mathbf{F}_M^{(k)}$  stopnia  $M$ , określone za pomocą (3) odpowiadające za dekompozycję na poziom  $k$  poszczególnych wierszy macierzy  $\mathbf{X}_{N \times M}$  zgodnie z (2).

$\mathbf{P}_{NM} = \underset{i=0}{\overset{M-1}{\blacksquare}} \left( \mathbf{I}_N \otimes \mathbf{e}_{1 \times M}^{(i)} \right)$  – macierz tasowania pierwszego rodzaju, gdzie  $\underset{i=0}{\overset{M-1}{\blacksquare}}$  jest symbolem konkatenacji pionowej [10]  $M$  macierzy indeksowanych przez  $i$ , natomiast  $\mathbf{e}_{1 \times M}^{(i)}$  jest wierzchem o numerze  $i$  macierzy jednostkowej  $\mathbf{I}_M$ .  $\mathbf{P}_{NM}$  jest macierzą monomialną; mnożenie przez nią ma na celu zamianę ustawienia według wierszy na ustawienie według kolumn, w celu umożliwienia dekompozycji względem kolumn.

$\overline{\Psi}_{NM}^{(k)} = \left( \mathbf{I}_{\frac{M}{2^{k-1}}} \otimes \mathbf{F}_N^{(k)} \right) \oplus \left[ \underset{i=k-1}{\overset{1}{\oplus}} \left( \mathbf{I}_{\frac{M}{2^i}} \otimes \mathbf{F}_N^{(i)} \right) \right]$  – macierz dekompozycji kolumn z blokami:  $\mathbf{F}_N^{(k)}, \mathbf{F}_N^{(k-1)}, \dots, \mathbf{F}_N^{(1)}$  na głównej przekątnej, odpowiadającymi za dekompozycję poszczególnych kolumn macierzy  $\mathbf{X}_{N \times M}$  zgodnie z (2). Na poziomie  $k$  dekomponowane są jednak tylko  $M/2^{k-1}$  początkowe kolumny, zaś pozostałe są dekomponowane na poziomy odpowiednio niższe.

$\overline{\mathbf{P}}_{NM}^{(k)}$  – macierz monomialna tasowania drugiego rodzaju; mnożenie przez nią ma na celu zamianę ustawienia według kolumn na ustawienie według wierszy, na początku elementów macierzy współczynników aproksymacji, a następnie elementów macierzy detali horyzontalnego, wertykalnego i diagonalnego z poziomu  $k$ , następnie w analogicznej kolejności ustawione wierszami elementy detali z poziomów  $k-1, k-2, \dots, 1$ . Macierz ta może być utworzona iteracyjnie w następujący sposób. Wprowadźmy macierze pomocnicze:

$$\left\{ \begin{array}{l} \mathbf{Q}_{\frac{NM}{4^j} \times \frac{NM}{2^j}}^{(j)} = \underset{i=0}{\overset{\frac{N}{2^{j-2}}-1}{\blacksquare}} \left( \mathbf{I}_{\frac{M}{2^j}} \otimes \mathbf{e}_{1 \times N}^{(i)} \right), \text{ dla } j = 1, 2, \dots, k, \\ \mathbf{S}_{\frac{NM}{2^{2k-1}} \times \frac{NM}{2^k}}^{(j)} = \mathbf{Q}_{\frac{NM}{4^j} \times \frac{NM}{2^j}}^{(j)} \underset{\frac{N}{2^j}}{\blacksquare} \mathbf{Q}_{\frac{NM}{4^j} \times \frac{NM}{2^j}}^{(j)} \left( \frac{N}{2^j} \rightarrow \right) \end{array} \right.$$

gdzie symbol  $\mathbf{Q}^{(i \rightarrow)}$  oznacza macierz  $\mathbf{Q}$ , której kolumny zostały cyklicznie przesunięte w prawo, względem jej pierwotnej postaci, o liczbę pozycji określoną przez  $i$ , natomiast  $\underset{\frac{N}{2^j}}{\blacksquare}$  jest symbolem konkatenacji pionowej dwóch macierzy [10]. Określamy teraz iteracyjnie macierze:

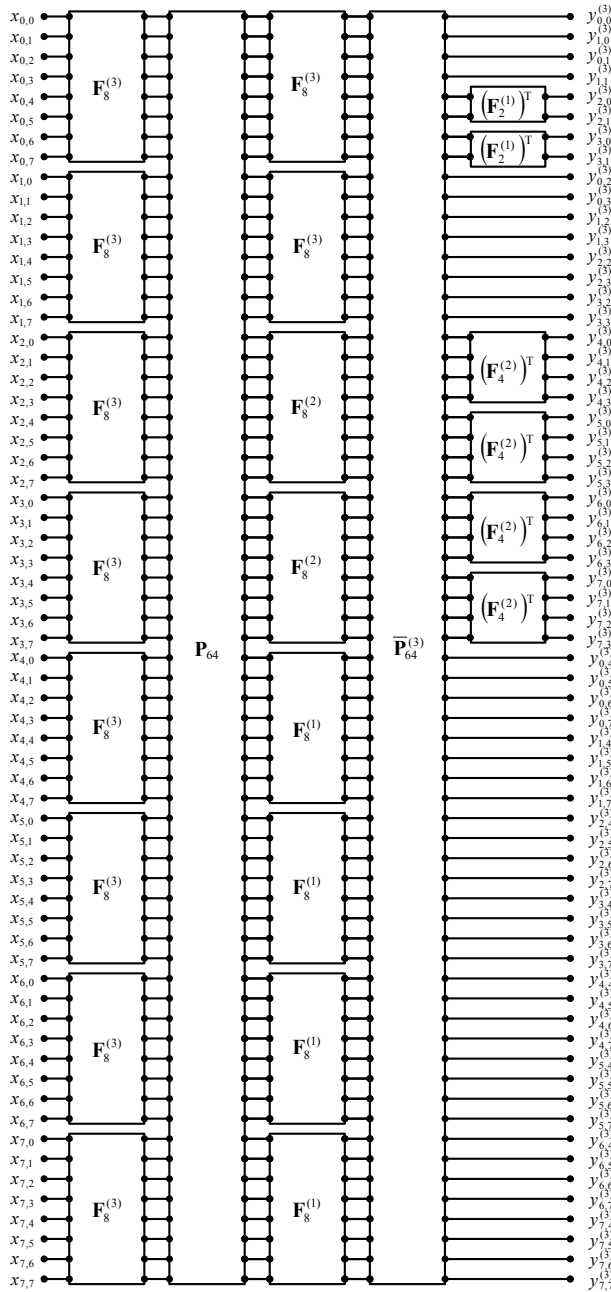
$$\mathbf{U}_{\frac{NM}{4^{k-1}} \times \frac{NM}{2^{k-1}}}^{(1)} = \mathbf{I}_2 \otimes \mathbf{S}_{\frac{NM}{2^{2k-1}} \times \frac{NM}{2^k}}^{(k)},$$

$$\mathbf{U}_{\frac{NM}{4^{k-j}} \times \frac{NM}{2^{k-j}}}^{(j)} = \left( \mathbf{U}_{\frac{NM}{4^{k-j+1}} \times \frac{NM}{2^{k-j+1}}}^{(j-1)} \underset{\frac{N}{2^{k-j+1}}}{\blacksquare} \mathbf{Q}_{\frac{NM}{4^{k-j+1}} \times \frac{NM}{2^{k-j+1}}}^{(k-j+1)} \left( \frac{N}{2^{k-j+1}} \rightarrow \right) \right), \text{ dla } j = 2, 3, \dots, k.$$

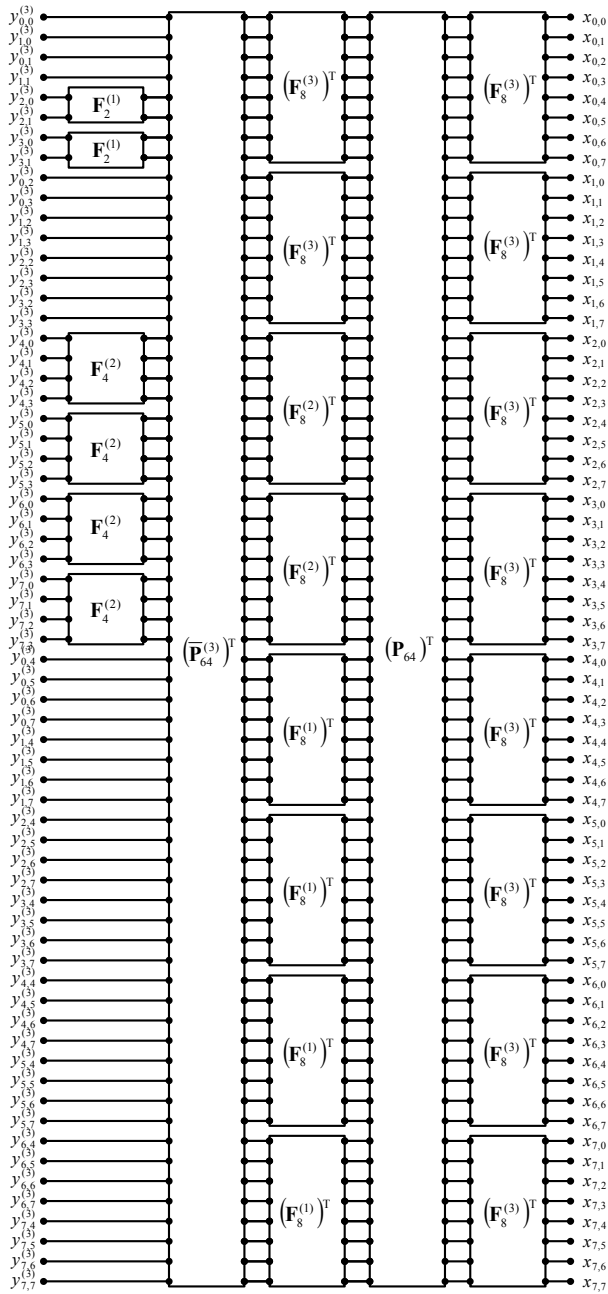
Ostatnia z utworzonych w ten sposób macierzy  $\mathbf{U}_{NM \times NM}^{(k)}$  jest właśnie macierzą tasowania drugiego rodzaju  $\overline{\mathbf{P}}_{NM}^{(k)} = \mathbf{U}_{NM \times NM}^{(k)}$ .

Pozostała jeszcze do określenia macierz  $\overline{\overline{\Psi}}_{NM}^{(k)}$  odpowiedzialna za odwrotną filtrację wierszy współczynników detali horyzontalnych z poziomów  $k-1, k-2, \dots, 1$  (patrz rys. 2 b). Można ją określić w następujący sposób:

$$\overline{\overline{\Psi}}_{NM}^{(k)} = \mathbf{I}_{\frac{NM}{4^{k-1}}} \oplus \left\{ \underset{i=1}{\overset{k-1}{\oplus}} \left[ \left( \mathbf{I}_{\frac{N}{2^{k-i}}} \otimes \left( \mathbf{F}_{\frac{M}{2^{k-i}}}^{(i)} \right)^T \right) \oplus \left( \mathbf{I}_2 \otimes \mathbf{I}_{\frac{NM}{4^{k-i}}} \right) \right] \right\}.$$



Rysunek 3. Model grafostukturalny realizacji obliczeń dla dwuwymiarowej dekompozycji falkowopodobnej według procedury obliczeniowej (6) dla  $N = M = 8$  i  $k = 3$



Rysunek 4. Model grafostukturalny realizacji obliczeń dla dwuwymiarowej rekonstrukcji falkowopodobnej według procedury obliczeniowej (7) dla  $N = M = 8$  i  $k = 3$

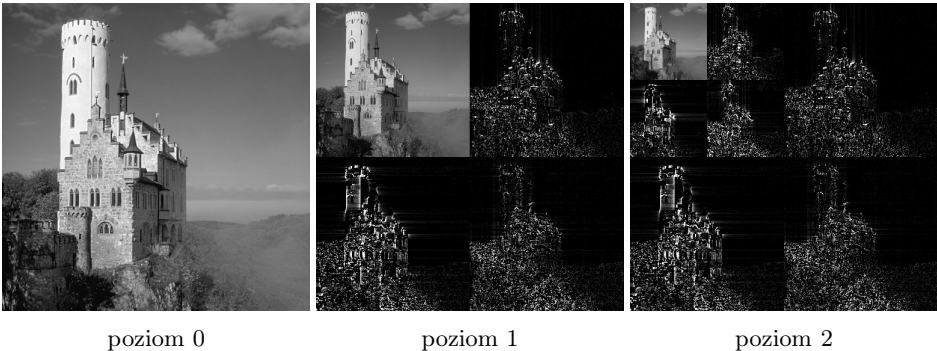
Procedura obliczeniowa rekonstrukcji będzie odbiciem lustrzanym procedury dekompozycji (6), z tym że wszystkie komponenty macierzowe zostaną transponowane. Ma ona postać:

$$\mathbf{X}_{NM \times 1} = \left( \Psi_{NM}^{(k)} \right)^T \left( \mathbf{P}_{NM} \right)^T \left( \overline{\Psi}_{NM}^{(k)} \right)^T \left( \overline{\mathbf{P}}_{NM}^{(k)} \right)^T \left( \overline{\overline{\Psi}}_{NM}^{(k)} \right)^T \mathbf{Y}_{NM \times 1}^{(k)}. \quad (7)$$

Na rysunku 3 został przedstawiony grafostukturalny model realizacji obliczeń dla dwuwymiarowej dekompozycji falkowopodobnej według procedury obliczeniowej (6), zaś na rysunku 4 model realizacji obliczeń dla dwuwymiarowej rekonstrukcji według procedury obliczeniowej (7). Zarówno dla dekompozycji jak i rekonstrukcji przyjęto następujące wartości parametrów:  $N = M = 8$  oraz  $k = 3$ . Indeksy elementów wektorów  $\mathbf{X}_{64 \times 1}$  oraz  $\mathbf{Y}_{64 \times 1}^{(3)}$  na rys. 3 i 4 pochodzą z ich macierzowych postaci  $\mathbf{X}_{8 \times 8}$  i  $\mathbf{Y}_{8 \times 8}^{(3)}$  odpowiednio, w celu ułatwienia interpretacji obliczanych współczynników.

Dekompozycja według (6) daje wyniki bardzo podobne do rezultatów otrzymanych metodą falkową. Na rysunku 5 przedstawiono wyniki dekompozycji falkowopodobnej, otrzymanej według (6), obrazu, którego dekompozycja falkowa została przedstawiona na rys. 1 b). Jako macierz bazy ortogonalnej  $\mathbf{B}$  wykorzystano macierz transformacji DCT-II (z unormowanymi wektorami bazowymi). Macierze  $\mathbf{F}_N^{(i)}$  oraz  $\left( \mathbf{F}_N^{(i)} \right)^T$  zostały określone za pomocą (3) i (5).

Jeżeli jako  $\mathbf{B}$  wykorzystamy macierze transformacji do klasycznych baz ortogonalnych, dla których istnieją szybkie algorytmy wyznaczania transformat, to wyznaczenie dekompozycji falkowopodobnej może okazać się obliczeniowo korzystniejsze niż wyznaczanie tradycyjnej dekompozycji falkowej. Zależy to od rozmiaru obrazu, stopnia dekompozycji oraz długości zastosowanych filtrów falkowych.



Rysunek 5. Dekompozycja falkowopodobna według (6) obrazu „Lichtenstein” [2]. W celu wizualizacji współczynniki poszczególnych składowych zdekomponowanego obrazu zostały przeskalowane tak, aby wykorzystać pełny zakres odcieni szarości.

### 3. Podsumowanie

W artykule zostały przedstawione procedury falkowopodobnej wielopoziomowej dekompozycji oraz rekonstrukcji obrazu. W odróżnieniu od dekompozycji falkowej obraz jest w przedstawionej metodzie dekomponowany bezpośrednio na wybrany poziom  $k$ . Pozwala to, pod warunkiem zastosowanie szybkich algorytmów wyznaczania transformat w klasycznych dyskretnych bazach ortogonalnych, przyspieszyć proces realizacji wielopoziomowej dekompozycji sygnału w stosunku do tradycyjnej dekompozycji falkowej, zwłaszcza jeśli poziom dekompozycji jest wysoki.

Ograniczeniem metody jest to, że liczby określające rozmiar obrazu muszą być podzielne przez  $2^k$ ; nie jest to wymagane przy dekompozycji falkowej. Przedstawiony sposób reprezentacji obrazu może być wykorzystany np. do kompresji stratnej lub odsumiania obrazów. Jak wynika z badań autorów możliwości kompresji przy zastosowaniu przedstawionej metody są podobne jak dla metody pakietów falkowych, przy zastosowaniu filtrów o średnich długościach, a przedstawiona metoda może być obliczeniowo korzystniejsza od metody falkowej ze względu na redukcję liczby operacji arytmetycznych, przy czym stopień redukcji zależy od rozmiarów obrazu, poziomu dekompozycji oraz długości zastosowanych filtrów falkowych.

### Bibliografia

- [1] Zieliński T. P. *Cyfrowe przetwarzanie sygnałów*, WKŁ, Warszawa 2005.
- [2] [http://commons.wikimedia.org/wiki/Image:Jpeg2000\\_2-level\\_wavelet\\_transform-lichtenstein.png](http://commons.wikimedia.org/wiki/Image:Jpeg2000_2-level_wavelet_transform-lichtenstein.png)
- [3] Țariova G., Țariov A. *Wielopoziomowa reprezentacja sygnałów w dyskretnych bazach ortogonalnych*, Metody informatyki stosowanej w zarządzaniu - rocznik NR 6, Informa, Szczecin 2004, s. 209-213.
- [4] Țariov A., Țariova G., Umiński W. *Generalized procedures of wavelet-like transformations in discrete orthogonal bases*, Polish Journal of Environmental Studies, vol. 15, no. 4C, 2006, pp. 188-190.
- [5] Țariov A., Majorkowska-Mech D. *The multilevel signal representation in discrete base of cosine functions*, Elektronika konstrukcje technologie zastosowania, Rok XLVIII, Nr 7/2007, s. 20-21.
- [6] Lee B. *A new algorithm to compute the discrete cosine transform*, IEEE Trans. Acoust. Speech Signal Process, 32, 1984, pp. 1243-1245.
- [7] Hou H. S. *A fast recursive algorithm for computing the discrete cosine transform*, IEEE Trans. Acoust. Speech Signal Process, 35, 1987, pp. 1455-1461.
- [8] Țariov A., Țariova G., Majorkowska-Mech D. *Computational Procedures for Two-dimensional Wavelet Transform in Matrix Notation*, Polish Journal of Environmental Studies, Vol. 16, No. 4A, 2007, pp. 347-350.
- [9] Graham A., *Kroneker Products and Matrix Calculus with Applications*, John Willey & Sons, Ltd, placeChichester, 1981.
- [10] Țariov A. *Modele algorytmiczne i struktury wysokowydajnych procesorów cyfrowej obróbki sygnałów*, Informa, Szczecin 2001, (w języku rosyjskim).





# Problem obiektów nietypowych w badaniach taksonomicznych

Kesra Nermend

*Uniwersytet Szczeciński, Wydział Nauk Ekonomicznych i Zarządzania*

## Abstract:

*The article is related with the problem of the influence of untypical objects on classification results in taxonomic research. It is proposed to construct patterns on the basis of first and third quartiles. In a consequence the influence of untypical objects on a synthetic measure value for other objects can be reduced.*

## Słowa kluczowe:

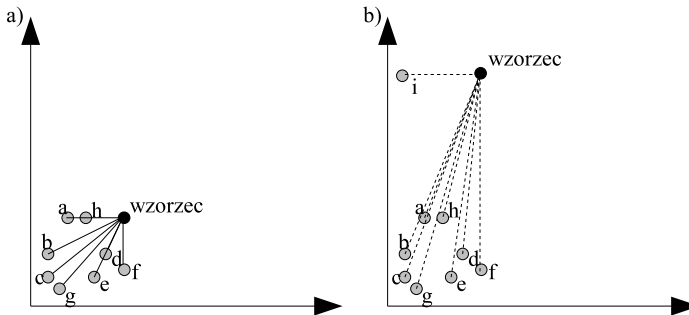
*Quartiles, taxonomic.*

## 1. Wstęp

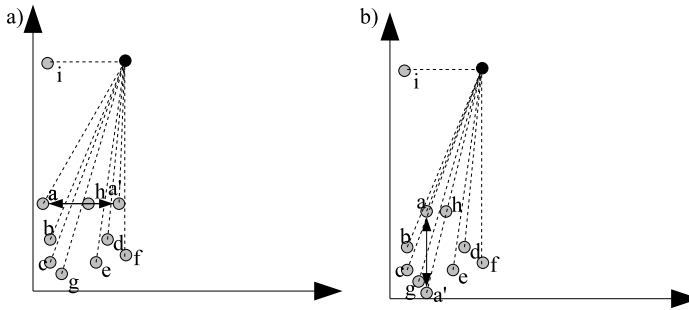
W każdym badaniu mogą wystąpić obiekty mocno różniące się od pozostałych, chociaż jest to mało prawdopodobne. Badając populację ludzi pod względem wzrostu możemy czasami natrafić na osoby nienaturalnie wysokie lub nienaturalnie niskie. Biorąc pod uwagę niewielką próbę badawczą możemy się spodziewać, że takie osoby nie wystąpią, ale nie powinniśmy całkowicie wykluczyć ich pojawienia się. W badaniach większość wskaźników opisujących obiekty mieści się w pewnym przedziale, jednak zawsze należy zakładać możliwość istnienia obiektów spoza tego przedziału. Pojawienie się obiektów z nietypowymi wartościami jest często przypadkowe i zależy od wyboru badanych obiektów. Przypadkowość pojawiania się tych obiektów rodzi poważne problemy przy tworzeniu wzorca.

## 2. Problem obiektów nietypowych

W badaniach taksonomicznych wzorzec budowany jest jako maksimum zmiennych które są stymulantami i minimum dla zmiennych które są destymulantami. Jego ułożenie zależy zatem od położenia najbardziej skrajnie położonych elementów. Rysunek 1a przedstawia pewien zbiór obiektów, dla którego wyznaczono wzorzec. Dla uproszczenia przyjęto, że wszystkie zmienne są stymulantami, a jako miarę podobieństwa obrano odległość euklidesową, która jest łatwa do interpretacji geometrycznej. Najbliżej wzorca znajduje się obiekt oznaczony literą d, drugie miejsce zajął obiekt h, trzecie f, a czwarte a. Założmy, że rozszerzymy zbiór badanych obiektów o obiekt nietypowy oznaczony literą i. Wzorzec ulegnie silnemu przesunięciu. Ilustruje to rysunek 1b. Przesunięcie wzorca będzie miało wpływ na



Rysunek 1. Zależność położenia wzorca od zbioru badanych obiektów: a) położenie wzorca dla zbioru obiektów bez nietypowego obiektu, b) położenie wzorca dla zbioru obiektów z nietypowym obiektem

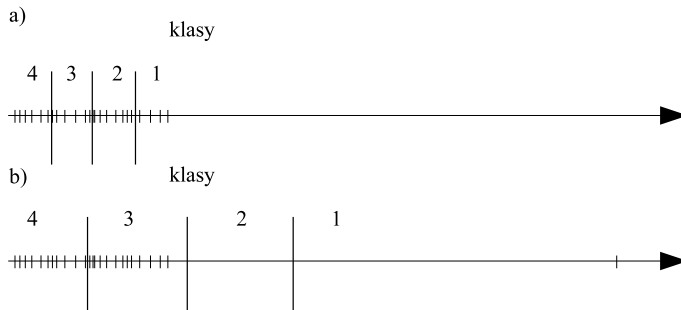


Rysunek 2. Wpływ przesunięcia obiektu na kolejność uszeregowania względem wzorca, a) przesunięcie wzdłuż osi poziomej, b) położenie wzdłuż osi pionowej

uszeregowanie elementów. Pierwszym elementem będzie obiekt i, drugim h trzecim a, a czwartym d.

Widzimy więc, że wynik uszeregowania elementów bardzo silnie zależy od dołączenia lub nie dołączenia obiektu nietypowego i. Jeżeli będziemy przesuwać obiekt a wzdłuż osi poziomej (rysunek 2a) to jego pozycja w rankingu najbardziej podobnych do wzorca nie ulegnie zmianie, jeżeli natomiast będziemy przesuwać go w tym samym przedziale wzdłuż osi pionowej (rysunek 2b) to obiekt będzie zmieniał swoją pozycję w rankingu od miejsca drugiego do czwartego. Widać więc, że waga zmiennej związanej z osią pionową i poziomą są różne. Zmienna, wzdłuż której obiekt nietypowy ma bardzo odstającą od pozostałych wartość będzie miała większą wagę.

Należy się teraz zastanowić jaki wpływ będzie miała standaryzacja wskaźników na szeregowanie obiektów wśród których znajduje się obiekt nietypowy. Generalnie każdy rodzaj standaryzacji powoduje zmianę skali zmiennych oraz przesunięcie wartości. Głównym celem standaryzacji jest zawsze doprowadzenie do takiej zmiany skali zmiennych, aby wpływ wszystkich zmiennych na wynik porównania



Rysunek 3. Klasyfikacja obiektów, a) bez obiektu nietypowego, b) z obiektem nietypowym

ze wzorcem był jednakowy. Pojawienie się wśród standaryzowanego zbioru danych wartości mocno odstającej od pozostałych będzie miało wpływ na wynik standaryzacji.

Niektóre metody standaryzacji mogą słabo reagować na pojawienie się wartości nietypowych. Oznacza to, że dodanie wartości nietypowej niewiele wpłynie na pozostałe wartości. Przy tworzeniu wzorca powstanie więc sytuacja identyczna do opisanej powyżej. Istnieją metody standaryzacji, które ściągają dane do pewnego zadanego przedziału np. zakresu wartości 0 i 1. Oznacza to, że najmniejsza wartość zmiennej będzie miała wartość 0, a największa 1. Pojawienie się wartości mocno odstającej od pozostałych spowoduje, że pozostałe wartości zostaną mocniej ściśnięte. W konsekwencji doprowadzi to do zwiększenia wagi tej zmiennej, a więc powstania sytuacji analogicznej do powyższej. Standaryzacja, oprócz zmiany skali, powoduje też przesunięcie wartości. Pojawienie się wartości nietypowej spowoduje silniejsze przesunięcie pozostałych wartości. Na miary podobieństwa, których wartość nie zależy od położenia obiektów w przestrzeni, a jedynie od ich wzajemnego ułożenia, nie będzie to miało wpływu.

Pojawienie się obiektu nietypowego będzie miało również wpływ na klasyfikację pozostałych obiektów. Przyglądając się rysunkowi 1b możemy zauważyć, że obiekt nietypowy będzie miał wartość miary syntetycznej znacząco odbiegającą od pozostałych obiektów. Wynika to z jego oddalenia od pozostałych obiektów. W konsekwencji na osi wartości miary syntetycznej będzie się on znajdował w znacznej odległości od pozostałych obiektów. Jeżeli wykonamy klasyfikację na podstawie odchylenia standardowego to otrzymamy rozszerzenie szerokości klas w ten sposób, że niektóre z nich mogą wejść na obszar pomiędzy obiektem nietypowym, a pozostałymi obiektami, gdzie nie znajdują się żadne obiekty (rysunek 3). W skrajnym przypadku wynikiem klasyfikacji mogą być dwie klasy nie zawierające żadnych obiektów (2 i 3) oraz jedna (1) zawierająca obiekt nietypowy, a także czwarta (4) zawierająca wszystkie pozostałe obiekty. W tej sytuacji klasyfikacja traci sens i trzeba uciec się do bardziej zaawansowanych metod grupowania.

Interpretacja uzyskanych wyników w przypadku kiedy pojawiają się obiekty nietypowe jest bardzo trudna. Waga zmiennych zależy od „nietypowości” obiektu,

a więc im bardziej wartość zmiennej opisująca ten obiekt odstaje od pozostałych tym jej waga jest większa. Założmy, że klasyfikujemy powiaty o charakterze wiejskim pod względem emisji i neutralizacji dwutlenku węgla. Bierzymy pod uwagę czynnik pozytywny (powierzchnia lasów) oraz czynniki negatywne (emisja dwutlenku węgla przez zakłady przemysłowe, komunikacja, piece centralnego ogrzewania). W powiatach wiejskich na ogół nie ma dużych zakładów przemysłowych, więc emisja dwutlenku węgla nie powinna być na zbyt wysokim poziomie. Jednak czasami zdarza się, że ze względu na dużą uciążliwość ekologiczną są tam lokalizowane ogromne zakłady przemysłowe, przykładowo elektrownie. Jeżeli w obrębie jakiegokolwiek powiatu znajdzie się taki zakład, powiat ten stanie się powiatem nietypowym. Automatycznie, niezależnie od wielkości obszarów zalesionych, wszystkie powiaty zostaną sklasyfikowane w oparciu o emisję dwutlenku węgla.

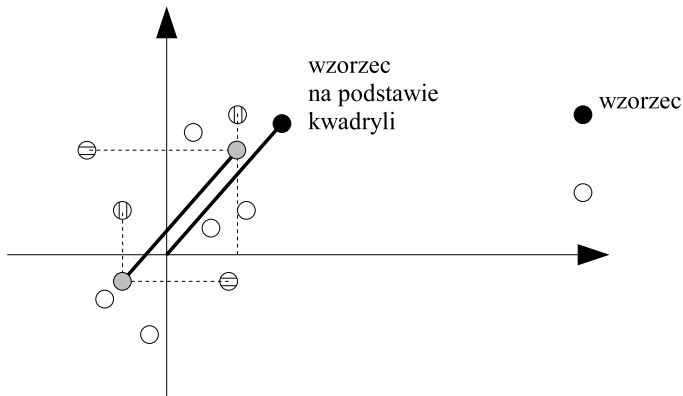
Należy zauważyć, że problem stwarza nie tylko pojawienie się obiektów nietypowych. Czasami może się zdarzyć, że wzdłuż jakiejś zmiennej obiekty są podzielone na dwie grupy wyraźnie od siebie oddalone. Obiekty w grupie oddalony od wzorca będą uszeregowane ze względu na tę zmienną, z pominięciem pozostałych, natomiast obiekty w grupie w pobliżu wzorca będą uszeregowane ze względu na wszystkie zmienne. Prowadzi to do porównywania obiektów w obu grupach względem różnych czynników.

W klasycznym podejściu, aby ustrzec się od opisanych wyżej problemów, konieczne jest aby zbiór obiektów był zbiorem jednego typu. Nie powinno się mieszać obiektów różnych typów (np. powiatów wiejskich i miejskich). W ramach obiektów tego samego typu ważne jest aby nie wystąpiły obiekty o anormalnych wartościach zmiennych.

### 3. Budowa wzorca na podstawie kwartyli

Problem z obiektami nietypowymi wynika z zasady budowy wzorca. Wybór obiektów o najwyższych (lub najniższych) wartościach zmiennych ma doprowadzić do stworzenia obiektu idealnego o najlepszych parametrach. Jednak z reguły najwyższe wartości zmiennych osiąągają obiekty nietypowe. Co więcej, obiekty te mają tak wysokie wartości z reguły dla niewielkiej liczby zmiennych, a często nawet jednej zmiennej. Jeżeli nie wzięlibyśmy tej zmiennej pod uwagę to może okazać się, że obiekt ten należy do obiektów średnich lub nawet słabych. W związku z tym obiekty te nie bardzo nadają się do tworzenia wzorca.

Tworząc miarę syntetyczną należałoby przeprowadzić analizę wartości zmiennych w celu zbadania jednorodności zbioru obiektów (określenia czy należą one do tego samego typu, oraz czy występują obiekty nietypowe). Analiza taka nie zawsze jest prosta do przeprowadzenia. W typowej sytuacji nie ma wyraźnego podziału na grupy, a obiekty nietypowe mają wartości odstające od pozostałych, jednak nie muszą znajdować się bardzo daleko od całego zbioru obiektów. Powstaje zatem problem określenia kiedy dany obiekt należy uznać za nietypowy. Podobny problem występuje przy określeniu czy wszystkie obiekty należą do tego samego typu. Jeżeli występuje więcej niż jeden typ obiektu to w obszarze przejścia między typami znajduje się niewielka liczba obiektów pośrednich.

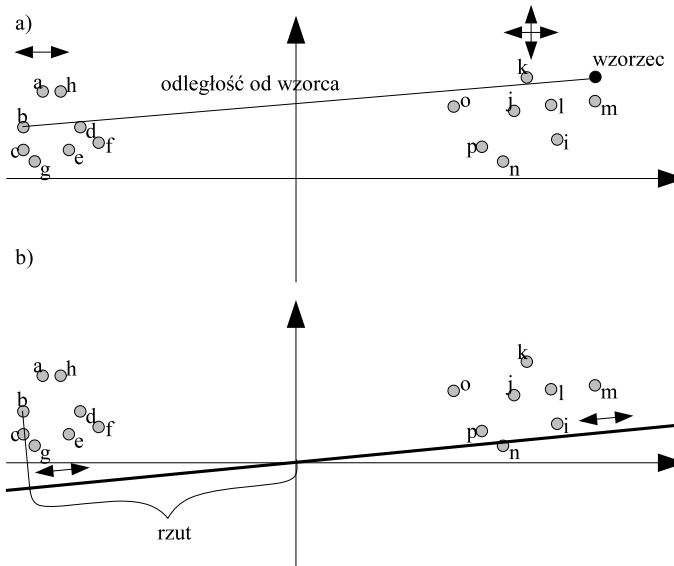


Rysunek 4. Porównywanie obiektów przy wzorcach wyliczanych jako wartość maksymalna i założeniu, że istnieją obiekty o anormalnie wysokich wartościach zmiennych

Alternatywnym do analizy jednorodności obiektów rozwiązaniem, może być inny sposób tworzenia wzorca. Wzorec powinien być wyłoniony wśród obiektów typowych, jako najlepszy obiekt typowy. Obiekty typowe znajdują się z reguły w środku zbioru. Możemy założyć, że granicą obiektów typowych jest pierwszy i trzeci kwartył [1] (kwartyle pierwszy i trzeci są także nazywane dolnym i górnym [8]). Kwartyle te wyznaczają dwa wzorce — pierwszego i trzeciego kwartyla. Różnica współrzędnych tych wzorców daje współrzędne wzorca na podstawie kwartyli. Przy czym, dla stymulanty odejmujemy od wzorca trzeciego kwartyla wzorec pierwszego kwartyla, dla destymulanty kolejność odejmowania jest odwrotna.

Na rysunku 4 na szaro zaznaczono wzorce kwartyli 1 i 3. Na tej podstawie, dzięki różnicy współrzędnych, powstał wzorec w oparciu o kwartyle. Przyglądając się położeniu wzorca powstałego na podstawie kwartyli oraz wzorca liczonego w oparciu o wartości maksimum i minimum, możemy zauważyć, że wzorec na podstawie kwartyli zmniejsza wagę zmiennej 2, przez co zmienna 1 i zmienna 2 będą podobnie wpływać na wartość miernika.

Przyglądając się rysunkowi 4 możemy zauważyć, że wzorec na podstawie kwartyli nie jest najlepszym możliwym obiektem. W badaniach taksonomicznych z założenia obiekt-wzorec musi być obiektem najlepszym w badanej grupie [2, 3, 4, 7, 9]. Niespełnienie tego warunku spowoduje, że obiekty lepsze od wzorca otrzymają tym niższą wartość miary syntetycznej im są lepsze od wzorca. Problem ten wynika z odległościowego charakteru stosowanych do budowy miernika miar podobieństwa. Istnieją dwa sposoby rozwiązania tego problemu. Pierwszy polega na wyznaczeniu równania prostej łączącej wzorec z początkiem układu współrzędnych i przesunięciu go wzdłuż prostej tak, aby wartości zmiennych opisujących jego położenie były większe lub równe w stosunku do wartości zmiennych opisujących wszystkie obiekty.



Rysunek 5. Wpływ przesunięcia obiektów na wartości miary syntetycznej, a) dla standardowej miary syntetycznej, b) dla miary syntetycznej liczonej na podstawie rzutu

Drugim sposobem rozwiązania problemu obiektów lepszych od wzorca jest zastosowanie zamiast miar podobieństwa o charakterze odległościowym rzutu [5, 6]. W przypadku rzutu obiekty lepsze od wzorca otrzymują wartość miary większą od jedności. Dodatkowo zaletą rzutu jest jednolite traktowanie wszystkich obiektów. Ilustruje to rysunek 5. Jeżeli badany zbiór obiektów składa się z obiektów różnego typu, to jak to napisano wcześniej, obiekty należące do typu leżącego bliżej wzorca są klasyfikowane według wszystkich zmiennych, natomiast obiekty należące do typu leżącego dalej od wzorca, tylko względem jednej zmiennej. Na rysunku 5a strzałkami zaznaczono kierunki przesuwania obiektów obu typów które powoduje zmianę ich uszeregowania w ramach typu.

W przypadku rzutu kierunek przesuwania obiektów, wzdłuż którego zmieniają one przynależność do klas zależy tylko i wyłącznie od położenia wzorca (rysunek 5a), nie zależy natomiast absolutnie od odległości obiektów od wzorca. Zatem wszystkie obiekty szeregowane są według tych samych zasad.

#### 4. Podsumowanie

Pojawienie się obiektów nietypowych w zbiorze obiektów powoduje zmianę wartości miernika syntetycznego wszystkich obiektów, co ma wpływ na kolejność ich uszeregowania. Częstość występowania obiektów nietypowych jest niewielka, a ich przynależność do grupy badanych obiektów zależy często jedynie od ram narysowanych przez badacza. Jeżeli nie stosuje się dodatkowych metod wykrywania i eliminacji tych obiektów to ich pojawienie się możemy traktować jako czynnik

losowy. Przy bardzo nietypowych obiektach uszeregowanie badanych obiektów zależy często bardzo silnie od ich pojawienia się czyli od czynnika losowego. Zatem dwa niezależne badania mogą bardzo silnie różnić się jeżeli zbiór badanych obiektów różni się o ten obiekt. Zastosowanie kwartyli do tworzenia wzorca praktycznie eliminuje wpływ obiektów nietypowych na wynik. Zapewnia zatem bardzo silną porównywalność wyników.

## Bibliografia

- [1] Aczel A. D. *Statystyka w zarządzaniu*. Warszawa, PWN 2000
- [2] Chudy K., Wierzbińska M. *Taksonomiczna analiza rozwoju społeczno-gospodarczego woj. podkarpackiego*. Wiadomości statystyczne, 2:77–83 2001
- [3] Hellwig Z. *Zastosowania metody taksonomicznej do typologicznego podziału krajów ze względu na poziom ich rozwoju oraz zasoby i strukturę wykwalifikowanych kadr*. Przegląd Statystyczny, 2 1977
- [4] Nermend K. *A synthetic measure of sea environment pollution*. Polish Journal of Environmental Studies, 15(4B):127–130 2006
- [5] Nermend K. *Regions grouping with similarity measure based on vector calculus*. Polish Journal of Environmental Studies, 16(5B):132–136 2007
- [6] Nermend K. *Taxonomic vector measure of region development (twmrr)* Polish Journal of Environmental Studies, 16(4A):195–198 2007
- [7] Pluta W. *Wielowymiarowa analiza porównawcza w badaniach ekonomicznych*. Warszawa, PWE 1977
- [8] Smirnow N. W., Dunin-Barkowski I. W. *Krótki kurs statystyki matematycznej dla zastosowań technicznych*. Warszawa, PWN 1966
- [9] Sobczak E. *Metody wielowymiarowej analizy porównawczej w kwantyfikacji konkurencyjności regionów*. Wrocław, Akademia Ekonomiczna we Wrocławiu, nr 885 2001





# Identyfikacja człowieka na podstawie zestawu obrazów twarzy pochodzących z sekwencji video

Adam Nowosielski

*Politechnika Szczecińska, Wydział Informatyki*

## **Abstract:**

*Traditional face recognition is based on the static image analysis. In a real-life scenario there is a sequence of consecutive images from a video camera. By tracking a face in a sequence and aggregating individual results of face recognition, a much more stable and reliable final recognition result may be generated. Based on the analysis of the problem the algorithm for people identification from a video stream is developed.*

## **Słowa kluczowe:**

*people identification, face recognition, video analysis*

## **1. Wprowadzenie**

Wykorzystywanie systemów biometrycznych w życiu codziennym jest coraz większe i stale rośnie. Powodem takiego stanu jest zarówno zwiększenie wymagań odnośnie bezpieczeństwa, jak i praktyczne możliwości wykorzystania zalet technik biometrycznych. Szczególną rolę pełni tutaj twarz człowieka. Kolejne raporty *International Biometric Group* [1, 2] pokazują, że wzrost udziału w rynku używała twarz (aż o 7%) i technologie wykorzystujące jednocześnie kilka cech biometrycznych. Twarz została także uznana za bazową technologię w dokumentach biometrycznych [4, 17].

Technologie wykorzystujące twarz są nieinwazyjne, bezkontaktowe i chyba najbardziej naturalne spośród wszystkich metod identyfikacji człowieka. Nie ograniczają w żaden sposób ruchów użytkownika. Systemy takie oferują nową, nieznaną do tej pory jakość. Dzięki identyfikacji człowieka przede wszystkim zwiększyć można poziom bezpieczeństwa oraz ułatwić życie poprzez tworzenie inteligentnych środowisk [18].

Dodatkowo w nowoczesnych systemach przechodzi się z analizy statycznych obrazów na rzecz analizy obrazów video. Identyfikacja osób na podstawie obrazów twarzy pochodzących z sekwencji video ma wiele zalet w porównaniu z tradycyjnym sposobem identyfikacji bazującym na pojedynczym obrazie [3]. Prowadzi do:

- zwiększenia skuteczności działania systemu,
- przyspieszenia procesu detekcji twarzy,

— stabilizacji wyników.

W dalszej części artykułu zostaną przedstawione możliwości oraz zalety wykorzystania sekwencji wideo w procesie identyfikacji ludzi na podstawie obrazu twarzy. Dyskusji poddana zostanie powszechnie panująca opinia, że systemy monitoringu świetnie nadają się do adaptacji do zadania rozpoznawania twarzy. Zaproponowana zostanie także strategia stabilizacji i zwiększenia wiarygodności wyników w systemach rozpoznawania twarzy wykorzystujących sekwencje wideo.

## 2. Zastosowania systemów rozpoznawania twarzy

Ze względu na realizowane zadanie systemy rozpoznawania twarzy dzielimy na systemy weryfikacji tożsamości i systemy identyfikacji.

Tradycyjne rozwiązania uwierzytelniania bazują na wiedzy (hasła, kody) lub posiadaniu (karty magnetyczne, tokeny, itp.). Mogą być one w łatwy sposób oszukane, ponieważ nie gwarantują, że osoba posługująca się dla przykładu kartą kredytową i znająca PIN jest prawowitym właścicielem posiadanej karty. Dzięki biometrycznemu systemowi weryfikacji sprawdzenie użytkownika pod względem tego „co ma?” lub „co wie?” można zastąpić sprawdzeniem „kim jest?”. Rozwiązania, które należy wymienić w tym miejscu to: systemy do porównywania fotografii w dokumentach tożsamości z ich posiadaczami, systemy kontroli dostępu do sieci i pomieszczeń (ang. *access control systems*), systemy bankowe i systemy uwierzytelniania na potrzeby transakcji finansowych.

W biometrycznych systemach identyfikacji na podstawie twarzy dokonuje się próby ustalenia tożsamości (identyfikowania) osoby na podstawie pobranej cechy biometrycznej i bazy wzorców. Systemy takie oferują nową, nieznaną do tej pory jakość. Dzięki identyfikacji człowieka przede wszystkim zwiększyć można poziom bezpieczeństwa oraz ułatwić życie poprzez tworzenie inteligentnych środowisk [18]. Cały czas trwają próby automatycznego wyszukiwania terrorystów na lotniskach na podstawie obrazów twarzy. Znaleźć można także zupełnie nowe zastosowania: sklep rozpoznający swoich najlepszych klientów, elektroniczne punkty informacyjne pamiętające swoich użytkowników (np. turystów), inteligentny budynek „znający” ludzi, którzy w nim mieszkają..

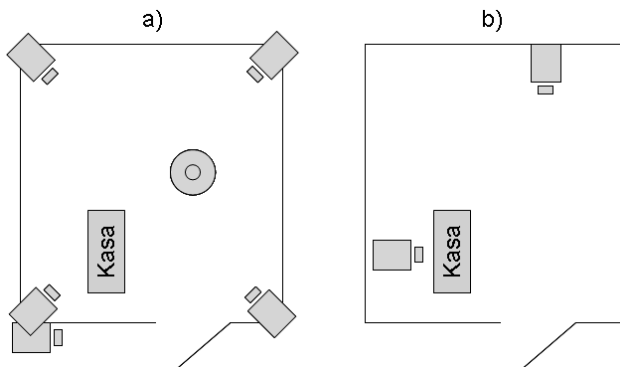
Obecne trwają intensywne prace nad wyposażaniem maszyn w zdolności percepcyjne. Rozpoznawanie twarzy jest istotne jeżeli chodzi o zadania interpretacji wyrazów twarzy, emocji ludzi, intencji i zachowania. Czynniki te stanowią kluczową koncepcję inteligentnych środowisk (ang. *smart environments*) [18]. W środowisku takim, dzięki mowie ciała, system potrafiłby np. dostosować szybkość pokazywania informacji. Znudzenie wskazywałoby, że informacja pokazywana jest za wolno lub jest zbyt nudna. Zmieszany wyraz twarzy wskazywałby, że informacja pokazywana jest za szybko [18].

Jak widać na przedstawionych przykładach potencjalne zastosowanie obrazu twarzy jest ogromne. Zakres zastosowań obejmuje zadania od statycznego porównywania fotografii aż do identyfikacji człowieka z sekwencji wideo w czasie rzeczywistym. Ta ostatnia kwestia poddana będzie analizie w dalszej części artykułu.

### 3. System rozpoznawania twarzy w systemie monitoringu

Przyczynami błędów powstających w systemach biometrycznych wykorzystujących twarze są problemy, które można zaklasyfikować do jednej z następujących grup [20]: przesłonięcia, kąt obserwacji twarzy (poza), rozmiar (skala), warunki zewnętrzne (oświetlenie i tło), mimika, czynniki zakłócające (np.: okulary, zarost, makijaż). O ile nie jesteśmy w stanie zapanować nad mimiką obserwowanego człowieka i czynnikami zakłócającymi, możemy operować czynnikami środowiskowymi. Spójrzmy na to zagadnienie bliżej.

W literaturze przedmiotu spotkać można opinie mówiące, że systemy monitoringu bez problemu można zaadaptować do zadania rozpoznawania twarzy. Cała bowiem infrastruktura doskonale nadaje się do adaptacji do omawianego zadania. Istnieją bowiem: kamery, okablowanie oraz sprzęt rejestrujący (przyjmujący coraz częściej formę cyfrową – w postaci komputera, a nie analogową). Wydawać mogłoby się, że dodanie kolejnego modułu – realizującego zadanie identyfikacji nie powinno stanowić problemu. Czy na pewno?



Rysunek 1. Przykłady rozmieszczenia kamer w sklepie

Przyjrzyjmy się schematowi a) przedstawionemu na rys. 1. Schemat przedstawia rozmieszczenie kamer w systemie monitoringu w pomieszczeniu na przykładzie sklepu. Kamery są tak rozlokowane, aby swoim zasięgiem pokryć możliwie duży obszar. Stąd najczęściej montuje się je w rogach pomieszczenia lub na suficie (kamery szerokokątne skierowane w dół). Taka lokalizacja nie jest korzystna z punktu widzenia systemu rozpoznawania twarzy. Twarze są bowiem rejestrowane pod znacznymi kątami i często są źle oświetlone. Zadanie rozpoznawania twarzy jest niezwykle skomplikowane w takich warunkach [19]. Różnice w oświetleniu oraz kącie obserwacji (kierunku patrzenia) wpływają na projekcje trójwymiarowego obiektu (także do pewnego stopnia elastycznego) twarzy na obrazie 2D. Mogą być one większe w ramach jednej klasy (osoby) niż pomiędzy różnymi klasami (osobami) rejestrowanymi w tych samych warunkach [21, 22]. Ludzie mają bowiem podobne twarze strukturalnie. Z drugiej jednak strony geometria twarzy jest

bardzo ograniczona i głównie zawęża się do dwóch wymiarów, gdy ludzie patrzą w stronę kamery [18].

W związku z powyższymi należy zaproponować „nietypową” (z punktu widzenia systemu monitoringu) lokalizację kamer w systemie. Urządzenia rejestrujące obraz powinny być tak rozmieszczone, aby przechwytywały obrazy twarzy patrzących na wprost. Sytuację tą schematycznie przedstawia schemat b) na rys. 2. Występują tu dwie kamery: pierwsza naprzeciw wejścia, druga w okolicy kasy sklepu.

Na szczególną uwagę zasługuje tutaj kamera umieszczona na wprost wejścia. W przypadku tradycyjnej kamery z systemu monitoringu twarze osób wchodzących do pomieszczenia mogą być zbyt małe do celów rozpoznawania. Można zalecić w takiej sytuacji stosowanie kamer megapikselowych lub kamer z dużym zoomem optycznym. W pierwszym przypadku, obraz jest wysokiej rozdzielczości – co za tym idzie, wyekstrahowany wycinek zawierający twarz będzie spełniał wymogi odnośnie rozmiaru. W przypadku kamer z zoomem optycznym jakość (rozdzielczość) samego obrazu nie musi być wysoka, bowiem dzięki możliwości zbliżenia i odpowiedniego wykadrowania kamery, otrzymywany obraz sceny zawiera już twarze wysokiej rozdzielczości.

W tym kontekście warto jeszcze wspomnieć o „pułapkach twarzy” (ang. *Face-Trap*) [20]. Są to miejsca, budzące zainteresowanie, przyciągające uwagę, w które ludzie kierują wzrok. W miejscach takich umieszcza się niewidoczne kamery. Przy odpowiednim ustawieniu oświetlenia uzyskać można niemal idealne warunki do realizacji zadania rozpoznawania twarzy. Twarz bowiem podążając za wzrokiem jest w pozycji *en face* do kamery. Bardzo łatwo jest przy tym zapewnić właściwe oświetlenie. Można nawet rozważyć w takiej sytuacji instalację kamer z oświetlaczami podczerwieni. Kamery takie emitują światło niewidoczne dla człowieka, poprawiając jakość uzyskiwanych obrazów (zwłaszcza w złych warunkach oświetleniowych).

„Pułapkami twarzy” mogą być kamery instalowane na kasach przy wyświetlaczu pokazującym cenę do zapłaty. W momencie sprawdzenia przez kupującego ceny zakupów następuje proces rozpoznawania.

Inny przykład to instalacja kamer w różnego rodzaju czytnikach. Wchodząc np. do metra należy włożyć bilet do kasownika lub przyłożyć kartę magnetyczną. Naturalnym odruchem jest kierowanie wzroku (i twarzy) na czytnik.

W [20] opisano przykład lotniska, gdzie kamerę umieszczono w tarczy zegara. Podróżni czekający na odprawę często kierują swój wzrok w stronę zegara.

#### 4. Sekwencja wideo w rozpoznawaniu

Problem identyfikacji ludzi na podstawie obrazów twarzy otrzymanych z sekwencji wideo może być zdefiniowany następująco [3]: Niech dana jest sekwencja wideo

$$S = \{S_1, S_2, \dots, S_T\}$$

obrazów sceny zawierających twarz (twarze) oraz referencyjna baza twarzy

$$B = \{X^{(1,1)}, X^{(1,2)}, \dots, X^{(1,L)}, \dots, X^{(K,1)}, X^{(K,2)}, \dots, X^{(K,L)}\},$$

gdzie  $K$  – liczba znanych klas,  $L$  – liczba wzorców w każdej klasie.

Identyfikacja osób na podstawie obrazów twarzy z sekwencji wideo polega na identyfikacji każdej z osób w scenie  $S_i$ , używając obrazów twarzy wyekstrahowanych z klatek:

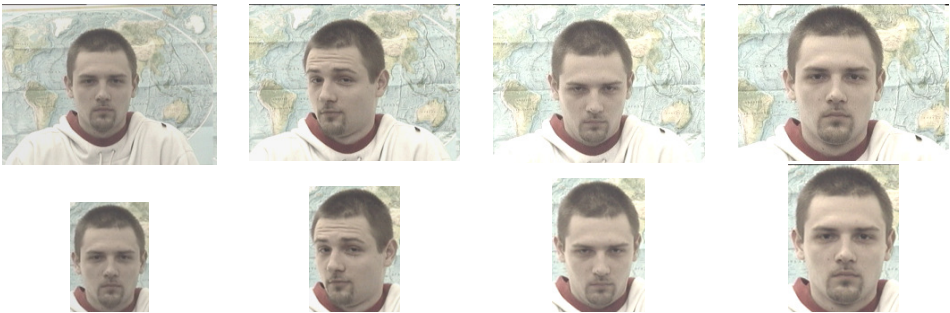
$$S_i, S_{i-1}, S_{i-2}, \dots, S_{i-j},$$

gdzie  $j$  oznacza liczbę poprzedzających klatek sekwencji branych pod uwagę. W tym przypadku konieczne jest zebranie kilku obrazów twarzy konkretnej osoby i wykonanie na ich podstawie procesu identyfikacji.

W poprzedzającym punkcie pokazano, że możliwe jest rozwiązanie zadania rozpoznawania twarzy w warunkach rzeczywistych, nie komplikując algorytmów, a jedynie instalując we właściwych miejscach odpowiednie kamery. W ten sposób wpływ negatywnych czynników, określonych mianem środowiskowych (kąąt obserwacji, oświetlenie, rozmiar twarzy), został w pewien sposób zmniejszony. Skuteczność działania takiego systemu jest przez to wyższa.

Pomimo powyższego należy pamiętać, że twarz jest „obiektem” elastycznym. Użytkownik nie będzie patrzył cały czas na wprost w kamerę. Do tego dochodzi mimika twarzy.

Dalsze zwiększenie skuteczności działania systemu jest możliwe dzięki wykorzystaniu cech sekwencji wideo. Bieżąca i poprzedzające klatki mogą zostać bowiem użyte do zebrania informacji o konkretnej osobie w czasie. W ten sposób dostępnych jest więcej informacji o identyfikowanym człowieku. Zapewnia to bardziej wiarygodną identyfikację, ponieważ badana jest nie tylko pojedyncza próbka, ale ich zbiór. Na rys. 2 został przedstawiony przykład. W górnym wierszu znajdują się 4 klatki z pewnej sekwencji wideo (widoczna jest cała scena). Osoba podchodzi do kamery i jej twarz staje się większa. W dolnym wierszu znajdują się obrazy twarzy wydzielone z górnych obrazów scen. Różnice występują w rozmiarze, oświetleniu, pozie i w tle.



Rysunek 2. Przykład śledzenia i wydzielania obszaru twarzy z tła

Druga zaleta wynika z faktu, że znając lokalizację twarzy w klatce  $S_i$ , możliwe jest ograniczenie przestrzeni poszukiwania twarzy w nowej klatce  $S_{i+1}$  [5]. Informacja o ruchu może być wykorzystana w detektorze twarzy, dostarczając istotny

czynnik pozwalający określić zmiany w scenie. Ma to duże znaczenie w przypadku występowania złożonego tła [3, 6].

System rozpoznawania twarzy oparty na podstawie bazowej struktury [23] przy wykorzystaniu sekwencji wideo napotyka pewne problemy. Identyfikacja oparta jest wówczas na pojedynczych obrazach (kolejnych klatkach sekwencji) – jest mniej wiarygodna i jednocześnie podatna na powstawanie błędów. Jeżeli osoba jest rozpoznawana (identyfikowana) z podobieństwem na pograniczu progu odrzucania lub badany obraz twarzy podobny jest do dwóch różnych osób zarejestrowanych w bazie twarzy systemu, wówczas w kolejnych klatkach sekwencji dana osoba może być raz rozpoznana raz nie lub raz podobna do jednej osoby raz do drugiej. System będzie funkcjonował niestabilnie, generując w kolejnych klatkach sekwencji sprzeczne rezultaty na wyjściu. Efekt ten określony został mianem „migotania” (ang. *flickering*) [3].

Wymienione cechy sekwencji wideo pozwalają na określenie strategii stabilizacji i zwiększenia wiarygodności wyników. System wykorzystujący sekwencje wideo do identyfikacji osób powinien charakteryzować się bardziej stabilnym i wiarygodnym wynikiem w porównaniu z rezultatami uzyskiwanymi w systemie zbudowanym wyłącznie w oparciu o bazową strukturę.

## 5. Śledzenie za twarzami i agregacji wyników

W literaturze tematu [7] wymienione zostały dwie grupy podejść stosowanych ogólnie do śledzenia poruszających się obiektów:

- wykorzystujące ruch (ang. *motion based*),
- wykorzystujące model (ang. *model based*).

Metody z pierwszej grupy są szybkie, ale nie gwarantują, że śledzony obszar ma jakiegokolwiek semantyczne znaczenie. Metody w drugiej grupie charakteryzują się dużą semantyczną wiedzą o śledzonym obiekcie, ale są złożone obliczeniowo. Występujące problemy związane są przede wszystkim ze zmianą skali, translacją, obrotami i deformacjami śledzonych obiektów.

Z kolei do śledzenia twarzy najczęściej używane są [7]:

- barwa,
- cechy twarzy,
- wzorce,
- kontury.

Powyższe mechanizmy lub ich kombinacje wykorzystywane są do szacowania (przewidywania) położenia i ewentualnie orientacji twarzy w sekwencji wideo w czasie rzeczywistym [6].

W pracy wysuwa się odmienną od istniejących propozycję. W każdej klatce sekwencji  $S_i$  dokonuje się detekcji twarzy. Niech wykryte twarze tworzą zbiór  $\Omega^i = \{F_1^i, F_2^i, \dots, F_{J_i}^i\}$ , gdzie  $F_j^i$  oznacza kolejny obraz twarzy  $j$  ( $j = 1, 2, \dots, J_i$ ), wykrytej w klatce o numerze  $i$  ( $J_i$  – liczba wszystkich twarzy wykrytych w klatce  $i$ ). Dla wykrytych twarzy, w scenie z klatki  $S_i$ , dokonuje się próby odszukania

odpowiedników spośród twarzy wykrytych w klatce  $S_{i-1}$ , czyli ze zbioru  $\Omega^{i-1}$ . Pojedynczy zestaw dopasowanych odpowiedników z kolejnych klatek tworzy więc zbiór obrazów twarzy śledzonej osoby, który użyty zostanie w procesie rozpoznawania (identyfikacji). Zbiór takich twarzy będzie w dalszej części nazywany „listą trackingu twarzy”. W każdym kroku, dla każdej wykrytej twarzy (nowej lub śledzonej) przeprowadzany jest proces identyfikacji na podstawie pojedynczego obrazu twarzy. Identyfikator (ewentualnie informacja o porażce identyfikacji) oraz wektor cech reprezentujący twarz są w każdym kroku zapisywane w przechowywanej tymczasowo liście trackingu. Wyjście systemu jest generowane na podstawie danych z kilku klatek, czyli na podstawie danych z istniejących list trackingu. Dla każdej listy obowiązuje mechanizm głosowania większościowego.

Proces znajdowania odpowiedników (faktycznego tworzenia i aktualizacji list trackingu) przebiega w następujący sposób. Niech dany jest obraz sceny, reprezentowany przez macierz poziomów szarości  $S_i$  o wymiarach  $M \times N$ , zawierający twarze osób  $F_j^i$ . Twarz  $F_j^i$ , ograniczona współrzędnymi  $y_1 \leq y \leq y_2$  oraz  $x_1 \leq x \leq x_2$  (w płaszczyźnie obrazu), posiada centrum w punkcie  $(y_c, x_c)$ , gdzie  $y_c = 0,5(y_1 + y_2)$  i  $x_c = 0,5(x_1 + x_2)$ . Dla twarzy  $F_a^i$  poszukiwany jest odpowiednik  $F_b^{i-1}$  wykryty we wcześniejszej klatce  $S_{i-1}$ . Współrzędne centrum  $F_b^{i-1}$  nie mogą jednak przekroczyć granicy obszaru twarzy  $F_a^i$  w  $S_i$ . Oznacza to, że dla twarzy wykrytej w klatce  $S_i$ , wyszukuje się poprzedniego położenia w klatce  $S_{i-1}$ , tylko w bliskim sąsiedztwie (wśród bliskich kandydatów, twarzy z poprzedniej klatki). Wyjątek stanowi sytuacja, gdy w poprzedniej klatce jest tylko jedna twarz. Jeżeli zatem w scenie znajduje się kilka osób (twarzy) poszukiwania ograniczone są tylko do najbliższych. Spośród nich wybierana jest najbardziej podobna. Przy czym, podobieństwo określane jest przy aktualnie stosowanym w systemie sposobie ekstrakcji cech, metryce oraz wartości progowej. Takie założenie jest słuszne, nawet w przypadku ustalenia rygorystycznych parametrów. Podobieństwo twarzy w sąsiednich klatkach jest bowiem bardzo wysokie [3]. Dodatkowo rozważane są następujące sytuacje:

1. Jeżeli dla wykrytej twarzy nie ma w jej sąsiedztwie, w poprzedniej klatce odpowiednika, utworzona jest nowa lista trackingu, do której dodaje się informacje o nowej twarzy.
2. Dla twarzy, która ma odpowiednik w poprzedniej klatce, odszukiwana jest odpowiednia lista trackingu, do której dodaje się informacje o nowej twarzy.
3. Jeżeli przeanalizowano wszystkie wykryte twarze w nowej klatce, a są listy trackingu, które nie zostały zaktualizowane, wówczas z nowej klatki wycina się obszar odpowiadający pod względem lokalizacji i wielkości obszarowi twarzy znajdującej się na końcu danej listy trackingu; taki obszar sprawdzany jest pod względem podobieństwa z ostatnią twarzą na liście (z poprzedniego kroku) i w przypadku stwierdzenia podobieństwa traktowany jest jak twarz (wykonywana jest procedura identyfikacji i dopisanie do listy trackingu), w przeciwnym wypadku lista trackingu jest usuwana.

Wykorzystywane w algorytmie listy trackingu pełnią w systemie bardzo ważną rolę – agregacji wyników indywidualnych identyfikacji. Dla śledzonej twarzy,

w każdej klatce formułowany jest końcowy wynik, na bazie indywidualnych wyników identyfikacji z bieżącej i poprzednich klatek. Wykorzystywany jest mechanizm głosowania. Przyjęto, że przynajmniej w połowie identyfikacji powinna zostać wskazana ta sama klasa. Z listą trackingu związany jest jeszcze problem jej długości. Ta kwestia zostanie poruszona w dalszej części.

## 6. Eksperymenty

W celu weryfikacji przyjętych założeń i zaproponowanego rozwiązania wykonane zostaną eksperymenty. W badaniach wykorzystane zostaną dwie bazy: ORL [8] i VIS\_DB [9]. Obie bazy dostępne są publicznie.

Pierwsza baza ze względu na dostępność jest szeroko stosowana w badaniach poświęconych rozpoznawaniu twarzy. Pomimo jej wieloletniego użytkowania zainteresowanie tą bazą nie słabnie i jest ona chętnie wykorzystywane w badaniach nad najnowszymi mechanizmami rozpoznawania twarzy (np.: [10, 11, 12]. W bazie ORL znajduje się 400 obrazów twarzy mężczyzn i kobiet podzielonych na 40 klas. W każdej klasie znajduje się po 10 obrazów twarzy.

Baza osób VIS\_DB zawiera z kolei 1350 obrazów całych scen podzielonych na 27 klas, z których każda zawiera 50 obrazów. Obrazy znajdujące się w każdej klasie to kolejne klatki sekwencji wideo rejestrowane z losowym interwałem czasowym równym 0,1-1,0 s. Baza ta dostępna jest publicznie na stronie internetowej Zakładu Przetwarzania i Rozpoznawania Obrazów Politechniki Szczecińskiej [9].

W testowanych systemach rozpoznawania twarzy w roli ekstraktora cech wykorzystana zostanie dwuwymiarowa transformata kosinusowa [13, 14, 15]. Klasyfikator oparty będzie o odległość Euklidesa. Wybór niskoczęstotliwościowych współczynników transformaty kosinusowej do wektora cech zostanie wykonany z wykorzystaniem metody trójkąta (przekątnych) [13, 14]. Uwzględnione zostaną dwa warianty wyboru przekątnych: 20 i 11. Dla każdego z tych wariantów sprawdzone będą wersje z zerowaniem (oznaczone jako: DIAG20\_0 i DIAG11\_0) i bez zerowania (oznaczone jako: DIAG20 i DIAG11) współczynnika transformaty o współrzędnych (0, 0). Współczynnik ten ma wartość wyższą od pozostałych, uzależniony jest silnie od intensywności obrazu, przez co ma duży wpływ na rozpoznawanie [13].

### 6.1. Eksperyment 1

W celu sprawdzenia, czy mechanizm śledzenia i dostępna dzięki niemu większa liczba próbek twarzy na osobę ma wpływ na wyniki rozpoznawania i czy rzeczywiście, jak zostało założone, wyniki te są bardziej stabilne i wiarygodne, zostanie przeprowadzony eksperyment na bazie ORL, podzielonej według wariantu 5/5 (5 obrazów z każdej klasy użyte jest do syntezy systemu, pozostałe 5 – do testów). Dokonana zostanie identyfikacja osób przedstawionych na obrazach testowych w bazowym i rozbudowanym systemie. Parametry pierwszego systemu to: ekstraktor cech – DCT (wersje DIAG20, DIAG20\_0, DIAG11, DIAG11\_0), klasyfikator – bazujący na odległości Euklidesa. W drugim systemie wykorzystane



zostaną aspekty wynikające ze śledzenia twarzy i akumulacji wyników. Pięć obrazów testowych przypadających na osobę zostanie podanych na wejście systemu. Dla każdego z tych obrazów wykonana zostanie procedura identyfikacji. Na podstawie uzyskanych pięciu indywidualnych wyników zostanie wygenerowany końcowy rezultat metodą głosowania większościowego. Oznacza to, że przynajmniej 3 z 5 (3/5) indywidualnych rezultatów musi wskazywać na tę samą klasę (osobę). Jeżeli tak jest, sprawdzone zostanie, czy wskazywana klasa rzeczywiście odpowiada osobie, której sekwencja 5 obrazów została zadana na wejście. Wynik rozpoznawania zostanie obliczony jako stosunek prawidłowo zidentyfikowanych klas do ich całkowitej liczby.

Rezultaty przeprowadzonego eksperymentu przedstawione zostały w tab. 1. System bazowy zrealizowany został w oparciu o bazową strukturę. W systemie rozbudowanym agregowane są wyniki z wcześniejszych identyfikacji.

Tabela 1. Wyniki identyfikacji w bazowym i rozbudowanym systemie (reguła 3/5)

System	DIAG20	DIAG20_0	DIAG11	DIAG11_0
Bazowy	91,50%	89,00%	92,50%	91,50%
Rozbudowany	97,50%	97,50%	97,50%	97,50%
Wzrost skuteczności	6,00%	8,50%	5,00%	6,00%

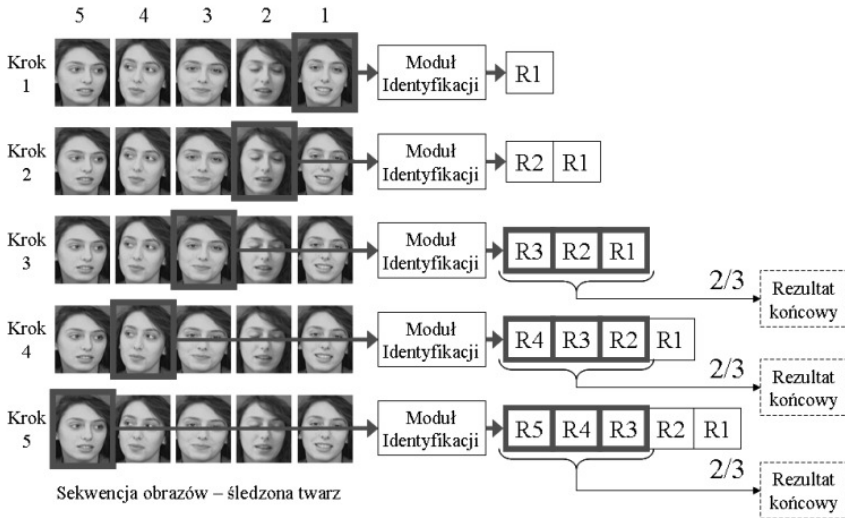
Na podstawie uzyskanych rezultatów widać, że mechanizm uwzględniający zestaw obrazów testowych jest bardziej skuteczny w porównaniu z mechanizmem uwzględniającym jedynie indywidualne obrazy. Wyniki pokazują również bardzo ważną cechę: nie ma znaczenia wybór parametrów w stosowanych metodach ekstrakcji cech. Oznacza to, że nawet przy gorszym ekstraktorze cech, dzięki wykorzystaniu sekwencji wideo i pochodzących z niej sekwencji obrazów twarzy uzyskać można bardzo dobre i stabilne rezultaty.

## 6.2. Eksperyment 2

Wykonany teraz zostanie eksperyment, który w większym stopniu przypomina warunki rzeczywiste. Podział danych i bazowa (porównawcza) struktura pozostaną identyczne jak przy pierwszym eksperymencie. Zmianie ulegnie sposób generowania wyników w proponowanym systemie z agregacją. Aby zasymulować pojawianie się nowych obrazów na wejściu systemu, do identyfikacji użyte zostaną tylko trzy ostatnie obrazy danej osoby oraz większościowy mechanizm wnioskowania – czyli 2 z 3.

Idea tego eksperymentu przedstawiona została obrazowo na rys. 3. Moduł identyfikacji na wejście otrzymuje pojedynczy (kolejny) obraz twarzy. Na wyjściu zwraca wynik identyfikacji – symbolicznie oznaczony symbolem „R” i numerem kroku. W skład modułu identyfikacji wchodzi: ekstraktor cech, klasyfikator i referencyjna baza twarzy. W pierwszym kroku na wejściu pojawia się więc obraz z numerem 1 i generowany jest wynik R1. Kolejne obrazy testowe i kolejne wyniki pojawiają się

w następnych krokach. Poczynając od kroku trzeciego dostępne są trzy indywidualne wyniki (z aktualnego kroku i dwóch poprzedzających). Jeżeli przynajmniej dwa z tych trzech wyników (reguła 2/3) wskazują na tę samą klasę to klasa ta stanowi wyjście systemu. Dla każdej klasy są więc trzy zbiorcze rezultaty (z kroku 3, 4 i 5). Współczynnik rozpoznawania zostaje obliczony jako stosunek wszystkich prawidłowych zbiorczych rezultatów (określonych na rysunku mianem „rezultatów końcowych”) do ich całkowitej liczby. Porównawcze wyniki przedstawione zostały w tab. 2.



Rysunek 3. Idea agregacji wyników i mechanizmu głosowania większościowego

Tabela 2. Wyniki identyfikacji w bazowym i rozbudowanym systemie (reguła 2/3)

System	DIAG20	DIAG20_0	DIAG11	DIAG11_0
<b>Bazowy</b>	91,50%	89,00%	92,50%	91,50%
<b>Rozbudowany</b>	96,67%	96,67%	96,67%	97,50%
<b>Wzrost skuteczności</b>	5,17%	7,67%	4,17%	6,00%

Okazuje się, że tak jak w pierwszym eksperymencie, tak i tutaj uzyskano znacznie lepsze rezultaty. Wzrost skuteczności wyniósł od 4,17% do 7,67% w zależności od parametrów systemu. Na podstawie danych z tab. 2 widać, że rezultaty otrzymywane przez system agregujący wyniki są stabilne i bardzo mało zależne od parametrów ekstraktora cech. Ponownie więc okazuje się, że identyfikacja opierająca się o zestaw obrazów testowych jest bardziej skuteczna aniżeli opierająca się na jednym tylko obrazie. Porównując wyniki pierwszego eksperymentu z drugim można stwierdzić, że nieco słabsze wyniki w drugim przypadku są wynikiem uwzględnienia

mniej liczby „wcześniejszych” obrazów. Ta obserwacja dodatkowo wskazuje, że zwiększenie liczby „historycznych” obrazów twarzy uwzględnianych w identyfikacji może wpłynąć korzystnie na otrzymywany rezultat. Jednak zbyt duża liczba nie gwarantuje sukcesu, a powodować może dodatkowe błędy. Kiedy bowiem podczas śledzenia za twarzami wystąpi błąd – system błędnie przeniesie śledzenie z jednej twarzy na inną – wyniki identyfikacji mogą zostać przekłamane. W tym kontekście liczba od 5 do 10 klatek wstecz stanowi rozsądną granicę. Doświadczenia wykonane na rzeczywistym systemie, przy pracy z kamerą, potwierdziły te założenia.

### 6.3. Eksperyment 3

Zadaniem trzeciego eksperymentu będzie sprawdzenie skuteczności bazowego systemu oraz rozbudowanego (ze zintegrowanym blokiem śledzenia za twarzami i agregacji wyników) na danych wideo. Wykorzystany zostanie detektor twarzy opisany w [16], ekstraktor cech DCT\_DIAG11 i klasyfikator oparty na metryce Euklidesowej. Niech śledzenie wynosi 3, 5 i 7 klatek. Do badań użyta zostanie baza VIS\_DB przy wariancie podziału 10/40 (10 obrazów z każdej klasy użyte jest do syntezy systemu, pozostałe 40 – do testów). Podczas rejestracji prezentowanych jest więc 270 obrazów całych scen, z których system sam tworzy odpowiednią reprezentację twarzy do referencyjnej bazy twarzy. Następnie do testów przedstawiane są sekwencje kolejnych obrazów testowych z bazy VIS\_DB (razem 1080 obrazów testowych, 40 na klasę). W tab. 3 przedstawione zostały zbiorcze wyniki identyfikacji osób w bazowym systemie oraz w systemie rozbudowanym.

Tabela 3. Wyniki identyfikacji na bazie VIS\_DB w bazowym systemie i systemie z blokiem śledzenia za twarzami i agregacji wyników

System bazowy	System z blokiem śledzenia za twarzami		
	3 klatki	5 klatek	7 klatek
84,44%	90,09%	92,04%	92,87%

W proponowanym podejściu wymagana była wartość progowa w procedurze śledzenia twarzy. Wykorzystano wartość 21,46, ustaloną eksperymentalnie dla bazowego systemu (mającego takie same parametry ekstraktora cech i klasyfikatora) na bazie ORL. W eksperymencie pominięto kwestię doboru optymalnej wartości progowej.

Przedstawione wyniki badań na rzeczywistych sekwencjach scen pokazały, że zaproponowany mechanizm śledzenia za twarzami i agregacji wyników z kolejnych klatek sekwencji jest skuteczny. Przy rosnącej liczbie obrazów uwzględnianych przy generowaniu bieżącego wyniku identyfikacji, skuteczność systemu rośnie. Przy założonym interwale czasowym 0,2-1 s poziom 5 klatek jest w pełni uzasadniony.

## 7. Zakończenie

W artykule przedstawiono problem identyfikacji człowieka na podstawie zestawu obrazów twarzy pochodzących z sekwencji wideo. Zaproponowano mecha-

nizm poszukiwania odpowiedników w lokalnym sąsiedztwie z poprzednich klatek sekwencji wideo. Wykazano, że wykorzystujący go algorytm agregacji wyników pozwala zwiększyć skuteczność identyfikacji. Końcowy wynik jest przy tym bardziej wiarygodny i stabilny. W artykule pokazano przy tym, że tradycyjny system monitoringu, który mógłby stanowić źródło sekwencji wideo do systemu rozpoznawania, niekoniecznie spełnia wymagane kryteria jakości obrazów. Zaproponowano więc odpowiednie podejścia do rozwiązania tego problemu.

Przeprowadzone eksperymenty pokazały, że wzrost jakości systemu przejawiający się w bardziej stabilnych i wiarygodnych wynikach jest możliwy dzięki integracji mechanizmów śledzenia za twarzami i agregacji wyników.

## Bibliografia

- [1] International Biometric Group (IBG). *Biometrics Market and Industry Report 2004-2008*. 272 s., [www.biometricgroup.com](http://www.biometricgroup.com), 2004
- [2] International Biometric Group (IBG). *Biometrics Market and Industry Report 2006-2010*. 221 s., [www.biometricgroup.com](http://www.biometricgroup.com), 2006
- [3] Kukharev G., Nowosielski A.. *Face Based Identification of People from a Video Stream*. Computing, Multimedia and Intelligent Techniques. Special issue on Live Biometrics and Security 1(1), 99-110, 2005
- [4] Commission of the European Communities. *Proposal for a Council Regulation on standards for security features and biometrics in EU citizens' passports COM(2004) 116 final*. 21 s., placeCityBrussels, 18.II.2004
- [5] Comaniciu D., Berton F., Ramesh V. *Adaptive Resolution System for Distributed Surveillance*. Real-Time Imaging 8(5), 427-437, 2002
- [6] Yang M., Kriegman D., Ahuja N. *Detecting Faces in Images: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34-58, 2002
- [7] Wang J. J., Singh S. *Video analysis of human dynamics - a survey*. Real-Time Imaging 9(5), 321-346, 2003
- [8] *The ORL face database. Baza twarzy ORL* (także pod nazwami: *Olivetti, AT&T*). <http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html>, 2007
- [9] Strona WWW Zakładu Przetwarzania i Rozpoznawania Obrazów. Wydział Informatyki. Politechnika Szczecińska. <http://zpiro.wi.ps.pl>, 2007
- [10] Howlanda P., Wang J., Park H. *Solving the small sample size problem in face recognition using generalized discriminant analysis*. Pattern Recognition 39(2), 277-287, 2006
- [11] Yu W., Teng X., Liu C.. *Face recognition using discriminant locality preserving projections*. Image and Vision Computing 24(3) 239-248, 2006
- [12] Saradha A., Annadurai S. *A Hybrid Feature Extraction Approach for Face Recognition Systems*. ICGST International Journal on Graphics, Vision and Image Processing 5(5), 23-30, 2005
- [13] Sanderson C., Paliwal K. K. *Fast Features for Face Authentication Under Illumination Direction Changes*. Pattern Recognition Letters 24(14), 2409-2419, 2003
- [14] Nowosielski A. *Ekstrakcja cech twarzy z wykorzystaniem dwuwymiarowej transformaty kosinusowej*. Materiały VIII Sesji Naukowej Informatyki Tom 1, 153-162, Szczecin, 2003
- [15] Sanderson C., Bengio S., Gao Y. *On transforming statistical models for non-frontal face verification*. Pattern Recognition 39(2), 288-302, 2006

- 
- [16] Kukharev G., Nowosielski A. *Fast and efficient algorithm for face detection in colour images*. Machine GRAPHICS & VISION 13(4), 377-399, 2004
  - [17] European Commission's Joint Research Centre (JRC). *Biometrics at the Frontiers: Assessing the Impact on Society EUR 21585 EN*. Technical Report Series. 146 s., II.2005
  - [18] Pentland A., Choudbury T. *Face Recognition for Smart Environments*. IEEE Computer 33(2), 50-55, 2000
  - [19] Bowyer K. W., Chang K., Flynn P. *A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition*. Computer Vision and Image Understanding 101(1), 1-15, 2006
  - [20] Woodward Jr. J. D., Horn Ch., Gatune J., Thomas A. *Biometrics A Look at Facial Recognition*. Prepared for the PlaceNameplaceVirginia PlaceNameState Crime Commission. RAND Public Safety and Justice. 32 s., 2003
  - [21] Belhumeur P. N., Hespanha J. P., Kriegman D. J. *Eigenfaces vs. Fisherfaces: Recognition Using class Specific Linear Projection*. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 711-720, 1997
  - [22] Zhao W., Chellappa R., Phillips P. J., Rosenfeld A. *Face Recognition: A Literature Survey*. ACM Computing Surveys 35(4), 399-458, 2003
  - [23] Kukharev G., Kuźmiński A. *Techniki Biometryczne. Część 1. Metody Rozpoznawania Twarzy*. Pracownia Poligraficzna WI PS, 310 s., Szczecin, 2003



# Poszukiwanie optymalnej strategii eksploracji z zastosowaniem uczenia ze wzmocnieniem

Marcin Pluciński

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*The paper presents an application of the reinforcement learning for a searching of an optimal policy in an exploration problem (also known as a Jeep problem). The continuous problem is unrealistic so the main work was concentrated on the discrete Jeep problem. There is examined and described an influence of main learning parameters on the learning speed and there are presented some found exemplary policies for different problem conditions.*

## Słowa kluczowe:

*reinforcement learning, exploration problem, Jeep problem*

## 1. Problem eksploracji

Założmy, że chcemy przebyć pustynię samochodem, mając do dyspozycji  $n$  jednostek paliwa. Założmy dalej, że zbiornik samochodu mieści tylko 1 jednostkę paliwa i że pozwala ona na przejechanie 1 jednostki odległości. Samochód może pozostawiać na trasie swojego ruchu dowolne ilości paliwa w celu późniejszego wykorzystania. Jaka powinna być optymalna strategia ruchu samochodu i jaką maksymalną odległość można pokonać, przy tak postawionych warunkach?

Opisany problem jest ważnym i ciekawym zadaniem logistycznym, mającym przykładowo znaczenie dla planowania ekspedycji badawczych (np. arktycznych) czy operacji wojskowych. W literaturze znany jest jako tzw. problem Jeepa (ang. *Jeep problem*) lub problem eksploracji.

Po raz pierwszy problem sformułowany został ponad 1000 lat temu w „*Propositiones ad acuendos iuvenes*” – jednym z najstarszych zbiorów łamigłówek matematycznych w języku łacińskim [1, 13]. We współczesnej literaturze matematycznej problem opisał i podał jego dokładne rozwiązanie N. Fine [4] w 1947 r., a od tego czasu przedstawiano go w wielu różnych wersjach, w których poszukiwano rozwiązania dla wielu pojazdów [5, 11, 17] oraz analizowano różne warianty dyskretnych wersji problemu [2, 6, 13]. W jeszcze innych pracach, badano stochastyczną wersję zadania [7] oraz jego rozszerzenie na 2 wymiary [8].

### 1.1. Zadanie ciągłe

Zadanie ciągłe, w którym w pojedynczym kroku samochód może pokonywać dowolne odległości i pozostawiać dowolne ilości paliwa, może być rozwiązane poprzez zastosowanie strategii rekurencyjnej [4, 10, 17]. W przypadku gdy ilość paliwa  $n = 0$ , samochód nie może się ruszyć, stąd oczywiście pokonana odległość to  $d_0 = 0$ . Dla  $n = 1$ , jedyna rozsądna strategia to zatankowanie całości paliwa i przejechanie jednej jednostki odległości:  $d_1 = 1$ . Jednak dla  $n = 2$ , samochód może zatankować 1 jednostkę paliwa, pokonać  $1/3$  jednostki odległości, zostawić  $1/3$  jednostki paliwa i wrócić do punktu początkowego. Tam może zatankować pozostałą 1 jednostkę paliwa, przejechać  $1/3$  odległości, dotankować  $1/3$  jednostki paliwa zostawioną wcześniej i dalej przejechać 1 jednostkę odległości. Maksymalna pokonana w ten sposób odległość to  $d_2 = 1 + 1/3$  jednostki odległości. Zrealizowaną strategię symbolicznie będziemy zapisywać w postaci [10]:

$$(\uparrow 1), (\rightarrow \frac{1}{3}), (\downarrow -\frac{1}{3}), (\leftarrow 0), (\uparrow 1), (\rightarrow \frac{1}{3}), (\uparrow \frac{1}{3}), (\rightarrow 1\frac{1}{3}),$$

lub:

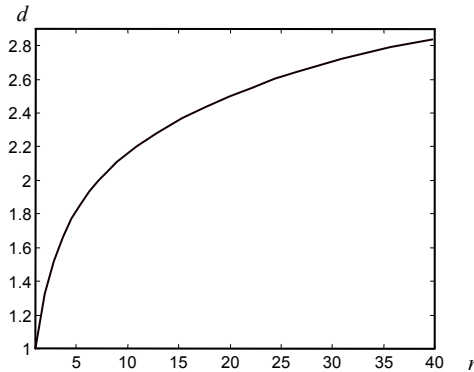
$$(T1), (R\frac{1}{3}), (T - \frac{1}{3}), (R0), (T1), (R\frac{1}{3}), (T\frac{1}{3}), (R1\frac{1}{3}),$$

gdzie: R – oznacza ruch do określonej pozycji, a T – tankowanie (gdy wartość jest dodatnia) lub pozostawianie paliwa (gdy wartość jest ujemna).

Można wykazać [4, 17], że maksymalna odległość jaką samochód jest w stanie pokonać w ten sposób wynosi:

$$d_n = \sum_{k=1}^n \frac{1}{2k-1}. \quad (1)$$

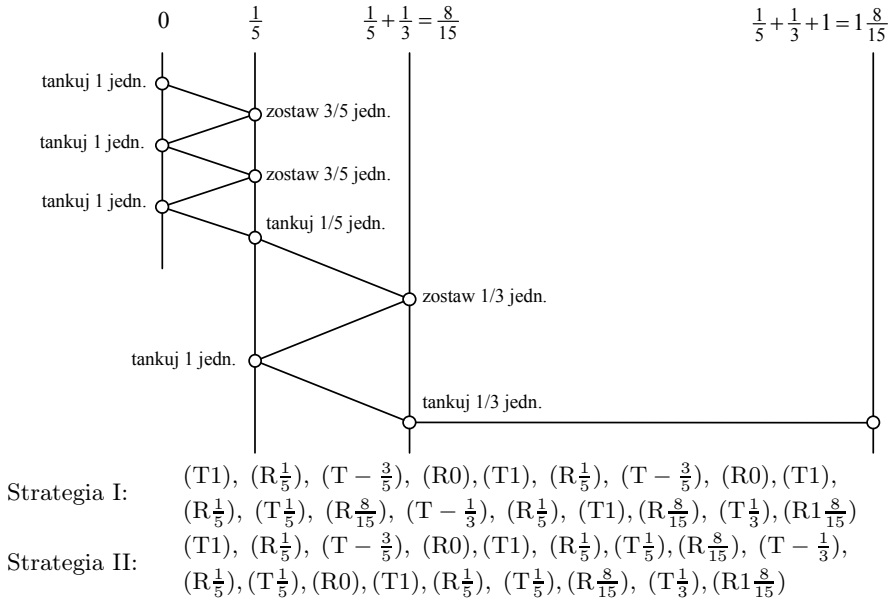
Dla  $n \rightarrow \infty$ , odległość również będzie dążyła do  $\infty$ , przy czym ilość możliwych optymalnych strategii będzie również nieskończenie duża. Fakt istnienia większej ilości optymalnych rozwiązań (dla  $n > 2$ ) dodatkowo komplikuje zadanie. Rys. 1 przedstawia wykres możliwej do pokonania drogi w funkcji ilości paliwa  $n$ .



Rysunek 1. Maksymalna możliwa do pokonania droga w funkcji ilości paliwa



Na rys. 2 pokazano przykładowe strategie dla  $n = 3$ .



Rysunek 2. Dwie strategie optymalne dla  $n = 3$ . Strategia pierwsza zilustrowana jest graficznie.

### 1.2. Zadanie dyskretne

Przedstawiony powyżej problem może stać się dużo trudniejszy do rozwiązania przy zmianie warunków zadania. Jest on ponadto nierealistyczny, stąd dalej zajmijemy się problemem dyskretnym.

Założmy, że samochód ma do dyspozycji  $n$  jednostek paliwa (gdzie  $n$  to liczba naturalna). Naszym zadaniem jest odnalezienie strategii poruszania się samochodu, tak aby pokonać jak największą odległość przy założeniu, że podczas jednej akcji samochód może poruszać się z dokładnością do kroku  $k$  i taka sama wartość określa minimalną ilość paliwa, jaką samochód może zostawić lub zabrać. Pojemność zbiornika w samochodzie wynosi  $z \cdot k$ , a  $z$  i  $k$  to oczywiście również liczby naturalne.

Ciekawe rozwiązania dla problemu dyskretnego, dla określonych z góry warunków zadania, pokazują prace [2, 13]. Zmiana tych warunków jednakże, wymusiłaby poszukiwanie rozwiązania całkowicie na nowo.

Brak jest w literaturze jednolitej metody określania rozwiązania dla dowolnych warunków problemu eksploracji. Pewną próbę odnalezienia takiej metody można znaleźć w pracy [10]. Autor wykorzystał w niej algorytm ewolucyjny do poszukiwania optymalnej strategii, jednak zadanie udało mu się rozwiązać jedynie dla  $n = 2$  i w ograniczonym zakresie dla  $n = 3$ .

W dalszej części artykułu zaprezentowane zostanie zastosowanie uczenia ze wzmocnieniem do poszukiwania optymalnej strategii ruchu. Metoda ta może być zastosowana dla dowolnie postawionych warunków zadania eksploracji, z tym że założymy, że mamy z góry zadaną odległość jaką chcemy pokonać, a naszym celem będzie odnalezienie strategii jej pokonania.

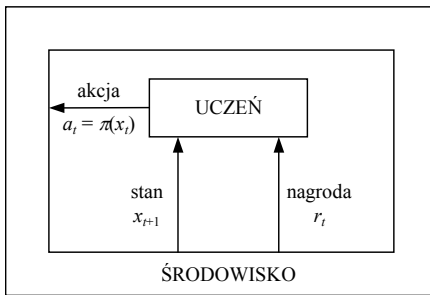
## 2. Uczenie ze wzmocnieniem

Najogólniej możemy napisać, że celem uczenia ze wzmocnieniem jest odnalezienie optymalnej strategii rozwiązywania określonego zadania w nieznanym środowisku. Uczeń może obserwować stan środowiska i na jego podstawie wybierać do wykonania akcję zgodną z jego aktualną strategią. Po każdej akcji uczeń obserwuje stan następny oraz otrzymuje od środowiska tzw. wzmocnienie (nagrodę), będące miarą jakości wykonanej akcji. Na podstawie uzyskanych obserwacji uczeń stara się tak modyfikować swoją strategię, aby wartości wzmocnienia uzyskiwane w przyszłości były jak największe [9, 15].

Strategia powinna zależeć jedynie od stanu, stąd podczas uczenia poszukujemy optymalnego odwzorowania [3]:

$$\pi : X \rightarrow A, \quad (2)$$

gdzie:  $\pi$  – to strategia,  $X$  – zbiór stanów środowiska i  $A$  – zbiór możliwych do wykonania akcji. Schematycznie przebieg uczenia ze wzmocnieniem przedstawiony jest na rys. 3.



dla każdego kroku  $t$   
 obserwuj stan środowiska  $x_t$   
 wybierz akcję  $a_t$  w stanie  $x_t$   
 (zgodnie z bieżącą strategią)  
 wykonaj akcję  $a_t$   
 obserwuj wzmocnienie  $r_t$   
 obserwuj następny stan  $x_{t+1}$   
 na bazie doświadczenia  $\langle x_t, a_t, r_t, x_{t+1} \rangle$   
 popraw strategię  
 koniec

Rysunek 3. Schemat procesu uczenia ze wzmocnieniem [3, 12]

Dla aktualnej strategii ucznia  $\pi$ , możemy zdefiniować tzw. funkcję wartości, która jest odwzorowaniem stanu  $x$  na całkowitą ilość nagród jakich uczeń spodziewa się uzyskać w przyszłości po rozpoczęciu realizacji strategii z tego stanu [3]:

$$V^\pi(x) = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t \cdot r_t \mid x_0 = x \right] \quad (3)$$

gdzie:  $E_\pi$  – oczekiwana (dla aktualnej strategii  $\pi$ ) suma przyszłych nagród  $r_t$ ,  $\gamma \in (0, 1]$  – współczynnik dyskontowania, który określa istotność przyszłych nagród dla wartości stanu.

Podczas uczenia poszukujemy optymalnej strategii  $\pi^*$  – strategii, dla której zawsze otrzymujemy największe wzmocnienia od środowiska. Dla takiej strategii, funkcja wartości  $V^{\pi^*}(x)$  jest dla każdego stanu  $x$  większa lub równa funkcji  $V^\pi(x)$  dla każdej innej strategii  $\pi$ . Jeżeli znamy optymalną funkcję wartości  $V^{\pi^*}(x)$ , możemy łatwo określić optymalną strategię  $\pi^*$  jako zachłanną wobec niej. Strategia zachłanna do funkcji  $V(x)$  wybiera zawsze akcję, która maksymalizuje wartość sumy natychmiastowej nagrody i zdyskontowanej wartości następnego stanu [3, 12].

## 2.1. Algorytm TD

Jednym z najistotniejszych algorytmów uczenia ze wzmocnieniem jest algorytm różnic czasowych TD (ang. *temporal difference*) [14, 16]. W algorytmie tym poszukujemy optymalnej funkcji wartości  $V^{\pi^*}(x)$ .

Początkową postać funkcji  $V(x)$  doбира się arbitralnie. Podczas kolejnych kroków uczenia funkcja jest modyfikowana na podstawie obserwacji  $\langle x_t, a_t, r_t, x_{t+1} \rangle$  w następujący sposób:

$$V_{t+1}(x_t) = V_t(x_t) + \eta \cdot \Delta, \quad (4)$$

gdzie:  $\eta$  – współczynnik szybkości uczenia i:

$$\Delta = r_t + \gamma \cdot V_t(x_{t+1}) - V_t(x_t). \quad (5)$$

$V_t(x_t)$  – to wartość funkcji dla bieżącego stanu,  $r_t + \gamma \cdot V_t(x_{t+1})$  to suma nagrody otrzymanej po akcji w kroku  $t$  i zdyskontowana wartość funkcji dla następnego stanu – jest to prawdopodobnie lepsze oszacowanie  $V_t(x_t)$  dla bieżącej strategii  $\pi$  [3, 11].

Jeśli akcje będą zawsze wybierane tak, aby następna wartość sumy nagrody i funkcji  $V^\pi(x)$  była maksymalna, wówczas nasza strategia będzie zachłanna do funkcji  $V^\pi(x)$  i można wykazać, że będzie lepsza lub równa strategii  $\pi$ . Na bazie równań (4) i (5) można zatem zapisać pełną metodę adaptacji funkcji  $V^\pi(x)$ :

$$V_{t+1}(x_t) = V_t(x_t) + \eta \cdot [\max_a (r_t + \gamma \cdot V_t(x_{t+1})) - V_t(x_t)]. \quad (6)$$

Operator  $\max_a$  oznacza wybór akcji w sposób zachłanny. Iteracyjne poprawianie funkcji wartości zgodnie z równaniem (6), będzie prowadziło do zbieżności do optymalnej funkcji wartości  $V^{\pi^*}(x)$ , dla której optymalna strategia może być określona jako zachłanna.

Aby zapewnić lepsze przeszukiwanie przestrzeni możliwych rozwiązań (strategii), w praktyce zależność (6) nie jest wykorzystywana do poprawiania  $V(x)$  przez cały czas, a jedynie przez jego większość. Z prawdopodobieństwem  $\varepsilon$ , w każdym kroku pracy algorytmu akcja może być wykonana losowo (po takiej akcji oczywiście nie modyfikujemy funkcji wartości), stąd będziemy mówili o strategii  $\varepsilon$ -zachłannej, która będzie kompromisem pomiędzy lepszym przeszukiwaniem przestrzeni możliwych strategii, a szybką zbieżnością do strategii optymalnej [12].

W rozwiązaniach praktycznych zależność (6) uzupełnia się jeszcze o tzw. ślad aktywności [3, 15]:

$$V_{t+1}(x_t) = V_t(x_t) + \eta \cdot [\max_a (r_t + \gamma \cdot V_t(x_{t+1})) - V_t(x_t)] \cdot e_t(x). \quad (7)$$

gdzie:

$$e_t(x) = \sum_{k=0}^t (\gamma \cdot \lambda)^{t-k} \cdot \chi_x(t),$$

$$\chi_x(t) = \begin{cases} 1 & \text{dla } x = x_t \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

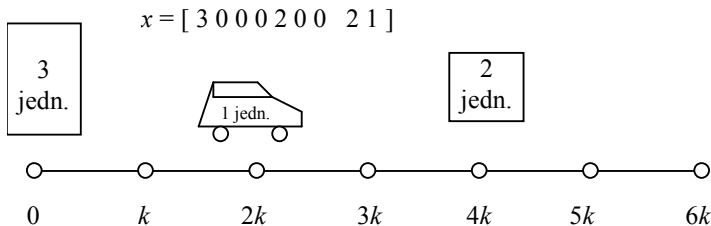
Parametr  $\lambda \in [0, 1]$  to tzw. współczynnik świeżości. W zależności (6)  $\lambda$  jest równe 0, jednak użycie  $\lambda > 0$  w wielu przypadkach przyspiesza znacznie proces uczenia, mimo większego nakładu obliczeń (konieczność obliczania  $e_t(x)$  oraz modyfikacji funkcji  $V(x)$  dla wszystkich stanów  $x$ ).

### 3. Poszukiwanie optymalnej strategii eksploracji

Jednym z najważniejszych zadań, przy formułowaniu ostatecznej wersji algorytmu jest właściwe zdefiniowanie stanu. Stan środowiska powinien cechować się własnością Markowa, tak aby decyzja o podejmowanej akcji zależała jedynie od jego wartości.

Rozpatrzmy dalej zadanie dyskretne opisane w podrozdziale 1.2. Zakładamy, że mamy do dyspozycji  $n$  jednostek paliwa. Położenie samochodu oraz stan zatkanowania mogą być zmieniane co najmniej o krok  $k$ , a pojemność zbiornika to  $z \cdot k$ . Chcemy znaleźć strategię, która umożliwi pokonanie odległości  $d$ , przy założeniu, że jedna jednostka paliwa wystarcza na pokonanie jednej jednostki odległości ( $n$ ,  $k$ ,  $z$ ,  $d$  – to liczby naturalne).

Do podjęcia decyzji o rodzaju akcji, jaka ma być wykonana w danym momencie, musimy znać ilość paliwa w punkcie startowym i wszystkich punktach pośrednich oraz oczywiście pozycję samochodu i zawartość jego zbiornika. Przykładową reprezentację stanu pokazuje rys. 4.



Rysunek 4. Reprezentacja stanu środowiska

Jeżeli założymy równomierny rozkład punktów, w których można zostawiać paliwo – co krok  $k$ , to pierwszych  $m/k + 1$  liczb w wektorze stanu będzie określało

dostępne na trasie ilości paliwa, a dwie ostatnie to odpowiednio położenie samochodu i stan jego zbiornika.  $m$  – to liczba naturalna, określająca maksymalny analizowany zasięg samochodu ( $m > d$ ). Tak zdefiniowany wektor stanu  $x$  jest wystarczający do podjęcia decyzji o tym, jaką akcję wykonać.

W praktyce akcje związane z ruchem samochodu i zmianą stanu zatankowania są wykonywane na przemian. Ruch samochodu rozpoczyna się będzie zawsze w punkcie 0 z zatankowanym do pełna zbiornikiem. Początkowa wartość funkcji  $V(x)$  dla każdego stanu została przyjęta jako 0.5. Akcje wybierane są zgodnie ze strategią  $\varepsilon$ -zachłanną (na przemian: ruch i tankowanie/zostawianie paliwa) spośród wszystkich możliwych, przy uwzględnieniu aktualnego położenia i stanu zbiornika samochodu. Po wykonaniu każdej akcji następuje przejście do kolejnego stanu, po czym przyznawane jest wzmocnienie. Wartość wzmocnienia wynosi 0 gdy pozycja samochodu  $p$  jest mniejsza od zadanej do przejechania odległości  $d$  lub 1 w przeciwnym przypadku. Na podstawie wybranej zachłannej akcji, po przejściu do nowego stanu i otrzymaniu nagrody – dokonywane są zmiany funkcji wartości  $V(x)$  zgodnie ze wzorem (7).

Kolejnym istotnym dla przebiegu procesu uczenia czynnikiem jest sposób reprezentacji funkcji wartości  $V(x)$ . W opisanych w dalszej części badaniach zastosowano reprezentację tablicową, w której oprócz stanu przechowywana była jego wartość  $V(x)$  oraz ślad aktywności  $e_t(x)$ .

### 3.1. Wyniki eksperymentów

W pierwszym etapie badań określono jaki jest wpływ parametrów metody na jej szybkość działania. W każdym eksperymencie zliczano liczbę epok potrzebnych do nauczenia się prawidłowej strategii pokonywania zadanej odległości. Przez epokę rozumiemy tu pełny cykl uczenia od wyruszenia z punktu startowego, aż do momentu, gdy dalszy ruch samochodem nie był możliwy z powodu braku paliwa.

Przyjęto: zadana droga  $d = 5$ , krok  $k = 1$ , pojemność zbiornika  $z = 3$ , współczynnik szybkości uczenia  $\eta = 0.2$ .

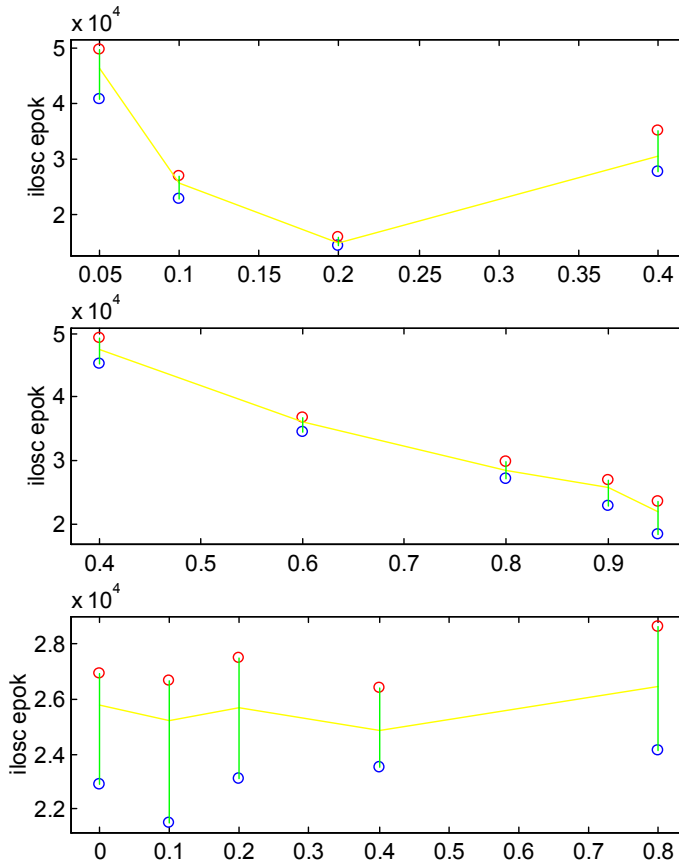
Na rys. 5 przedstawione są wyniki eksperymentów. Wykresy przedstawiają czas nauki (w epokach) w zależności od parametrów  $\varepsilon$ ,  $\gamma$  i  $\lambda$ . Czas trwania jednej epoki wynosił około 1ms<sup>8</sup>.

Wykres na rys. 5a obrazuje wpływ współczynnika  $\varepsilon$  ( $\gamma = 0.9$  i  $\lambda = 0$ ). Dla każdej wartości  $\varepsilon$  przeprowadzono 10 prób. Linia ciągła łączy średnie wartości uzyskane w badaniach, a dodatkowo zaznaczono na wykresie przedziały zmienności wyników. Najszybciej uczenie przebiegało dla  $\varepsilon = 0.2$ . Mniejsze wartości powodowały wolniejsze uczenie – było to wynikiem słabszego przeszukiwania przestrzeni rozwiązań. Dla  $\varepsilon = 0$  metoda uczenia stosowała zawsze strategię zachłanną i proces uczenia stawał się nieskuteczny. Większe wartości  $\varepsilon$  powodowały zbyt losowy charakter poszukiwań, przez co czas nauki również się zwiększał.

<sup>8</sup> Badania przeprowadzono na komputerze z procesorem AMD Athlon 64 3000+ (zegar 2GHz).

Wykres na rys. 5b pokazuje wpływ współczynnika  $\gamma$  ( $\varepsilon = 0.1$  i  $\lambda = 0$ ). Podobnie jak powyżej dla każdej wartości  $\gamma$  wykonano 10 prób. Najszybciej uczenie przebiegało dla  $\gamma > 0.9$ , jednak przyjęcie zbyt dużych wartości (np.  $\gamma = 1$ ) powodowało, że uczenie stawało się nieskuteczne.

Wykres na rys. 5c pokazuje wpływ współczynnika  $\lambda$  ( $\varepsilon = 0.1$  i  $\gamma = 0.9$ ). Dla każdej wartości  $\lambda$  znowu przeprowadzono 10 prób. Duży rozrzut wartości czasu nauki i kształt wykresu wskazują, że parametr ten praktycznie nie ma wpływu na szybkość uczenia mierzoną w epokach. Jednak ze względu na konieczność modyfikacji funkcji  $V(x)$  dla każdego  $x$  oraz konieczność obliczania śladu aktywności, każda epoka trwała tu około 1.5 razy dłużej.



Rysunek 5. Wpływ parametrów  $\varepsilon$ ,  $\gamma$  i  $\lambda$  na szybkość uczenia

Dalsze badania miały na celu określenie jak zmienia się czas nauki w zależności od zadanej drogi  $d$ . Przyjęto:  $k = 1$ ,  $z = 3$ ,  $\eta = 0.2$ ,  $\varepsilon = 0.2$ ,  $\gamma = 0.9$  i  $\lambda = 0$ , a wyniki eksperymentów pokazane są na rys. 6. Dodatkowo zaprezentowane są odnalezione strategie.



strategie przedstawione są poniżej. Dla każdej z nich zadana droga  $d = 6$ , a pojemność zbiornika  $z = 3$ .

stan początkowy  $x = [3\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 3]$ : R1 T-1 R0 T3 R1 T1 R3 T2 R6  
 średni czas nauki: 2400 epok  
 zużycie paliwa: 8 jednostek

stan początkowy  $x = [3\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 3]$ : R1 T-1 R0 T3 R1 T1 R2 T1 R4 T1 R6  
 średni czas nauki: 3250 epok  
 zużycie paliwa: 8 jednostek

stan początkowy  $x = [0\ 0\ 2\ 1\ 0\ 0\ 0\ 0\ 3]$ : R2 T2 R3 T1 R6  
 średni czas nauki: 330 epok  
 zużycie paliwa: 5 jednostek

stan początkowy  $x = [3\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 3]$ : R1 T-1 R0 T3 R1 T1 R2 T1 R3 T1 R6  
 średni czas nauki: 4440 epok  
 zużycie paliwa: 8 jednostek

#### 4. Wnioski

Zaprezentowana metoda okazała się bardzo skutecznym narzędziem w poszukiwaniu strategii eksploracji. Dla długich sekwencji akcji czas poszukiwań był dość długi, jednak rozwiązanie zostało odnalezione dla wszystkich testowanych warunków zadania i ostatecznie metodę można uznać za uniwersalne narzędzie do poszukiwania strategii dla problemu dyskretnego.

Metoda obciążona jest przekleństwem wymiarowości – dla zadanej drogi  $d = 7$  i pojemności zbiornika  $z = 3$  szacunkowy czas obliczeń wyniósłby prawdopodobnie około 2 miesiące (ekstrapolacja na podstawie wykresu z rys. 6). Jednak dla zadań prostszych, dla których rozwiązaniem jest sekwencja kilkadziesiąt akcji, czas poszukiwania rozwiązania jest względnie krótki.

#### Bibliografia

- [1] Ball W.W.R., Coxeter H.S.M. *Mathematical Recreations and Essays*, 13edn, New York: Dover, 1987.
- [2] Brauer W., Brauer U. *Reconsidering the Jeep Problem – or how to transport a birthday present to Salosauna*, Proceedings of the Colloquium in Honor of Arto Salomaa on Results and Trends in Theoretical Computer Science, s. 30-33, 1994.
- [3] Cichosz P. *Systemy uczące się*, Wyd. Naukowo-Techniczne, Warszawa, 2000
- [4] Fine N.J. *The Jeep Problem*, The American Mathematical Monthly, v. 54, nr 1, s. 24-31, 1947.
- [5] Franklin J.N. *The range of a fleet of aircraft*, Journal of the Society for Industrial and Applied Mathematics, v. 8, nr 3, s. 541-548, 1960.
- [6] Gale D. *The Jeep Once More or Jeeper by the Dozen*, The American Mathematical Monthly, v. 77, nr 5, s. 493-501, 1970.



- 
- [7] Giffen W.J. *Deterministic and stochastic extensions of the Jeep problem*, Rozprawa doktorska, Purdue University, 2004.
  - [8] Imada A. *Can learning robot solve a 2-D jeep problem?* Proceedings of the International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE'2007), Buenos Aires, Argentina, 2007.
  - [9] Kaelbling L.P., Littman M.L., Moore A.W., *Reinforcement learning: A survey*, Journal of Artificial Intelligence Research, v. 4, s. 237-285, 1996.
  - [10] Kłesk P. *The Jeep Problem, searching for the best strategy with a genetic algorithm*, Information Processing and Security Systems, v. 3, Springer US, s. 453-464, 2005.
  - [11] Phipps C.G. *The Jeep Problem: A more general solution*, The American Mathematical Monthly, v. 54, s. 458-462, 1947.
  - [12] Pluciński M. *Application of the probabilistic RBF neural network in the reinforcement learning of a mobile robot*, Proceedings of the 14th International Multi-Conference on "Advanced Computer Systems", Międzyzdroje, Poland, 2007.
  - [13] Rote G., Zhang G. *Optimal logistics for expeditions – the jeep problem with complete refilling*, 'Spezialforschungsbereich F 003' Technical Report No. 71, TU Graz, Austria, 1996.
  - [14] Sutton R.S., *Learning to predict by the methods of temporal differences*, Machine Learning, v. 3, s. 9-44, 1992.
  - [15] Sutton R.S., Barto A.G., *Reinforcement learning: An introduction*, The MIT Press, 1998.
  - [16] Tesauro G., *Practical issues in temporal differences learning*, Machine Learning, v. 8, s. 257-277, 1992.
  - [17] E.W. Weisstein *Jeep Problem*, Published by Wolfram Research, MathWorld – A Wolfram Web Resource, <http://mathworld.wolfram.com/JeepProblem.html>.



# Rozpoznawanie twarzy dla obrazów widzialnych i podczerwieni z zastosowaniem funkcji transformacji w przestrzeni cech

Magdalena Szaber

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*Face recognition systems based on visual images have reached a significant level of maturity with some practical success. However, the performance of visual face recognition may degrade under poor illumination conditions and in completely darkness. Infrared images represent a viable alternative to visible images in the search for practical face recognition. While contemporary face recognition systems are limited to cases when identified and database images belong to the same category. We investigate the face recognition scheme where category of identified face images differs from that stored in face database.*

## Słowa kluczowe:

*face recognition, infrared images*

## 1. Wprowadzenie

Rozpoznawanie twarzy ma szerokie zastosowania w różnych systemach w tym: bezpieczeństwa, kontroli dostępu, komercyjnych i innych. Jednym z ważniejszych problemów rozpoznawaniu twarzy w świetle widzialnym jest wpływ oświetlenia. Z tego powodu w ostatnich latach powstało kilka rozwiązań w których rozpatrywane jest połączenie obrazów widzialnych z termicznymi podczerwieni [2–5]. W tych pracach rozpatrzono kilka wariantów bazujących na odpowiednim wagowym dodawaniu jednego obrazu do drugiego w przestrzeni obrazu oraz przestrzeni cech tych obrazów [4, 5]. Pozwala to na wyrównanie oświetlenia w otrzymanym obrazie, ponieważ obraz termiczny eliminuje zmienne oświetlenie w obrazie widzialnym.

Inne podejście zaproponowano w [8] polegające na wzajemnej transformacji cech obrazu widzialnego w termiczny podczerwieni i na odwrót. Pozwala to na budowę nowych metod, które działają niezależnie od rodzaju obrazu na wejściu odpowiedniego systemu rozpoznawania. Przykładem zastosowań takich systemów może być dostęp do własnych mień (domki, garaże, piwnice) niezależnie od oświetlenia, jeśli na wejściu systemu są używane dwie kamery jednocześnie – jedna dająca obraz w świetle widzialnym, a druga w podczerwieni.

Funkcja transformacji zaproponowana w [8] bazuje na operacji pokropkowego dzielenia i mnożenia odpowiednich zestawów cech. Metoda [8] działa szybko, ale

ma jedną istotną wadę związaną z „nie analityczną” procedurą wyliczania funkcji transformacji. Jest to związane z operacją dzielenia przez wartości zerowe. W [8] stosuje się zamianę wartości zerowej na wartość bliską zeru co ratuje sytuację, ale wprowadza dodatkowe błędy obliczeniowe.

W niniejszym artykule zaproponowano funkcję transformacji bazującą na odejmowaniu i dodawaniu odpowiednich zestawów cech. W porównaniu z podejściem przedstawionym w [8] zalety nowej metody będą takie same, ale wspomniana wada jest wyeliminowana.

### 1.1. Koncepcja nowego podejścia

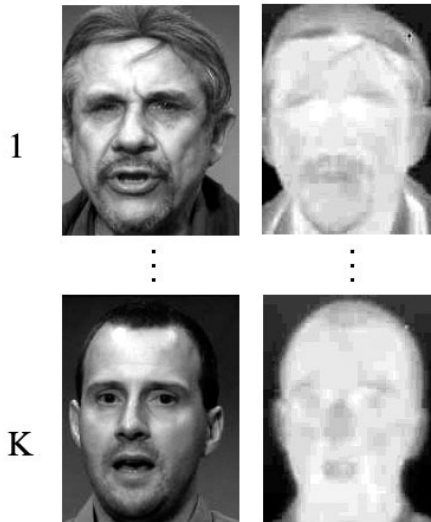
Zakładamy, że w bazie danych systemu jest  $K$  klas obrazów twarzy, a w każdej klasie jest para obrazów : jeden obraz widzialny ( $O^{(VIS)}$ ) i jeden obraz termiczny podczerwieni ( $O^{(TH)}$ ). Przykład takiej bazy danych pokazany jest na rysunku 1, gdzie w lewej kolumnie jest przedstawiony obrazy w świetle widzialnym, a prawej kolumnie obrazy termiczne.

W procesie rozpoznawania, na wejściu systemu występują dwa warianty obrazów:

- pierwszy wariant: obraz widzialny,
- drugi wariant: obraz termiczny podczerwieni.

Każdy obraz w klasie można zapisać w następującej formie:

$$O^{(k,VIS)} \text{ lub } O^{(k,TH)}, \text{ gdzie } \forall k \in \overline{1, K}.$$



Rysunek 1. Początkowa baza systemu rozpoznawania

W przestrzeni cech otrzymanej za pomocą macierzy transformacji  $T$ , każdy obraz można zapisać następująco:

$$C^{(k, \text{VIS})} = T[O^{(k, \text{VIS})}], \quad (1)$$

$$C^{(k, \text{TH})} = T[O^{(k, \text{TH})}], \quad (2)$$

gdzie  $C^{(k, \text{VIS})}$  i  $C^{(k, \text{TH})}$  to wynik transformacji w przestrzeni cech.

Na podstawie (1) i (2) można zapisać funkcję transformacji  $f^{(k, \text{TH} \rightarrow \text{VIS})}$  zestawu cech obrazu termicznego podczerwieni w zestaw cech obrazu widzialnego:

$$f^{(k, \text{TH} \rightarrow \text{VIS})} = C^{(k, \text{VIS})} - C^{(k, \text{TH})} \quad (3)$$

Jeśli na wejściu mamy obraz widzialny (w świetle dziennym) to taki obraz jest rozpoznawany w tradycyjny sposób. Jeśli na wejściu mamy dwa obrazy widzialny i termiczny podczerwieni to dokonujemy rozpoznawania przez fuzję dwóch obrazów [3–5].

W czasie trwania zmerchu lub nocy, na wejściu systemu rozpoznawania wykorzystujemy jedynie obraz termiczny (Query Face):  $O^{(\text{QF}, \text{TH})}$ . Jest to najtrudniejszy wariant dla systemu rozpoznawania, dlatego w niniejszym artykule rozpatrujemy tylko tę sytuację.

W tym celu wyliczamy cechy obrazu  $O^{(\text{QF}, \text{TH})}$ , w wyniku czego otrzymujemy następujący zestaw cech:

$$C^{(\text{QF}, \text{TH})} = T[O^{(\text{QF}, \text{TH})}] \quad (4)$$

Teraz otrzymujemy  $K$  zestawów cech  $\hat{C}_k^{(\text{QF}, \text{VIS})}$  dla obrazu  $O^{(\text{QF}, \text{TH})}$  z użyciem funkcji transformacji (3) według następujących obliczeń:

$$\hat{C}_k^{(\text{QF}, \text{VIS})} = f^{(k, \text{TH} \rightarrow \text{VIS})} + C^{(\text{QF}, \text{TH})}, \quad \forall k \in \overline{1, K} \quad (5)$$

Równanie (5) może być zapisane także w następującej formie:

$$\hat{C}_k^{(\text{QF}, \text{VIS})} = C^{(k, \text{VIS})} - C^{(k, \text{TH})} + C^{(\text{QF}, \text{TH})}, \quad \forall k \in \overline{1, K} \quad (6)$$

Daszek nad literą  $C$  podkreśla, że obrazy  $\hat{C}_k^{(\text{QF}, \text{VIS})}$  i  $C^{(k, \text{VIS})}$  różnią się pomiędzy sobą, ponieważ  $C^{(k, \text{TH})} \neq C^{(\text{QF}, \text{TH})}$ .

Kolejnym krokiem jest klasyfikacja obrazu  $\hat{C}_k^{(\text{QF}, \text{VIS})}$ , której dokonujemy porównując ten obraz ze wszystkimi obrazami  $C^{(k, \text{VIS})}$  znajdującymi się w bazie danych systemu. Przykład takiej klasyfikacji można zobaczyć w [8].

## 2. Obliczenie zestawu cech oraz odwrotnej transformacji

Przykładowo, pokażemy sposób wyliczenia cech na podstawie dwuwymiarowej transformacji kosinusowej (2D DCT). W tym przypadku zamiast (2) mamy następujące rozwiązanie [1]:

$$C_{p \times p}^{(k, \text{TH})} = T_{p \times M} O_{M \times N}^{(k, \text{TH})} T_{N \times p} \quad (7)$$

gdzie:  $O_{M \times N}^{(k, \text{TH})}$  – obraz w dziedzinie czasu;  $M \times N$ ,  $p \times p$ ,  $p \times M$  oraz  $N \times p$  – rozmiary odpowiednich macierzy;  $T_{p \times M}$  oraz  $T_{N \times p}$  – macierze kosinusowej transformacji:

$$T_{P \times M} = [t_M^{(p, m)}]$$

$$t_M^{(p, m)} = \begin{cases} 1/\sqrt{M}, & p = 0, & m = 0, 1, \dots, M-1 \\ \sqrt{\frac{2}{M}} \cos \frac{\pi(2m+1)p}{2M}, & p = 1, \dots, P-1, & m = 0, 1, \dots, M-1 \end{cases}$$

$$T_{N \times P} = [t_N^{(n, p)}]$$

$$t_N^{(n, p)} = \begin{cases} 1/\sqrt{N}, & n = 0, 1, \dots, N-1, & p = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{\pi(2n+1)p}{2N}, & n = 0, 1, \dots, N-1, & p = 1, \dots, P-1 \end{cases}$$

Schemat transformacji (7) przedstawiony został na rysunku 2 [9].

W wyniku wykonania transformacji (7) zamiast obrazu mającego rozmiar  $M \times N$  pikseli otrzymujemy zestaw cech (w danym przypadku widma) przedstawionej macierzą  $C$  o rozmiarze  $p \times p$  pikseli. Najistotniejsze składowe tego widma znajdują się w górnej części tej macierzy. Dlatego wynikowy zestaw cech zapisuje się tylko z tej części. Zwykle wykorzystuje się zapis składowych po „zygzaku” [7].

Jeżeli system działa z obrazem rekonstruowanym z widma cech do przestrzeni czasowej to wykorzystujemy  $\forall k \in \overline{1, K}$  dwa następujące równania:

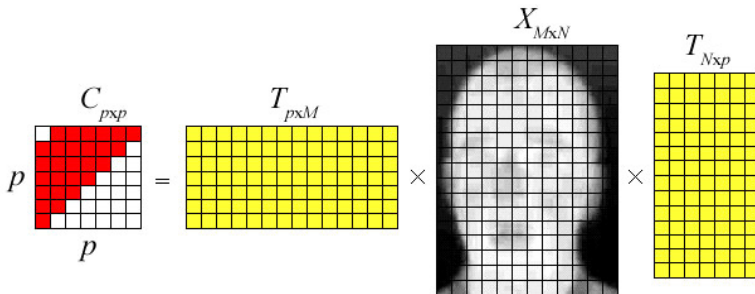
$$O_k^{(\bullet)} = T^{-1}[C_k^{(\bullet)}]; \quad (8)$$

$$O_{M \times N}^{(k, \text{TH})} = T_{N \times p}^{-1} C_{p \times p}^{(k, \text{TH})} T_{p \times M}^{-1} \quad (9)$$

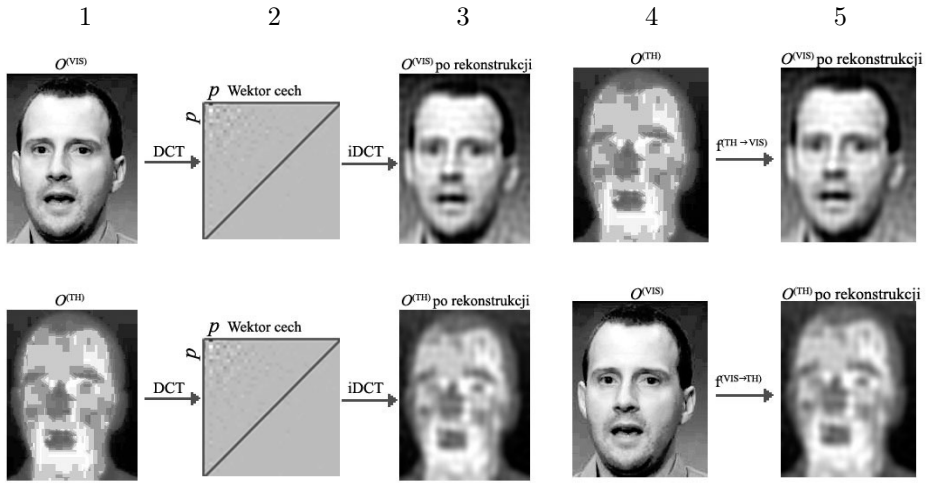
gdzie  $(\bullet)$  oznacza dowolny obraz, a znak „-1” oznacza odwrotną macierz.

Rekonstrukcja obrazów na podstawie prostej (7) i odwrotnej (9) transformacji 2D DCT pokazane jest na rysunku 3 w kolumnach 1, 2, 3. Przy czym: 1 – obrazy wejściowe w świetle widzialnym i termicznym podczerwieni; 2 – schemat widma tych obrazów; 3 – rekonstrukcja obrazu z odpowiedniego widma.

Natomiast, rekonstrukcja obrazów z odpowiedniego widma na podstawie funkcji transformacji (3) i (5) przedstawiona jest na rysunku 3 w kolumnach 4 i 5. Przy



Rysunek 2. Wyliczenie cech z obrazu wejściowego na podstawie 2D DCT



Rysunek 3. Warianty rekonstrukcji obrazów z jednej kategorii do drugiej

czym w górnym wierszu rezultaty dokładnie odpowiadają wzorom (3) i (5). W dolnym wierszu sytuacja jest przeciwna – obraz wejściowy w świetle widzialnym jest konwertowany do obrazu termicznego podczerwieni. Używany jest dlatego wzór (3) i (5) z odpowiednim uwzględnieniem obrazów.

Zamiast 2D DCT procedury (7) i (9) można rozwiązać na podstawie 2D DFT lub innych odwrotnych transformacji.

### 3. Eksperyment

Eksperymentu dokonano na bazie Equinox [6]. Dla eksperymentu używana jest część bazy Equinox zawierającej 89 klas i w każdej klasie używamy jeden obraz rodzaju  $O^{(k,VIS)}$  i dziewięć obrazów  $O^{(k,TH)}$ .

W każdym eksperymencie używano odpowiednią parę obrazów  $O^{(k,VIS)}$  i  $O^{(k,TH)}$  oraz  $89 \times 8 = 712$  obrazów testowych (według liczby klas pomnożonych przez liczbę obrazów termicznych podczerwieni w każdej klasie). Eksperymenty dokonano wraz z procedurą krosvalidacji co oznacza, że w każdym eksperymencie dla pary  $O^{(k,VIS)}$  i  $O^{(k,TH)}$  zmieniany był drugi obraz. Przy tym odpowiednio zmieniały się “8” obrazów testowych, co dało we wszystkich eksperymentach wspólną liczbę obrazów testowych równą  $6408 (712 \times 8)$ .

Dla wyliczenia cech wykorzystano procedurę 2D DCT według (7) z parametrem  $p=11$ , w rezultacie otrzymując wektor cech po procedurze ‘zygzaku’ zawierał 65 składowych [7].

Cechy obrazu  $QF$  (termicznego) transformowane są do cech obrazu widzialnego przy pomocy równania (3).

Porównanie cech dwóch obrazów ( $QF$  i każdego obrazu z bazy danych) realizuje się w komparatorze systemu rozpoznawania, a miarą bliskości wektorów cech

określona jest na podstawie kryterium minimalnej odległości z użyciem metryki L2.

Obraz zaliczamy do odpowiedniej klasy wtedy, kiedy znajduje się na pierwszym miejscu (rank=1) posortowanych rosnąco odległości. Oznacza to, że testowany obraz ma minimalną odległość dla odpowiedniej klasy.

Używając notacji schematu eksperymentów przedstawionej w [10], można przedstawiony powyżej eksperyment zapisać następująco:

Equinox(89/1<sup>(VIS)</sup>/8<sup>(TH)</sup>/CrossValidation){DCT/65/f<sup>(TH→VIS)</sup>/L2, rank = 1}

Wyniki rozpoznawania zostały wyliczone jako proporcja pomiędzy obrazami poprawnie rozpoznanymi, a całkowitą liczbą obrazów testowych i wyliczone zostały procentowo.

Dla przedstawionego wyżej eksperymentu z 6408 obrazów testowych zostało prawidłowo rozpoznanych 6166 obrazów, co dało w rezultacie dokładność rozpoznawania 96.22% dla bazy, zawierającej 89 klas obrazów.

## 4. Wnioski

Przedstawione wyniki dowodzą, że jest możliwe użycie funkcji transformacji, kiedy kategoria obrazu twarzy na wejściu systemu rozpoznawania różni się od kategorii obrazu w bazie.

Wysokie współczynniki rozpoznawania otrzymano dla obrazów termicznych podczerwieni potwierdzającą wysoką skuteczność metody, co może mieć bardzo szerokie zastosowanie w praktyce.

## Podziękowania

Autor pracy chciałby złożyć szczególne podziękowania i wyrazy szacunku swojemu Promotorowi – Panu Profesorowi Georgy Kukharev’owi – za poświęcony czas, przekazaną wiedzę i inspirację. Jego wsparcie i pasja były nieocenioną pomocą przy powstaniu niniejszego artykułu.

## Bibliografia

- [1] Kukharev G., Kuźmiński A., *Techniki Biometryczne. Cz.1. Metody Rozpoznawania Twarzy*. Wydział Informatyki, Politechnika Szczecińska, 2003.
- [2] Selinger A., Socolinsky Diego A.. *Appearance-Based Facial Recognition Using Visible and Thermal Imagery: A Comparative Study*. Technical Report 02-01, Equinox Corporation, February 2002.
- [3] Kong S. G., Heo J., Abidi B., Paik J. and Abidi M. A., *Recent advances in visual and infrared face recognition—a review*. Computer Vision and Image Understanding, 97(1):103–135, 1997.
- [4] Heo J., Abidi B., Kong S.G. and Abidi M. A., *Performance comparison of visual and thermal signatures for face recognition*. Biometric Consortium Conference, Arlington, VA, September 2003.



- 
- [5] Singh S., Gyaourova A., Bebis G. and Pavlidis I., *Infrared and visible image fusion for face recognition*. In Proc. SPIE Defense and Security Symposium (Biometric Technology for Human Identification), pp. 585–596, 2004.
  - [6] Available at : <http://www.equinoxsensors.com/products/HID.html>
  - [7] Kukharev G., Nowosielski A., *Face Recognition Using the Simple Feature Extractors*. Computing, Multimedia and Intelligent Techniques. Special issue on Live Biometrics and Security 1(1), 87-98, 2005.
  - [8] Kukharev G., Szaber M., Nowosielski A., *Transform Function for Visual and Infrared Images for the Face Recognition System*, Polish Journal of Environmental Studies Vol. 16, No 3C, 293–296, 2007.
  - [9] Forczmański P., Kukharev G., *Comparative analysis of simple facial features extractors*. Journal of Real-Time Image Proc, pp. 239–255, 2007.
  - [10] Kukharev G., Mikłasz M., Nguyen The Binh, *Strategia budowy systemów klasy „Face Retrieval”*. Metody Informatyki Stosowanej nr 2, s. 61-72, 2007.



# Strategie racjonalizacji obliczeń przy wyznaczaniu iloczynów macierzowo-wektorowych

Alexandr Țariov

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*Strategies of rationalization of computing of constant coefficient matrix by vector multiplication are offered. The concrete example of synthesis of fast algorithm for matrix by vector multiplication is considered. The offered example allows tracking all stages of construction of the algorithm rationalized from point of view number multiplication minimization.*

## Słowa kluczowe:

*matrix-vector multiplication, fast algorithms, reduction of number of arithmetical operations*

## 1. Wprowadzenie

Konieczność wyznaczania iloczynu macierzowo-wektorowego występuje w wielu zadaniach numerycznych związanych z cyfrowym przetwarzaniem sygnałów, rozpoznawaniem obrazów, kompresją plików audio i video [1]. Wystarczy powiedzieć, iż typowe dla cyfrowego przetwarzania danych operacje: splot kołowy i liniowy, korelacja, transformaty w rozmaitych bazach, dyskretne przekształcenie falkowe są z natury rzeczy iloczynami macierzowo-wektorowymi.

Poprzez analizę struktur różnych macierzy bazowych, odkryto, iż niektóre relacje pomiędzy ich elementami pozwalają na zmniejszenie liczby operacji arytmetycznych. Stwierdzenie tego faktu dało początek pracom dotyczącym syntezy algorytmów związanych z realizacją iloczynów macierzowo-wektorowych optymalizowanych pod kątem liczby operacji mnożenia i dodawania. Przykładem algorytmu tworzonoego na podstawie wykorzystania „sprzyjających” relacji elementów wewnątrz macierzy jest FFT (*Fast Fourier Transform*) autorstwa Cooley’a i Tukey’a czyli „szybka transformata Fouriera”, wykorzystująca w celu redukcji liczby operacji arytmetycznych niezbędnych przy jej realizacji, cechy multiplikatywności oraz okresowości dyskretnej funkcji wykładniczych będących elementami macierzy bazy dyskretnej przekształcenia Fouriera [2]. Pojawiły się również inne „szybkie” algorytmy wyznaczania dyskretnej przekształcenia Fouriera oraz „szybkiego splotu” opracowane przez S. Winograda na podstawie chińskiego twierdzenia o resztach, algorytm Cooka-Tooma – wyznaczania splotu liniowego na podstawie efektywnego sposobu mnożenia wielomianów opartego na interpolacji wielomianu

Lagrange'a [2-4]. Powstały również rozmaite modyfikacje algorytmów „szybkich transformat ortogonalnych”: DFT (*discrete Fourier transform*), DCT (*discrete cosine transform*), DST (*discrete sine transform*), DHT (*discrete Hartley transform*) i inne, opisane w pracach R. C. Agarwala, N. Ahmeda, C. S. Burrusa, D. P. Kolby, T. W. Parksa, C. M. Radera, K. R. Rao, J. H. McClellana, H. J. Nussbaamera, R. N. Bracewella [2-7].

Należy podkreślić, iż aparat matematyczny, który stosują autorzy do udowodnienia idei wyprowadzenia szybkich algorytmów często jest dość skomplikowany (teoria wielomianów, teoria liczb oraz pierścieni Galois itp.). W efekcie dla praktyków nieznających tych zagadnień trudne jest zrozumienie sedna proponowanych rozwiązań. Ponadto wspomniane teorie oraz wynalezione algorytmy obejmują niezbyt szeroki krąg specjalizowanych zadań. Poza wymienionymi rozwiązaniami istnieje wiele innych zadań, gdzie występują macierze posiadające strukturę pozwalającą na zmniejszenie liczby działań arytmetycznych przy realizacji operacji mnożenia wektora przez macierz. Niestety dla tych zadań nie istnieje żadna ogólna oraz przystępna metoda, pozwalająca syntetyzować szybkie algorytmy.

Niespodziewanie okazało się, że istnieje nadzwyczaj prosta a jednocześnie skuteczna metoda (a raczej zestaw strategii) poszukiwania rozwiązań algorytmicznych przy syntezy zoptymalizowanych (lub co najmniej zracjonalizowanych) pod względem liczby niezbędnych operacji mnożenia i dodawania iloczynów macierzowo-wektorowych. Właśnie te strategie zostaną przedstawione Czytelnikowi w dalszej części niniejszej pracy.

## **2. Określenie zestawu wzorcowych struktur macierzy pozwalających na redukcję liczby operacji arytmetycznych przy wyznaczaniu iloczynów macierzowo-wektorowych**

Zadaniem racjonalizacji obliczeń przy wyznaczaniu iloczynów macierzowo-wektorowych jest minimalizacja łącznej liczby operacji mnożenia oraz dodawania [8]. Należy zastrzec, iż nie zawsze udaje się zapewnić jednoczesną redukcję liczby operacji mnożenia oraz operacji dodawania. Czasem występuje sytuacja, gdy możliwa jest redukcja liczby operacji mnożenia kosztem zwiększenia liczby operacji dodawania. Przy tym są przypadki, gdy łączna liczba operacji arytmetycznych okazuje się jednakże mniejsza od pierwotnej. W tych przypadkach „zysk obliczeniowy” jest praktycznie oczywisty. Natomiast, jeśli łączna liczba operacji arytmetycznych jest większą od pierwotnej w wyniku pojawienia się dodatkowych operacji dodawania, korzyść takiej racjonalizacji należy uzasadnić.

Tym nie mniej przedstawione zostaną strategii redukcji operacji arytmetycznych przy wyznaczaniu iloczynów macierzowo-wektorowych.

W tabeli 1 przedstawiono zestaw typowych struktur macierzy (nazwiemy je „wzorcami” lub „wzorcowymi”), których układ blokowy pozwala na zmniejszenie liczby operacji arytmetycznych niezbędnych przy wyznaczaniu iloczynów macierzy przez wektory. Możliwość redukcji operacji arytmetycznych dla części tych struk-

tur była między innymi pokazana w pracy [9] dla macierzy o rozmiarach  $(2 \times 2)$  oraz w pracy [10] dla macierzy o rozmiarach  $(N \times N)$ . Poszerzony zestaw procedur obliczeniowych, uwzględniających taką redukcję został przedstawiony w tabeli 2<sup>9</sup>.

Tabela 1. Struktury blokowe wzorców macierzowych

No	Typ wzorca	Liczba operacji arytmetycznych		No	Typ wzorca	Liczba operacji arytmetycznych	
		„×”	„+”			„×”	„+”
0	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{C}_{\frac{N}{2}} & \mathbf{D}_{\frac{N}{2}} \end{array} \right)$	$N^2$	$N(N-1)$	6	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \\ \hline \mathbf{B}_{\frac{N}{2}} & \mathbf{C}_{\frac{N}{2}} \end{array} \right)$	$\frac{3N^2}{4}$	$\frac{N(3N-2)}{4}$
1	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \\ \hline \mathbf{A}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{N^2}{4}$	$\frac{N}{2}$	7	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{A}_{\frac{N}{2}} & \mathbf{C}_{\frac{N}{2}} \end{array} \right)$	$\frac{3N^2}{4}$	$\frac{N(3N-2)}{4}$
2	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \\ \hline \mathbf{A}_{\frac{N}{2}} & -\mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{N^2}{2}$	$N$	8	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{C}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{3N^2}{4}$	$\frac{3N^2}{4}$
3	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \\ \hline \mathbf{B}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \end{array} \right)$	$\frac{N^2}{2}$	$\frac{N(N-1)}{2}$	9	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{C}_{\frac{N}{2}} & -\mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{3N^2}{4}$	$\frac{3N^2}{4}$
4	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \end{array} \right)$	$\frac{N^2}{2}$	$\frac{N(N-1)}{2}$	10	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{B}_{\frac{N}{2}} & -\mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{3N^2}{4}$	$\frac{3N^2}{4}$
5	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline \mathbf{B}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{N^2}{2}$	$\frac{N(N+2)}{2}$	11	$\left( \begin{array}{c c} \mathbf{A}_{\frac{N}{2}} & \mathbf{B}_{\frac{N}{2}} \\ \hline -\mathbf{B}_{\frac{N}{2}} & -\mathbf{A}_{\frac{N}{2}} \end{array} \right)$	$\frac{3N^2}{4}$	$\frac{3N^2}{4}$

### 3. Strategie konstruowania „szybkich” algorytmów mnożenia wektora przez macierz

Pierwsza strategia racjonalizacji obliczeń przy wyznaczaniu iloczynów macierzowo-wektorowych sprowadza się do poszukiwania takich fragmentów struktury macierzy, których kompozycji zgadzałyby się z jakąkolwiek strukturą wzorcową. Na szczęście występujące w zadaniach cyfrowego przetwarzania sygnałów macierze akurat posiadają taką specyficzną kompozycją bloków lub mogą zostać sprowadzone do takiej postaci za pomocą mieszania wierszy lub kolumn tych macierzy.

Tak więc pierwsze, co należy zrobić to podzielić myślowo macierz pierwotną na cztery podmacierze i znaleźć zgodność kompozycji powstałych w wyniku tego podziału bloków z kompozycją bloków jednego z wzorców (macierzy wzorcowych z tabeli 1).

W ten sposób wybierana jest jedna z procedur obliczeniowych z tabeli 2. Jeśli znaleziona kompozycja bloków odpowiada wzorcom 1-11, oznacza to, że już, (czyli po pierwszym kroku) udało się zminimalizować liczbę operacji arytmetycznych.

<sup>9</sup> Macierze  $\mathbf{0}_{\frac{N}{2}}$  oraz  $\mathbf{I}_{\frac{N}{2}}$  w tabeli 2 oraz w dalszej części artykułu oznaczają odpowiednio macierz zerową oraz jednostkową o wymiarach oznaczonych za pomocą dolnych indeksów.



Jeśli struktura blokowa macierzy odpowiada wzorcowi o numerze 0, oznacza to, że mnożenie przez taką macierz, jak na razie, nie da się zrationalizować.

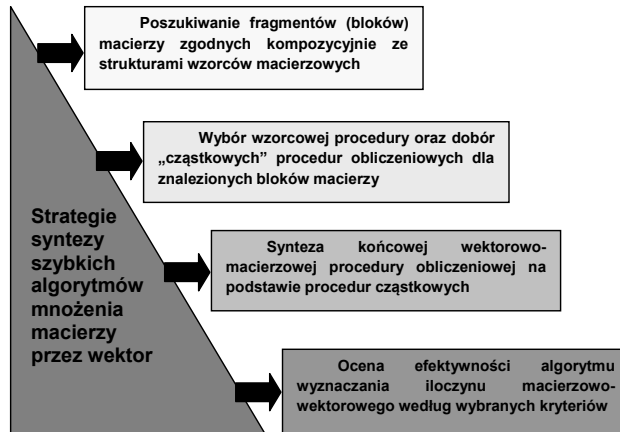
W takim przypadku można wykonać tasowanie wierszy i/lub kolumn macierzy pierwotnej zanim powtórzy się próbę. Sukces wyszukania odpowiedniej kompozycji bloków ściśle zależy od trafnego wyboru sposobu tasowania wiersz i/lub kolumn. Należy jednak pamiętać, że tasowanie wierszy macierzy powoduje konieczność tasowania elementów wektora wyjściowego (rezultatu), zaś tasowanie kolumn wywołą potrzebę wstępnego tasowania elementów wektora wejściowego.

Należy też podkreślić, iż jeśli wstępnie (czyli w pierwszym kroku) nie udało się znaleźć zrationalizowanej procedury mnożenia wektora przez macierz, to nie można wyciągnąć wniosku, że nie uda się zrealizować ten zamiar w kolejnych krokach. Więc niezależnie od wyniku pierwszego kroku przejść należy do kroku drugiego.

Kolejnym krokiem racjonalizacji algorytmu mnożenia wektora przez macierz jest analogiczny podział każdej z trzech (lub czterech – w najgorszym przypadku) podmacierzy powstałych w wyniku syntezy procedury obliczeniowej na poprzednim kroku oraz poszukiwanie adekwatnej, czyli zgodnej ze wzorcem, kompozycji bloków dla każdej z tych podmacierzy. Dalej wybrana winna być odpowiednia procedura dla każdej z podmacierzy i na jej podstawie syntetyzowana wypadkowa procedura dla tego kroku racjonalizacji. W razie znalezienia zgodności kompozycji blokowej z jakimś ze wzorców (o numerach od 1 do 10) chociażby dla jednej z podmacierzy, uzyskuje się jeszcze bardziej zrationalizowaną wypadkową procedurę obliczeniową.

Postępując w taki sposób można dojść do podmacierzy rzędu  $2 \times 2$  maksymalnie redukując liczbę niezbędnych operacji arytmetycznych. Jednak taka kompletna dekompozycja udaje się nie zawsze, a tym bardziej nie dla wszystkich podmacierzy pojawiających się w wyniku podziału blokowego pierwotnych macierzy na każdym etapie dekompozycji. Zależy to z jednej strony od właściwości strukturalnych macierzy, z drugiej zaś – od inwencji projektanta, ponieważ nietrafny dobór wymienianych przy tasowaniu wierszy i kolumn macierzy (lub podmacierzy) na każdym etapie przetwarzania może spowodować „nieoptymalną” dekompozycję macierzy pierwotnych.

Należy podkreślić również, że nie zawsze iteracyjny podział macierzy pierwotnych na cztery podmacierze doprowadza do redukcji liczby operacji nawet po kilkukrotnym tasowaniu wiersz i/lub kolumn tych macierzy. W takim razie warto poszukać innych wariantów. Na przykład można przedstawić macierz pierwotną w postaci sumy lub też różnicy dwóch macierzy, a następnie spróbować zastosować opisane wyżej sposoby racjonalizacji obliczeń dla każdej z tych macierzy. Oczywiście, iż takie przekształcenie ma sens wtedy, gdy łączna liczba operacji powstających w wyniku takiej dekompozycji jest mniejsza od pierwotnej. Innym podejściem może być poszukiwanie pojedynczych fragmentów (subbloków) macierzy, mających strukturę wzorcową. W tym ostatnim przypadku zmniejszenie liczby operacji arytmetycznych można prawdopodobnie uzyskać tylko dla znalezionych fragmentów.



Rysunek 1. Strategie racjonalizacji obliczeń przy syntezie szybkich algorytmów mnożenia macierzy przez wektor

#### 4. Przykład syntezy „szybkiego” algorytmu mnożenia macierzy przez wektor

Rozważana niech będzie synteza szybkiego algorytmu wyznaczenia iloczynu macierzowo-wektorowego dla jakiejś macierzy wartości stałych. Niech macierz ta ma następującą postać:

$$\mathbf{A}_8 = \left[ \begin{array}{cccc|cccc} a & b & d & e & f & g & h & k \\ c & a & e & d & g & f & k & h \\ d & e & a & b & h & k & f & g \\ e & d & c & a & k & h & g & f \\ \hline f & g & h & k & a & b & d & e \\ g & f & k & h & c & a & e & d \\ h & k & f & g & d & e & a & b \\ k & h & g & f & e & d & c & a \end{array} \right].$$

Należy wyliczyć wynik mnożenia tej macierzy przez wektor danych  $\mathbf{X}_{8 \times 1}$ , czyli iloczyn:

$$\mathbf{Y}_{8 \times 1} = \mathbf{A}_8 \mathbf{X}_{8 \times 1} \quad (1)$$

Podkreślić należy, iż w przypadku realizacji procedury mnożenia w sposób bezpośredni, tzn. zgodnie z (1), niezbędne jest wykonanie 64 mnożeń oraz 56 dodawań. Podjęta niech będzie próba racjonalizacji procesu obliczeniowego w ten sposób, żeby zmniejszyć co najmniej liczbę mnożeń.



Można zauważyć, iż macierz  $\mathbf{A}_8$  ma specyficzną strukturę blokową:

$$\mathbf{A}_8 = \left[ \begin{array}{c|c} \mathbf{A}_4 & \mathbf{B}_4 \\ \hline \mathbf{B}_4 & \mathbf{A}_4 \end{array} \right],$$

$$\mathbf{A}_4 = \begin{bmatrix} a & b & d & e \\ c & a & e & d \\ d & e & a & b \\ e & d & c & a \end{bmatrix}, \mathbf{B}_4 = \begin{bmatrix} f & g & h & k \\ g & f & k & h \\ h & k & f & g \\ k & h & g & f \end{bmatrix}.$$

Kompozycja bloków odpowiada strukturze wzorcowej, która podana jest w tabeli 1 pod numerem 5. Istnieje więc możliwość redukcji operacji arytmetycznych.

Wtedy zrationalizowana procedura wyznaczenia rozważanego iloczynu przybierze następującą postać:

$$\mathbf{Y}_{8 \times 1} = (\mathbf{E}_2 \otimes \mathbf{I}_4) \mathbf{D}_8^{(1)} (\mathbf{E}_2 \otimes \mathbf{I}_4) \mathbf{X}_{8 \times 1} \quad (2)$$

gdzie macierz  $\mathbf{E}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$  jest macierzą Hadamarda drugiego rzędu [2], natomiast macierz  $\mathbf{D}_8^{(1)} = \frac{1}{2}(\mathbf{A}_4 + \mathbf{B}_4) \oplus \frac{1}{2}(\mathbf{A}_4 - \mathbf{B}_4)$ ;

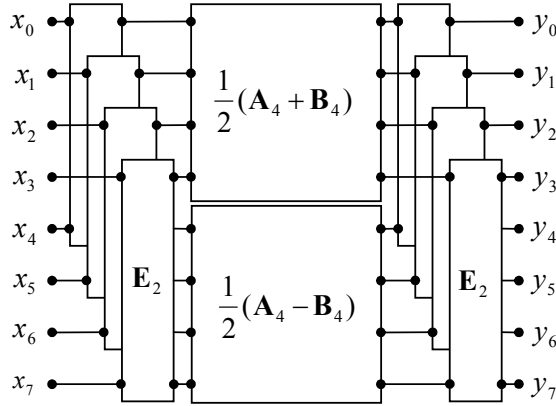
$$\mathbf{E}_2 \otimes \mathbf{I}_4 = \left[ \begin{array}{cccc|cccc} 1 & & & & & & & 1 \\ & 1 & & & & & & 1 \\ & & 1 & & & & & 1 \\ & & & 1 & & & & 1 \\ \hline 1 & & & & & & & -1 \\ & 1 & & & & & -1 & \\ & & 1 & & & & -1 & \\ & & & 1 & & & -1 & \end{array} \right],$$

zaś

$$\mathbf{A}_4 + \mathbf{B}_4 = \left[ \begin{array}{cc|cc} a+f & b+g & d+h & e+k \\ \hline c+g & a+f & e+k & d+h \\ \hline d+h & e+k & a+f & b+g \\ e+k & d+h & c+g & a+f \end{array} \right], \quad (3)$$

$$\mathbf{A}_4 - \mathbf{B}_4 = \left[ \begin{array}{cc|cc} a-f & b-g & d-h & e-k \\ \hline c-g & a-f & e-k & d-h \\ \hline d-h & e-k & a-f & b-g \\ e-k & d-h & c-g & a-f \end{array} \right]. \quad (4)$$

Symbole „ $\otimes$ ” oraz „ $\oplus$ ” tu i w dalszej części artykułu oznaczają odpowiednio operację iloczynu tensorowego oraz sumy prostej dwóch macierzy [5,7,11].



Rysunek 2. Model grafostukturalny organizacji procesu obliczeniowego wyznaczenia iloczynu macierzy  $\mathbf{A}_8$  przez wektor  $\mathbf{X}_{8 \times 1}$  po pierwszym etapie racjonalizacji

Na rysunku 2 został przedstawiony model grafostukturalny, reprezentujący strukturę algorytmiczną organizacji procesu wyznaczenia iloczynu na pierwszym poziomie detalizacji. Liniami prostymi oznaczone są operacje transferu danych. Prostokątami oznaczone zostały bloki mnożenia wpisanych w nich macierzy przez odpowiednie podwektory danych.

Zauważmy, iż podmacierze (3) oraz (4), powstałe w wyniku syntezy procedury (2), mają też taką samą kompozycję bloków, jaką zauważono w przypadku macierzy  $\mathbf{A}_8$ :

$$\mathbf{A}_4 + \mathbf{B}_4 = \left[ \begin{array}{c|c} \mathbf{A}_2 & \mathbf{B}_2 \\ \hline \mathbf{B}_2 & \mathbf{A}_2 \end{array} \right], \quad \mathbf{A}_4 - \mathbf{B}_4 = \left[ \begin{array}{c|c} \mathbf{C}_2 & \mathbf{D}_2 \\ \hline \mathbf{D}_2 & \mathbf{C}_2 \end{array} \right],$$

gdzie:

$$\mathbf{A}_2 = \left[ \begin{array}{c|c} a+f & b+g \\ \hline c+g & a+f \end{array} \right], \quad \mathbf{B}_2 = \left[ \begin{array}{c|c} d+h & e+k \\ \hline e+k & d+h \end{array} \right],$$

$$\mathbf{C}_2 = \left[ \begin{array}{c|c} a-f & b-g \\ \hline c-g & a-f \end{array} \right], \quad \mathbf{D}_2 = \left[ \begin{array}{c|c} d-h & e-k \\ \hline e-k & d-h \end{array} \right].$$

Można więc jeszcze raz zastosować ten sam sposób redukcji liczby operacji dla każdej z powstałych podmacierzy.

Wykonując odpowiednie dekompozycje komponentów macierzowych, wchodzących w skład procedury (2) oraz modelu na rysunku 2, syntetyzuje się nową uściśloną procedurę obliczeniową dla drugiego kroku racjonalizacji algorytmu. Wtedy procedura wyznaczenia iloczynu przybierze następującą postać:

$$\mathbf{Y}_{8 \times 1} = (\mathbf{E}_2 \otimes \mathbf{I}_4)(\mathbf{I}_2 \otimes \mathbf{E}_2 \otimes \mathbf{I}_2)\mathbf{D}_8^{(2)}(\mathbf{I}_2 \otimes \mathbf{E}_2 \otimes \mathbf{I}_2)(\mathbf{E}_2 \otimes \mathbf{I}_4)\mathbf{X}_{8 \times 1}, \quad (5)$$

$$\text{gdzie } \mathbf{D}_8^{(2)} = \text{diag} \left( \begin{array}{c} \frac{1}{4}(\mathbf{A}_2 + \mathbf{B}_2) \\ \frac{1}{4}(\mathbf{A}_2 - \mathbf{B}_2) \\ \frac{1}{4}(\mathbf{C}_2 + \mathbf{D}_2) \\ \frac{1}{4}(\mathbf{C}_2 - \mathbf{D}_2) \end{array} \right),$$

$$\mathbf{I}_2 \otimes \mathbf{E}_2 \otimes \mathbf{I}_2 = \left[ \begin{array}{ccc|ccc} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ 1 & & & -1 & & \\ & & & & -1 & \\ \hline & & & & & 1 & & 1 \\ & & & & & & 1 & \\ & & & & & & & 1 \\ & & & & & 1 & & -1 \\ & & & & & & 1 & -1 \end{array} \right],$$

$$\mathbf{A}_2 + \mathbf{B}_2 = \left[ \begin{array}{c|c} \frac{a+f+d+h}{c+g+e+k} & \frac{b+g+e+k}{a+f+d+h} \end{array} \right] \quad (6)$$

$$\mathbf{A}_2 - \mathbf{B}_2 = \left[ \begin{array}{c|c} \frac{a+f-d-h}{c+g-e-k} & \frac{b+g-e-k}{a+f-d-h} \end{array} \right] \quad (7)$$

$$\mathbf{C}_2 + \mathbf{D}_2 = \left[ \begin{array}{c|c} \frac{a-f+d-h}{c-g+e-k} & \frac{b-g+e-k}{a-f+d-h} \end{array} \right] \quad (8)$$

$$\mathbf{C}_2 - \mathbf{D}_2 = \left[ \begin{array}{c|c} \frac{a-f-d+h}{c-g-e+k} & \frac{b-g-e+k}{a-f-d+h} \end{array} \right] \quad (9)$$

Model grafostrukturalny, odzwierciedlający strukturę algorytmiczną procesu obliczeniowego na tym etapie racjonalizacji obliczeń został przedstawiony na rysunku 3.

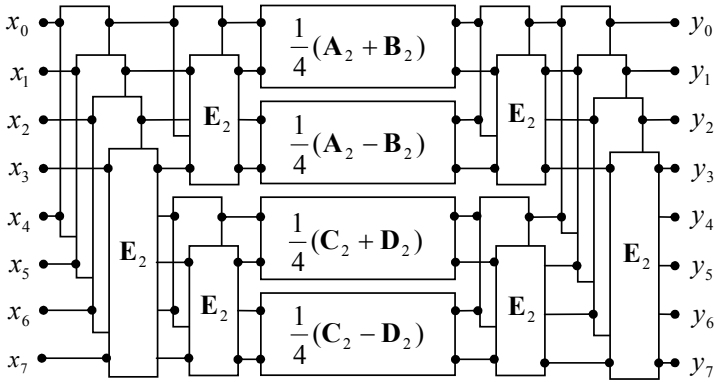
Wreszcie można zauważyć, że macierze (6), (7), (8), (9), wchodzące w skład macierzy  $\mathbf{D}_8^{(2)}$  procedury (5) posiadają kompozycje bloków zgodne z kompozycją o numerze 8 z tabeli 1. Dla tego przypadku wyznaczenie iloczynów odpowiednich podwektorów i oznaczonych macierzy może zostać też zracjonalizowane za pomocą procedury 8 z tabeli 2.

Wówczas otrzymujemy:

$$\mathbf{Y}_{8 \times 1} = (\mathbf{E}_2 \otimes \mathbf{I}_4)(\mathbf{I}_2 \otimes \mathbf{E}_2 \otimes \mathbf{I}_2)(\mathbf{I}_4 \otimes \mathbf{T}_{2 \times 3}) \times \mathbf{D}_{12}^{(3)}(\mathbf{I}_4 \otimes \mathbf{T}_{3 \times 2})(\mathbf{I}_2 \otimes \mathbf{E}_2 \otimes \mathbf{I}_2)(\mathbf{E}_2 \otimes \mathbf{I}_4)\mathbf{X}_{8 \times 1}, \quad (10)$$

gdzie:

$$\mathbf{T}_{3 \times 2} = \begin{bmatrix} 1 & \\ & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{T}_{2 \times 3} = \begin{bmatrix} & 1 & 1 \\ 1 & & 1 \end{bmatrix},$$



Rysunek 3. Model grafostukturalny organizacji procesu obliczeniowego wyznaczenia iloczynu macierzy  $\mathbf{A}_8$  przez wektor  $\mathbf{X}_{8 \times 1}$  po drugim etapie racjonalizacji

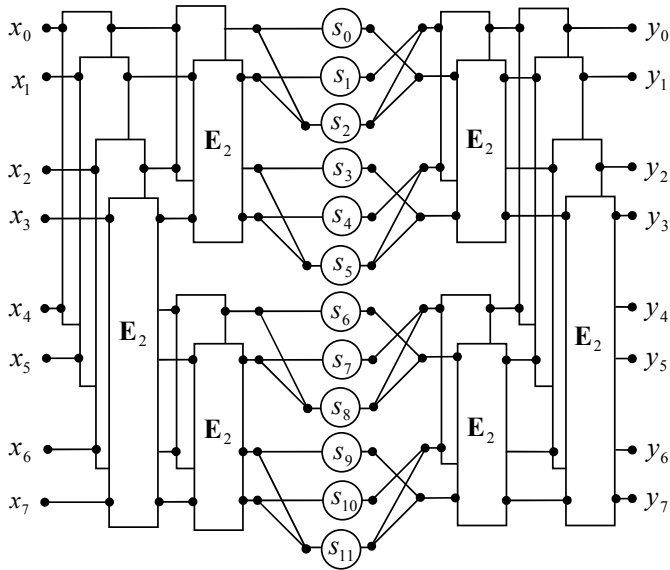
$$\mathbf{D}_{12}^{(3)} = \frac{1}{4} \text{diag}(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}),$$

zaś:

$$\begin{aligned} s_0 &= c + g + e + k - a - f - d - h, \\ s_1 &= b + g + e + k - a - f - d - h, \\ s_2 &= a + f + d + h, \\ s_3 &= c + g - e - k - a - f + d + h, \\ s_4 &= b + g - e - k - a - f + d + h, \\ s_5 &= a + f - d - h, \\ s_6 &= c - g + e - k - a + f - d + h, \\ s_7 &= b - g + e - k - a + f - d + h, \\ s_8 &= a - f + d - h, \\ s_9 &= c - g - e + k - a + f + d - h, \\ s_{10} &= b - g - e + k - a + f + d - h, \\ s_{11} &= a - f - d + h \end{aligned}$$

Na rysunku 4 został przedstawiony model grafostukturalny reprezentujący ostateczną wersję zrationalizowanego procesu obliczeniowego wyznaczenia iloczynu macierzy  $\mathbf{A}_8$  i wektora  $\mathbf{X}_{8 \times 1}$ . W przypadku tego modelu skupienie linie prostych w odpowiednich punktach oznacza operację dodawania, natomiast linie rozchodzące się (rozgałęzienia) – zwykle operacje dublowania danych.

Jak widać syntetyzowany algorytm mnożenia macierzy wartości stałych dla rozpatrywanego przykładu wymaga tylko 12 mnożeń zamiast 64 oraz 44 operacji dodawania zamiast 56.



Rysunek 4. Model grafostukturalny organizacji procesu obliczeniowego wyznaczenia iloczynu macierzy  $A_8$  przez wektor  $X_{8 \times 1}$  po trzecim etapie racjonalizacji

## 5. Podsumowanie

Zaproponowane w pracy strategie poszukiwania racjonalnych rozwiązań przy syntezie przyspieszonych algorytmów mnożenia wektora przez macierz stanowią użyteczny instrument projektowania efektywnych algorytmów cyfrowego przetwarzania sygnałów. Jest to proste i zarazem skuteczne podejście do redukcji liczby operacji arytmetycznych przy wyznaczeniu iloczynów macierzowo-wektorowych w oparciu o specyficzną strukturę tych macierzy. Staje się możliwe wykorzystanie proponowanego podejścia przy syntezie zoptymalizowanych algorytmów dla tych zadań, których istniejące rozwiązania dotychczas nie obejmują. Ciekawostką jest również ten fakt, iż proponowane podejście pozwala powtórzyć wyniki lub też syntetyzować zupełnie inne (posiadające inną strukturę algorytmiczną) szybkie algorytmy dla już rozwiązanych i przedstawionych w literaturze typowych zadań cyfrowego przetwarzania sygnałów. W kolejnych artykułach autor zamierza przedstawić rozmaite przykłady takich algorytmów.

## Bibliografia

- [1] Blahut R. E. *Fast Algorithms for Digital Signal Processing*. Addison-Wesley 1985
- [2] Zieliński T. P. *Cyfrowe przetwarzanie sygnałów*. WKŁ, Warszawa, 2007
- [3] Nussbaumer H.J. *Fast Fourier Transform and Convolution Algorithms*. Springer-Verlag 1982
- [4] Burrus S., Parks T. W., Potts J. F. *DFT/FFT and Convolution Algorithms and Implementation*. John Wiley & Sons 1985

- 
- [5] Tolimieri R., An M., Lu C. *Algorithms for Discrete Fourier Transform and Convolution*. Springer-Verlang, New York, 1989
  - [6] Moon T. K., Stirling W. C. *Mathematical methods and algorithms for signal processing*. Prentice-Hall 2000
  - [7] Dagman E., Kukharev G. *Szybkie dyskretne transformaty ortogonalne*. Nauka, Nowosybirsk, 1983 (w języku rosyjskim)
  - [8] Stasiński R. *O liczeniu dyskretnej transformaty Fouriera*. Wydawnictwo Politechniki Poznańskiej 2005.
  - [9] Țariov A., Adamski M. *Redukcja liczby mnożeń przy wyznaczeniu iloczynów wektora przez macierze wartości stałych drugiego rzędu*. Metody informatyki stosowanej w technice i technologii. Roczniki Informatyki Stosowanej WIPS, Nr 8, Szczecin, Informa, 2005,
  - [10] Țariov A., Adamski M. *Redukcja liczby mnożeń przy wyznaczeniu iloczynów wektora przez macierze wartości stałych  $N$ -tego rzędu*. Metody informatyki stosowanej. Roczniki informatyki stosowanej WIPS, Nr 10, str. 33-41, Szczecin, Informa, 2006.
  - [11] Țariov A. *Modele algorytmiczne i struktury wysokowydajnych procesorów cyfrowej obróbki sygnałów*. Szczecin, Informa, 2001.

# Badania możliwości wyboru optymalnej zmiennej obserwowanej jako zmiennej predykcyjnej w zbiorze danych o rynkach finansowych

Antoni Wiliński, Piotr Czapiewski

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*The paper presents the results of computer simulations performed using the historical quotes on several securities (WIG20, S&P500, Dow Jones, DAX, EUR/USD, gold, oil, etc.) in order to analyse the possibility of finding such variables, that can be explained in terms of the others better, than the rest. It is assumed, that the ultimate goal of every investment strategy is finding the opportunity of gaining a financial profit (always considering the risk). Such opportunity is being sought by investigating the possibility of using each variable (each security) in turn as the one to be predicted. In order to reach that goal, authors use several variants of one of the algorithms belonging to the Group Method of Data Handling (GMDH), namely the combinatorial algorithm. The results reveal some interesting features of regression models, indicating the prospect of further applications of the method.*

## Słowa kluczowe:

*prognostic methods, GMDH, multiple regression, prediction, transaction systems, investment strategies, securities, financial mathematics*

## 1. Wprowadzenie

Wielu, także często wybitnych, teoretyków analizy technicznej rynków finansowych niejednokrotnie przez całe życie zajmuje się doskonaleniem metod prognostycznych tak, że po pewnym czasie celem ostatecznym badań staje się jakość predykcji w szeregach czasowych. Tymczasem w istocie rzeczy doskonalenie metod prognostycznych jest jedynie etapem pośrednim do osiągnięcia celu głównego – zbudowania lepszego narzędzia spekulacji. Idzie więc nie o to, by precyzyjnie prognozować, lecz by umieć dobrą metodę prognostyczną zastosować w strategii inwestycyjnej, umieć zbudować dobry system transakcyjny. Predykcja wieloraka (wielokrotna) jest przykładem takiego podejścia do pośredniej roli dokładności prognozowania. Dokładności, która przestaje być celem samym w sobie, a staje się podstawą do zbudowania dobrej strategii, najlepiej automatycznego handlu. Predykcja wieloraka to umowne określenie swobody wyboru zmiennej objaśnianej spośród wielu innych zmiennych stanowiących rezultat pewnych obserwacji

(pomiarów). Przez analogię do pojęcia regresji wielorakiej [1] zastosowana tu zostanie zasada możliwości wyrażania jednej zmiennej za pomocą innych. Zazwyczaj jest tak, że z góry wiadomo co jest wynikiem jakiegoś eksperymentu, a co (jaki zbiór zmiennych) wymaga obserwacji dla ustalenia statystycznego związku pomiędzy wejściami do obserwowanego systemu i obserwowanym wynikiem. Np. poziom wody w rzece zależy zapewne od m.in. sumy opadów (nie odwrotnie), liczba strażaków gaszących pożar zależy od wielkości pożaru (nie odwrotnie). Ten związek przyczynowo-skutkowy nie zawsze jest tak oczywisty, często jest zupełnie nieoczywisty. Np. czy wartość pary walutowej EUR/USD zależy od ceny ropy, czy odwrotnie?

Na rynkach finansowych występuje interesująca sytuacja w aspekcie, który można byłoby nazwać aksjologicznym. Z punktu widzenia poszukiwania jakiejś wartości (przydatności, korzyści) w zbiorze danych nie będzie miało znaczenia, która zmienna jest w istocie skutkiem, a która przyczyną. Można nawet wyobrazić sobie okresową przemienność tych ról. Może tak być, gdy perturbacje ekonomiczne występują wyraziście na jednym rynku – „ważniejszym” – i wpływają na inny, a po jakimś czasie duża zmienność na rynku przed chwilą traktowanym jako rynek „skutku” może stać się „przyczyną” dla rynku pozornie „ważniejszego”.

Jeżeli więc celem nadrzędnym inwestora interesującego się rynkiem papierów wartościowych jest spekulacja, to zrozumiałą będzie zamiar zbudowania modelu umożliwiającego zarabianie w jakikolwiek sposób na wiedzy wyekstrahowanej z danych. Zamiar pełnego zrozumienia ekonofizycznej istoty (fundamentalnej) wszystkich procesów zachodzących na rynkach będzie wówczas oczywistym etapem pośrednim do osiągnięcia celu końcowego – zarabiania pieniędzy.

Można sobie oczywiście wyobrazić, że ten cel pośredni – model zachowania się rynków – może mieć inne bardzo istotne np. społeczne czy militarne znaczenie, nie jest jednak ono przedmiotem rozważań w niniejszej pracy.

## 2. Charakterystyka danych

Stosując się do powyższego podejścia wybrano  $M = 12$  zmiennych jako podstawową macierz danych, na której przeprowadzono eksperymenty. Na te 12 zmiennych złożyły się najważniejsze, zdaniem autorów, papiery wartościowe współczesnego świata – najważniejsze indeksy giełdowe, pary walutowe, ceny ropy i złota.

Wybór tych a nie innych zmiennych ma tu znaczenie symboliczne. Są to po prostu najważniejsze papiery wartościowe, a celem autorów będzie wykazanie, czy zbiór opartych na nich danych pozwoli na skuteczną spekulację. Wracając do wyrażonej we wprowadzeniu retoryki należałoby powiedzieć, że zbiór papierów wartościowych, w którym poszukuje się możliwości zysku, nie powinien być budowany według takiego kryterium jak „medialna wielkość”. Racjonalny inwestor nigdy nie buduje portfela posługując się takimi pozarynkowymi kryteriami. Korzystniejsze więc byłoby poszukiwanie zmiennych (papierów wartościowych) maksymalizujących prawdopodobieństwo udanej spekulacji – maksymalizujących potencjalną stopę zwrotu z inwestycji.



Wśród tych zmiennych (papierów wartościowych) znalazł się indeks WIG20, nie będący oczywiście jednym z najważniejszych na świecie. Został tu jednak ulokowany z oczywistych powodów subiektywnych – ciekawości badaczy.

Przed wykorzystaniem wspomnianych zmiennych sprawdzono współczynnik korelacji liniowej Pearsona wszystkich par wytypowanych zmiennych. Zakładano, że rozpatrywane zmienne będą skorelowane ze sobą, zwłaszcza zmienne objaśniające z wytypowaną zmienną wyjściową. Zakładano także, że każda z rozpatrywanych zmiennych będzie posiadała odpowiednio dużą własną zmienność (wariancję), gdyż inaczej iluzoryczny byłby jej wpływ na pozostałe zmienne.

Współczynnik korelacji liniowej badano za pomocą formuły:

$$r_{ij} = \frac{\text{cov}(x_i, x_j)}{s^2(x_i) \cdot s^2(x_j)} \quad (1)$$

gdzie:

$\text{cov}(x_i, x_j)$  – kowariancja wybranej pary zmiennych,  
 $s^2(x_i)$  – wariancja zmiennej  $x_i$ ,  $x_i \in \mathbf{X}$  ( $i = 1, 2, \dots, M$ ).  
 Zmienność własną badano jako:

$$v_i = \frac{s(x_i)}{\bar{x}_i} \quad (2)$$

gdzie:

$s(x_i)$  – odchylenie standardowe zmiennej  $x_i$ ,  
 $\bar{x}_i$  – średnia arytmetyczna zmiennej  $x_i$ .

Rozpatrywany zbiór 12 zmiennych obejmował notowania następujących instrumentów finansowych:

1. Indeks giełdy we Frankfurcie DAX.
2. Para walutowa EUR/USD (kurs euro wyrażony w dolarach amerykańskich).
3. Indeks giełdy w Londynie FTSE 100.
4. Para walutowa GBP/USD (kurs funta brytyjskiego wyrażony w dolarach amerykańskich).
5. Złoto (cena jednej uncji troy złota wyrażona w dolarach amerykańskich).
6. Indeks giełdy w Hong Kongu HAI (*Hang Seng Index*).
7. Indeks giełdy w Tokio Nikkei 225.
8. Ropa naftowa (*Brent Crude Oil*).
9. Warszawski indeks giełdowy WIG 20.
10. Amerykański indeks giełdowy DJIA (*Dow Jones Industrial Average*).
11. Amerykański indeks giełdowy S&P 500.
12. Amerykański indeks giełdowy NASDAQ-100.

Wykorzystane dane historyczne pochodzą z zasobów firmy MetaQuotes Software Corp. i zostały pobrane za pośrednictwem platformy MetaTrader<sup>10</sup>.

Przygotowano  $N = 626$  wierszy macierzy danych  $\mathbf{X}$ . Każda kolumna zawierała jednodobowe wartości *Close* świecy danego papieru wartościowego. Przykład stosowanego formatu i dokładności tych zmiennych przedstawiono w tabeli 1.

<sup>10</sup> Zobacz <http://www.metaquotes.net>

Tabela 1. Fragment macierzy danych, którą zastosowano do predykcji wielokrotnej

	DAX	EUR/USD	FTSE 100	GBP/USD	GOLD	HAI	NIKKEI 225	OIL	WIG 20	DJIA	S&P 500	NASDAQ
1	4350	1,2920	4917	1,8741	425,70	13412	11600	53,10	1970	10404	1163,0	1462
2	4346	1,2914	4906	1,8787	425,90	13426	11566	51,70	1980	10539	1179,0	1490
3	4349	1,2962	4897	1,8904	427,90	13517	11669	54,07	1998	10521	1181,0	1484
4	4365	1,2905	4912	1,8816	425,65	13491	11724	55,43	1989	10388	1168,0	1466
5	4340	1,2853	4899	1,8762	424,05	13513	11668	56,20	1985	10417	1174,0	1476
...	...	...	...	...	...	...	...	...	...	...	...	...
626	7479	1,4648	5954	1,9616	882,00	24410	13481	88,64	2964	12513	1377,8	1875

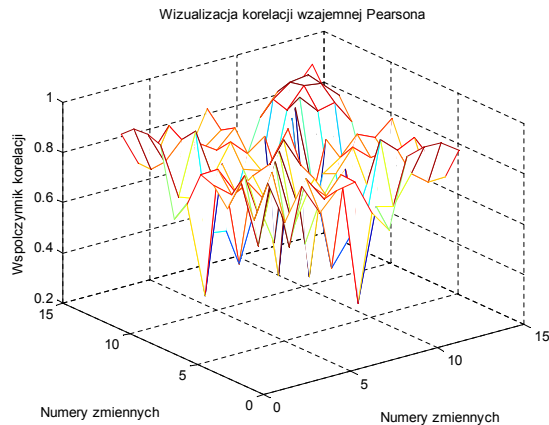
Każdy wiersz tej macierzy był skrupulatnie sprawdzony pod kątem jednoczesności zdarzeń – notowania poszczególnych instrumentów zostały zsynchronizowane. W przypadku, gdy dla co najmniej jednej zmiennej brakowało danych z pewnego dnia, cały wiersz danych był usuwany. Wynikające z tego luki informacyjne nie są jednak zbyt częste i nie umniejszają praktycznej stosowalności opisywanej metody. W przyszłości należałoby rozważyć inne sposoby wypełniania luk informacyjnych, jak choćby uzupełnianie brakujących danych poprzedzającymi wartościami szeregu czasowego lub średnią arytmetyczną wartości sąsiadujących.

Macierz  $\mathbf{X}$  może być rozpatrywana jako zbiór ułożonych kolumnowo szeregów czasowych [8]. Zawiera ona wartości pobrane ze świec dobowych, a więc obejmuje dość znaczny okres rzędu 30 miesięcy – od marca 2005 do stycznia 2008.

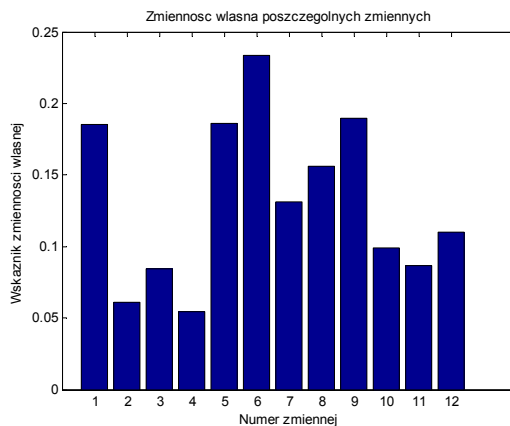
Wyniki badań zmienności i korelacji wzajemnej przeprowadzono w środowisku Matlab. Współczynnik korelacji liniowej Pearsona przedstawiono na rys. 1.

Z wykresu (rys. 1) wynika, że współczynnik korelacji pomiędzy poszczególnymi parami zmiennych waha się od ok. 0,3 do ponad 0,9 (na przekątnej wynosi oczywiście 1,0). To duża i bardzo duża korelacja, charakterystyczna dla rynków finansowych. Nie przesądza ona w tej chwili o niczym. Negatywnym wynikiem byłaby raczej bardzo niska korelacja pomiędzy parami zmiennych, ujawniająca całkowity brak relacji i przypadkowość zmian obu badanych czynników.

Na wykresie (rys. 2) przedstawiono wyniki badania zmienności własnej, obliczonej zgodnie z (2). Pożądaną wysoką zmienność rozpatrywanych czynników zaobserwowano u większości zmiennych. Jedynym wyjątkiem z wartością  $v$  mniejszą od 0,1 są tu zmienne o indeksach 2, 3 i 4 (to kolejno: para walutowa EUR/USD, londyński indeks FTSE 100, para walutowa GBP/USD). Do możliwości prognozy tych trzech czynników należałoby podejść ostrożniej, o ile zbudowane modele dałyby interesujące rezultaty. Pozostałe wyniki wstępnych testów nie wykluczają ewentualności pomyślniej prognozy.



Rysunek 1. Wizualizacja macierzy współczynników korelacji liniowej



Rysunek 2. Badania zmienności własnej czynników rozpatrywanych w macierzy danych – papierów wartościowych

### 3. Zastosowane algorytmy kombinatoryczne GMDH

Zgodnie z zamiarem autorów, do prognozy wielorakiej wybranej grupy czynników (papierów wartościowych) zastosowano proste stosunkowo algorytmy kombinatoryczne z rodziny algorytmów należących do metody GMDH.

Istotą metody grupowania argumentów jest zasada samoorganizacji modelu opisującego rzeczywistość. Przez tę samoorganizację rozumie się najczęściej zdolność systemu odpowiedzialnego za automatyczne dążenie do celu do adaptacyjnego

określania struktury modelu najlepiej (w sensie pewnego kryterium) opisującego obserwowany proces.

W rozpatrywanym przypadku celem systemu jest predykcja zdarzeń na rynkach finansowych – to cel bezpośredni, bliższy. Celem dalszym, pośrednim, osiąganym poprzez realizację celu bliższego, jest zysk osiągany w pewnym systemie transakcyjnym za pomocą strategii zbudowanej na metodzie predykcji zdarzeń.

Przez model rozumie się pewien wielomian (jakikolwiek, prosty lub złożony z innych elementów) szeregu Gabora [2, 3, 4]:

$$y = a_0 + \sum_{i=1}^M a_i x_i + \sum_{i=1}^M \sum_{j=1}^M a_{ij} x_i x_j + \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M a_{ijk} x_i x_j x_k + \dots \quad (3)$$

gdzie:

$y$  – zmienna wyjściowa (objaśniana),  
 $x_i$  –  $i$ -ta zmienna wejściowa (objaśniająca),  
 $a_i, a_{ij}, a_{ijk}$  – współczynniki wielomianu, estymowane w procesie uczenia modelu.

Jest on modelem wówczas, gdy przedstawiająca go zmienna  $\hat{y}$  jest odnoszona do rzeczywiście zaobserwowanych czy pomierzonych wartości zmiennej objaśnianej  $y$ .

Szereg ten (3) jest oczywiście nieskończony i heurystyczne podejście badacza sprowadzi się do takiej selekcji modeli cząstkowych (opisywanych przez niektóre tylko wyrazy szeregu Gabora), by rozwiązać problem predykcji w rozsądnym czasie (maszynowym). Ten czas ma zresztą także inny aspekt praktyczny. Czas obliczeń może mieć bezpośredni związek z systemem transakcyjnym. Jeżeli byłby zbyt długi dla pewnego wybranego wariantu strategii (np. systemu zbudowanego na bardzo krótkich świecach), to cała metoda byłaby nieprzydatna, gdyż nie spełniałaby warunku jej realizacji w czasie rzeczywistym.

Sytuacja taka to dowód na to, że technologia (tu oferta platformy brokerskiej) ogranicza ideę matematyczną, o czym pisze np. Jajuga [5].

Tak więc, jeżeli będzie ustalona w ten sposób dopuszczalna złożoność modelu jako pewnej instancji szeregu Gabora, to można przystąpić do analizy dokładności tych modeli i ich przydatności predykcyjnej.

Uwaga o ograniczeniu złożoności przez rozsądny czas obliczeń nie będzie raczej dotyczyła algorytmu zbudowanego dla omawianych tu danych. Rozpatrywane są bowiem świece jednodniowe, co oznacza, że w ewentualnym systemie transakcyjnym byłaby teoretycznie doba na wykonanie obliczeń i podjęcie decyzji.

Należy jednak uzmysłwić sobie, że złożoność modelu wybranego z szeregu Gabora może być tak znaczna, że i ten czas (tu – doba, wspomniano o świecach jednodobowych) nie będzie wystarczający w sposób oczywisty. Rozważania takie mają jednak sens tylko wtedy, gdy dowiedzie się istnienia związku pomiędzy rosnącą złożonością modelu a jego przydatnością predykcyjną. Twórca metody Ivakhnenko [2] (także wielu jego następców) zauważył, że występuje optimum złożoności modelu, powyżej którego zaszumienie modelu pogarsza precyzję odtwarzania (lub prognozowania) rzeczywistości.

Uwzględniając powyższe zbadano dwie klasy modeli (3) – liniowe i nieliniowe.

Modele liniowe miały postać:

$$\hat{\mathbf{Y}}_1 = \begin{bmatrix} 1 & \mathbf{x}_i & \mathbf{x}_j & \mathbf{x}_k & \mathbf{x}_l \end{bmatrix} \cdot \mathbf{A}_1 \quad (4)$$

lub inaczej:

$$\hat{\mathbf{Y}}_1 = \mathbf{X}_w \cdot \mathbf{A}_1 \quad (5)$$

gdzie:

$\mathbf{X}_w$  – to macierz danych utworzona z wybranych (tu czterech) wektorów (kolumn) macierzy wszystkich danych  $\mathbf{X}$ . Do tych czterech wybranych wektorów nie mogą wchodzić dane z pierwszej kolumny, gdyż reprezentuje ona zmienną obserwowaną (umownie). W związku z tym:

$$\mathbf{X}_w = \begin{bmatrix} 1 & \mathbf{x}_i & \mathbf{x}_j & \mathbf{x}_k & \mathbf{x}_l \end{bmatrix} \quad (6)$$

Każda zmienna  $\mathbf{x}_m$ ,  $m \in \{i, j, k, l\}$  to wektor kolumnowy o długości  $L$ , gdzie  $L$  to liczba obserwacji wzięta pod uwagę przy obliczaniu współczynników liniowego równania regresji opisującego zmienną obserwowaną  $y$ .

$\mathbf{1}$  – to kolumna jedynek rozszerzająca podmacierz danych  $\mathbf{X}_w$ , po to, by równanie regresji liniowej posiadało wyeksponowany tzw. wyraz wolny, nie związany z żadnym czynnikiem  $\mathbf{x}_m$ .

$\mathbf{A}_1$  – to wektor (kolumnowy) współczynników liniowego równania regresji, zawierający także wyraz wolny  $a_0$ .

W ostatecznej więc postaci liniowy model (wybrany z szeregu Gabora) będzie miał postać:

$$\hat{y} = a_0 + a_1 x_i + a_2 x_j + a_3 x_k + a_4 x_l \quad (7)$$

Współczynniki  $\mathbf{A}_1$  pozwalające na utworzenie powyższego modelu są otrzymywane poprzez zastosowanie mechanizmu regresji na danych  $\mathbf{X}_w$  z rzeczywiście obserwowanymi wynikami wybranej zmiennej  $y$  w taki sposób, że

$$\mathbf{Y} = \mathbf{X}_w \cdot \mathbf{A}_1 \quad (8)$$

$$\mathbf{A}_1 = \text{pinv}(\mathbf{X}_w) \cdot \mathbf{Y} \quad (9)$$

Rozwiązanie równania (8) ze względu na  $\mathbf{A}_1$  wymaga odwrócenia macierzy  $\mathbf{X}_w$ . Ponieważ w równaniu tym macierz  $\mathbf{X}_w$  nie jest macierzą kwadratową (ma wymiary  $L \times 5$ ), nie jest możliwe jej odwrócenie. Równanie to jest rozwiązywane z zastosowaniem tzw. pseudoinwersji Moore'a-Penrose'a [6] (oznaczanej tu operatorem  $\text{pinv}$ ), która daje przybliżone rozwiązanie i spośród wielu możliwych wskazuje najlepsze w sensie metody najmniejszych kwadratów.

Przez rozwiązanie najlepsze rozumie się tu najmniejszą długość wektora  $\|\mathbf{Y} - \mathbf{X}\mathbf{A}\|^2$ , czyli pewnego kryterium:

$$C = \left\| \mathbf{Y} - \hat{\mathbf{Y}} \right\|^2 \rightarrow \min \quad (10)$$

Zarówno model  $\hat{Y}$  jak i oceniające go kryterium  $C$  może być wynikiem jednej z wielu prób (kombinacji) takiego doboru kolumn  $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l\}$ , które wyznaczają lepsze i gorsze modele  $\hat{Y}^k$ ,  $k = 1, 2, \dots, K$ , gdzie  $k$  to numer kolejnej symulacji, kolejnego  $k$ -go rozpatrywanego szeregu czasowego.

Niech  $p$  oznacza kolejny numer próby wykonywanej z  $k$ -tym szeregiem czasowym, a rozsądek badacza wyznacza liczbę dopuszczalnych badanych modeli jako  $P$ . Dla każdego  $k$ -tego szeregu poszukiwany jest więc taki model  $\hat{Y}_{opt}$ , dla którego indeks  $p$  (w sensie kolejno rozpatrywanej kombinacji) jest taki, że:

$$p_{opt} = \arg \min_p C^p \quad (11)$$

gdzie

$$C^p = \frac{1}{L} \sum \left\| \mathbf{Y} - \hat{\mathbf{Y}}^p \right\|^2 \quad (12)$$

dla  $p = 1, 2, \dots, P$ .

$P$  jest właśnie tą liczbą, tym ograniczeniem heurystycznym, które daje nadzieję na znalezienie dobrego rozwiązania w rozsądnym czasie. Jej wyznaczenie jest najczęściej syntezą doświadczenia, intuicji, rozsądku i oczekiwań badacza. Wymienione czynniki z trudem (lub w ogóle) dają się opisać formalnie. Podkreślają z pewnością indukcyjny charakter rozważanego algorytmu.

W przypadku rozpatrywanych modeli liniowych liczba  $P$  może być wyznaczana po prostu jako liczba kombinacji  $P = \binom{M-1}{4}$  i dla rozpatrywanych tu danych ( $M = 12$ ) nie jest liczbą uniemożliwiającą rozważenie wszystkich kombinacji. Inaczej będzie w przypadku modeli nieliniowych.

Wybór takiej, a nie innej liczby wyrazów w modelu liniowym (4) jest rezultatem arbitralnie podjętej decyzji. Jest to po prostu świadomy (jednak nie przypadkowy) wybór takiej postaci modeli. Zbudowanie modeli na części dostępnych danych zapewnia pewną równowagę pomiędzy dokładnością modelu w części „uczącej”, tam gdzie model poprzez regresję „uczy się” współczynników, a częścią „testową”, gdzie następuje predykcja. Wzięcie do modelu większej liczby wyrazów np. 5, 6 czy więcej spowoduje lepsze opisanie przeszłości i niekoniecznie lepszą predykcję. Zazwyczaj mówi się wówczas o „przeuczeniu” modelu. Wstępne autorskie testy spowodowały, że można było wcześniej napisać, że dobór modelu nie jest przypadkowy.

W rozbudowanych systemach GMDH, tu nie omawianych, często tworzy się, np. w formie lokalnej sieci neuronowej, moduł rozpoznający optymalną w danym kroku strukturę modelu opisującego zmienną objaśnianą. W rozpatrywanym tu przypadku oznaczałby automatyczne rozpoznawanie liczby wyrazów w modelu (4). W dalszej części opracowania, z uwagi na jego objętość, przedstawione będą wyniki badań wyłącznie modelu liniowego.

#### 4. Liniowy model regresji w predykcji wielorakiej

Opisana na wstępie macierz obserwowanych danych o najważniejszych papierach wartościowych zawiera  $M = 12$  kolumn i kilkaset wierszy. Postawiona teza,

że być może są takie zmienne (kolumny), które dadzą się dobrze opisać za pomocą pozostałych (w modelu liniowym za pomocą wybranych najlepszych czterech) wymaga sprawdzenia i statystycznego potwierdzenia.

Niech  $\mathbf{Y}$  będzie tożsamy z  $\mathbf{x}_s$ , gdzie  $\mathbf{x}_s$  jest  $s$ -tą kolumną w macierzy  $\mathbf{X}$ ,  $s = 1, 2, \dots, M$ , i będzie ustawiony na miejscu pierwszej kolumny, a kolumna pierwsza zajmie miejsce  $s$ . Tak zmodyfikowana macierz danych  $\mathbf{X}_s$  stanowi wyjściowy materiał badawczy do weryfikacji jakości modelu predykcyjnego. To pewna konwencja organizacji obliczeń – zawsze na miejscu pierwszej kolumny w macierzy danych będzie znajdowała się zmienna pełniąca funkcję zmiennej objaśnianej w danym wariancie obliczeń.

$$\hat{\mathbf{Y}}_s = f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_M), \quad i \neq s, \quad \text{dla } s = 1, 2, \dots, M \quad (13)$$

Powstaje w ten sposób  $M = 12$  systemów predykcyjnych, z których każdy obliczany jest według algorytmu opisanego poprzez (7) do (12), czyli każdy najlepszy model w systemie wybierany jest według kryterium (11) spośród wielu modeli charakterystycznych dla danej struktury przestrzeni danych  $\mathbf{X}_s$ .

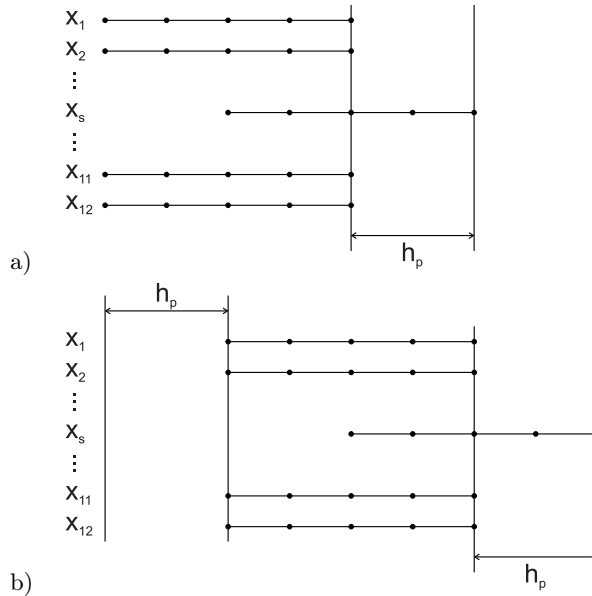
W kryterium (11)-(12) pozornie nie widać jego predykcyjnego charakteru, o ile nie wyjaśni się wzajemnego przesunięcia wprzód o horyzont predykcyjny  $h_p$  kolumny pierwszej (zmiennej obserwowanej) w stosunku do kolumn pozostałych. Jest to zabieg charakterystyczny dla całej idei GMDH stosowanej w celu predykcyjnym. Polega on na wykonaniu regresji i obliczeniu współczynników dla sytuacji jak na rys. 3a, a następnie obliczeniu kryterium jakości predykcji w sytuacji jak na rys. 3b.

Na rysunku symbolicznie zaznaczono wybraną zmienną  $\mathbf{x}_s$  jako ułożoną wewnątrz macierzy danych, przed jej przemieszczeniem na pozycję pierwszej kolumny (macierz danych na rysunku przedstawiona jest po jej „położeniu” – kolumny są tu poziomo).

Na rys. 3b widać więc predykcyjny charakter obliczeń dla zmiennej  $\mathbf{x}_s$  (w poprzedniej ogólnej notacji posiadającej symbol zmiennej obserwowanej  $y$ ). Zmienna ta wyprzedza bowiem chwilę bieżącą o horyzont predykcyjny  $h_p$ .

Podsumowując – dla jednego systemu danych (dla wybranego indeksu  $s$  zmiennej obserwowanej) należy kolejno wykonać następujące kroki:

1. Utworzyć chwilową macierz danych  $\mathbf{X}_s$ , wybierając z obserwowanych szeregów czasowych  $\mathbf{x}_i$ ,  $i = 1, 2, \dots, 12$ ,  $i \neq s$  wektory o długości  $L_w$ , rozpoczynające się  $L_w + h_p$  przed chwilą bieżącą, a kończące się  $h_p$  przed chwilą bieżącą.
2. Utworzyć wektor zmiennej obserwowanej  $\mathbf{x}_s$  o długości  $L_w$  kończący się w chwili bieżącej.
3. Obliczyć współczynniki modelu według (9); tu  $\mathbf{A}_1 = \text{pinv}(\mathbf{X}_s) \cdot \mathbf{x}_s$ .
4. Utworzyć nową chwilową macierz danych  $\mathbf{X}_s^o$  z wykorzystaniem wszystkich dostępnych danych pomiarowych, czy obserwowanych, czyli macierz złożoną z wektorów długości  $L_w$ , ale kończących się w chwili bieżącej (oczywiście do tej macierzy nie wchodzi wybrana zmienna  $\mathbf{x}_s$ ).
5. Obliczyć model predykcyjny  $\mathbf{x}_s$  przesunięty o  $h_p$  do przodu według (5) czyli tu  $\hat{\mathbf{x}}_s = \mathbf{X}_s^o \cdot \mathbf{A}_1$ .



Rysunek 3. Przesunięcie  $M - 1$  (z wyjątkiem  $\mathbf{x}_s$ ) szeregów czasowych o  $h_p$  wstecz (a) oraz przesunięcie ich do chwili bieżącej  $t_0$  dla obliczenia predykcji zmiennej  $\mathbf{x}_s$  (b)

- Obliczyć kryterium rozbieżności modelu  $\hat{\mathbf{x}}_s$  i rzeczywistości (po upływie odpowiedniego czasu lub po pobraniu informacji z danych historycznych). Kryterium to może dotyczyć rozbieżności obserwowanej tylko na odcinku predykcyjnym (lepiej) lub na całej długości wektorów  $\mathbf{x}_s$  i  $\hat{\mathbf{x}}_s$ .

Powyższy, złożony z 6 kroków algorytm należy powtarzać dla każdego  $s = 1, 2, \dots, 12$  po to, by osiągnąć cel postawiony na wstępie niniejszego opracowania – wybrać zmienną (lub kilka), która jest dobrze opisywana przez pozostałe.

## 5. Wyniki eksperymentu

Eksperymenty dla modelu liniowego przeprowadzono w różnych miejscach szeregu czasowego, dla różnych wartości  $L_w$  (np. od 10 do 100), dla różnych wartości horyzontu predykcyjnego  $h_p$  (np. od 1 do 10).

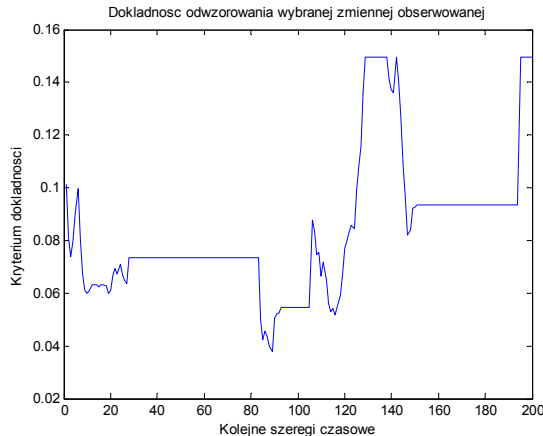
Dla  $L_w = 40$  i  $h_p = 2$  uzyskano wyniki przedstawione na rys. 4, 5, 6. W badaniach odniesiono się także do praktycznej przydatności prognoz w możliwym do implementacji systemie transakcyjnym – opisanym nieco dalej.

Indeks  $s = 9$  oznacza wybór indeksu WIG20 jako zmiennej obserwowanej. Rezultaty dotyczą więc początkowo wyłącznie tej zmiennej.

Na rys. 5 przedstawiono wyniki symulacji dokładności szacowanej według kryterium (10) określającej rozbieżność pomiędzy najlepszym modelem a rzeczywistym przebiegiem zmiennej obserwowanej na całym odcinku.



Badania wykonano dla  $s = 9$ , czyli dla pojedynczej wybranej zmiennej obserwowanej, którą był WIG20. Wykonano je dla 200 kolejnych szeregów czasowych. Kryterium dotyczy rozbieżności pomiędzy zmiennymi ( $\hat{y}$  i  $y$ ) skonwertowanymi do wartości względnych – po normalizacji takiej, że  $\max(y) = 1$  a  $\min(y) = 0$ .

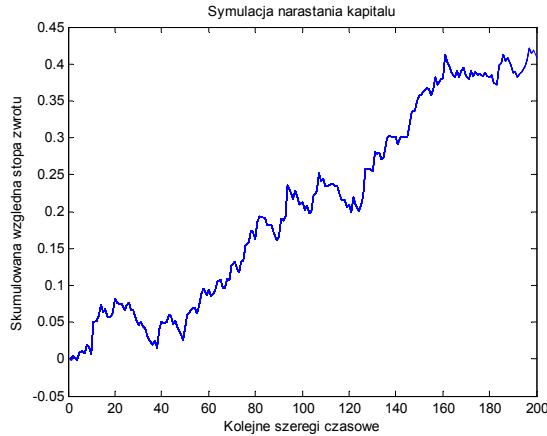


Rysunek 4. Symulacja dokładności odwzorowania szeregów czasowych dla kolejnych 200 przypadków

Wyniki przedstawione na rys. 4 wskazują na dość dobrą dokładność odtwarzania. Ivakhnenko [2] uważa, że dokładność rzędu 10% jest bardzo dobrym wynikiem, zwiastującym sukcesy prognostyczne.

Wykorzystując tę wstępną pozytywną ocenę zbudowano koncepcję systemu transakcyjnego i przeprowadzono symulację narastania kapitału. Wynik tej symulacji przedstawia rys. 5.

Wyniki te są na pierwszy rzut oka całkowicie satysfakcjonujące. Przy zauważalnej wariancji wartość średnia zwrotu jest wyraźnie dodatnia, co skutkuje dość systematycznym wzrostem kapitału. Dla wykresu przedstawionego na rys. 5 średnia wartość stopy zwrotu w każdym z 200 kroków wyniosła 0,0020 przy odchyleniu standardowym 0,0107. Stopa zwrotu rejestrowana co dobę i dodawana do dotychczasowego wyniku (skumulowana) obliczana jest w przestrzeni względnych wartości – tu jako ułamek maksymalnej wartości indeksu WIG20 odnotowanej w dostępnych i wykorzystywanych tu danych. Tak więc obserwowany przyrost rzędu kilku dziesiątych (maksymalnie ok. 0,4) to przyrost rzędu 40% (1600 jednostek indeksu WIG20) od wartości maksymalnej WIG20, która wyniosła dla badanego okresu 3933. Wynik ten uzyskany został dla  $k=200$  odcinków czasowych (tu 200 dni). W dziedzinie rzeczywistych zmian tego indeksu średni przyrost dobowy dla wyników względnych to ok. 8 jednostek WIG20 przy standardowym odchyleniu rzędu 42 jednostek.



Rysunek 5. Symulacja narastania kapitału w systemie transakcyjnym opartym na predykcji krótkoterminowej indeksu WIG20

Strategia inwestycyjna, która doprowadziła do takiego przebiegu skumulowanej wartości obserwowanego indeksu opierała się na kilku założeniach.

Po pierwsze – wprowadzono powszechnie stosowany w platformach brokerskich mechanizm ograniczania straty SL (*Stop Loss*). Przyjęto, że SL będzie równe 1% wartości maksymalnej indeksu, czyli ok. 40 jednostek WIG20.

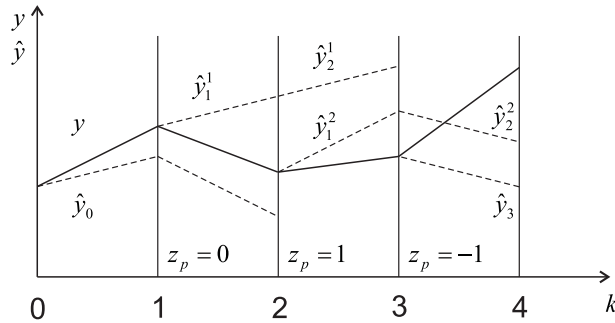
Następnym warunkiem postawionym w strategii był sposób przewidywania kierunku zmian zmiennej obserwowanej (w górę lub w dół) oraz potwierdzenie tego kierunku przez poprzednią prognozę.

Sytuację tę przedstawiono na rys. 6. Na rysunku tym dla każdego początku symulacji  $k = 0, 1, 2 \dots K$  (przeprowadzono  $K = 200$  takich symulacji) dokonano prognozy dwa kroki wprzód  $y_1^k$  i  $y_2^k$ . Przez  $y_1^k$  rozumiano prognozę wartości  $y$  po jednym odcinku prognostycznym zaczynającym się od  $k$ -go kroku symulacji. Rzeczywiste wartości zmiennej obserwowanej oznaczano  $y_1, y_2, \dots, y_k, \dots$ .

W praktycznych implementacjach systemów transakcyjnych bardziej istotna jest trafność przewidywania kierunku zmian niż trafność oceny rzeczywistych wartości stopy zwrotu. Jeżeli bowiem poprawnie byłby przewidziany kierunek zmian, to przy wielokrotnie wyższym rzeczywistym odchyleniu (ale w prognozowanym kierunku) wielokrotnie wyższa będzie wygrana.

W aspekcie zarządzania ryzykiem inwestycyjnym, taka pomyłka nie prowadzi więc do frustracji. W przypadku pomyłki co do znaku zmian obserwowanej zmiennej (np. otwarcie pozycji *long* zamiast *short*) wynik będzie wyłącznie stratą. Z tych powodów stratę tę próbuje się ograniczyć wspomnianym mechanizmem *Stop Loss*.

W sensie matematycznym zgodność prognozy co do kierunku zmiany zmiennej obserwowanej oznacza badanie pierwszych pochodnych  $y$  i  $\hat{y}$  na odcinku prognostycznym  $\Delta t - \frac{dy}{dt}$  i  $\frac{d\hat{y}}{dt}$  lub w układzie dyskretnym  $\frac{y_2^k - y_1^k}{\Delta t}$  i  $\frac{\hat{y}_2^k - \hat{y}_1^k}{\Delta t}$ .



Rysunek 6. Przebiegi rzeczywiste ( $y$ ) i modelowane ( $\hat{y}$ ) zmiennej obserwowanej. Odcinek  $\hat{y}_1^k$  oznacza pierwszy odcinek horyzontu prognostycznego, a  $\hat{y}_2^k$  - drugi.

Niech pewna zmienna  $z(y)$  określa zgodność znaku (sign) zmiennej prognozowanej i rzeczywistej.

$$z(y^k) = \begin{cases} 1 & \text{gdy } \text{sign} \left( \frac{y_2^k - y_1^k}{\Delta t} \right) = \text{sign} \left( \frac{\hat{y}_2^k - \hat{y}_1^k}{\Delta t} \right) \\ 0 & \text{w przeciwnym przypadku} \end{cases} \quad (14)$$

Zmienna przyjmuje więc wartość jeden wtedy, gdy prognozowany kierunek zmian wyrażony poprzez modelowaną stopę zwrotu ( $\hat{y}_2^k - \hat{y}_1^k$ ) pokrywa się z kierunkiem rzeczywistych zmian.

W systemie transakcyjnym dodatnia wartość ( $\hat{y}_2^k - \hat{y}_1^k$ ) sugeruje otwarcie pozycji długiej - zmienna leżąca w szeregu czasowym później  $\hat{y}_2^k$  jest większa niż wcześniejsza  $\hat{y}_1^k$ .

Jeżeli później (w warunkach rzeczywistych lub w danych historycznych) wystąpi także dodatnia różnica ( $y_2^k - y_1^k$ ) to znaki obu różnic będą jednakowe (tu dodatnie) a  $z(y^k)$  będzie równa jeden. Wystąpi zgodność prognozy z rzeczywistością i wygrana w systemie transakcyjnym.

Podobnie - dla sytuacji, gdy obie, modelowana i rzeczywista stopa zwrotu będą ujemne. Oznaczać to będzie otwarcie pozycji krótkiej zakończone sukcesem. Prawdopodobieństwo wystąpienia  $z(y^k) = 1$ , dla  $k = 1, 2, \dots$  może być miarą skuteczności prognozy i jej praktycznej przydatności w systemie transakcyjnym.

W rozważaniach przedstawionych na rys. 6 rozpatrywano także inną zmienną opisującą jakość prognozy -  $z_p(y^k)$ . Zmienna ta zawiera w sobie element potwierdzenia prognozy bieżącej przez prognozę poprzedniego kroku.

$$z_p(y^k) = \begin{cases} 1 & \text{gdy } \text{sign} (y_2^k - y_1^k) = \text{sign} (\hat{y}_2^k - \hat{y}_1^k) \wedge \\ & \text{sign} (y_2^k - y_1^k) = \text{sign} (\hat{y}_3^{k-1} - \hat{y}_2^{k-1}) \\ 0 & \text{gdy } \text{sign} (y_2^k - y_1^k) \neq \text{sign} (\hat{y}_3^{k-1} - \hat{y}_2^{k-1}) \\ -1 & \text{gdy } \text{sign} (y_2^k - y_1^k) \neq \text{sign} (\hat{y}_2^k - \hat{y}_1^k) \wedge \\ & \text{sign} (y_2^k - y_1^k) = \text{sign} (\hat{y}_3^{k-1} - \hat{y}_2^{k-1}) \end{cases} \quad (15)$$

Zastosowanie reguł (15) do przykładowych przebiegów rzeczywistych ( $y$ ) i modelowanych ( $\hat{y}$ ) zmian przedstawiono na rys. 6. W dolnej części rysunku przedstawiono wartości  $z_p$  ustalone według (15).

Rozważane są tam kolejno trzy różne przypadki:

- gdy drugi odcinek predykcyjny rozpoczynającej się w  $y_0$  prognozy sugeruje pozycję krótką, a pierwszy odcinek prognozy rozpoczynającej się w  $y_1$  – pozycję długą, to  $z_p = 0$ , gdyż jest rozbieżność prognoz; system transakcyjny nie wykonuje żadnej akcji;
- gdy drugi odcinek predykcyjny prognozy rozpoczynającej się w  $y_1$  wskazuje na pozycję długą, a także potwierdza to pierwszy odcinek prognozy rozpoczynającej się w  $y_2$ , to przy zgodności z rzeczywistym przebiegiem wystąpi  $z_p = 1$ ; system transakcyjny otworzy pozycję długą i odniesiony zostanie sukces;
- gdy drugi odcinek predykcyjny modelu rozpoczynającego się w  $y_2$  wskazuje na pozycję krótką i potwierdzony jest przez pierwszy odcinek modelu z  $y_3$ , a rzeczywisty przebieg jest przeciwny, to wystąpi sytuacja  $z_p = -1$ . System transakcyjny otworzy pozycję krótką i wystąpi porażka. Wielkość straty jest w tym systemie ograniczana przez mechanizm SL.

Dla tak zdefiniowanych funkcji  $z_p(y^k)$  i  $z(y^k)$  przeprowadzono badania w tej samej przestrzeni danych, dla której przeprowadzono symulację przedstawioną na rys. 4 i 5.

Dla rozpatrywanych 200 szeregów czasowych, każdy o długości  $L_w = 40$  uzyskano dla zmiennej  $z$  według (14) wynik  $\sum_k z(y^k) = 108$  (tj. 54% trafnych prognoz), a dla zmiennej  $z_p$  według (15) wynik  $\sum_k z_p(y^k) = 14$ .

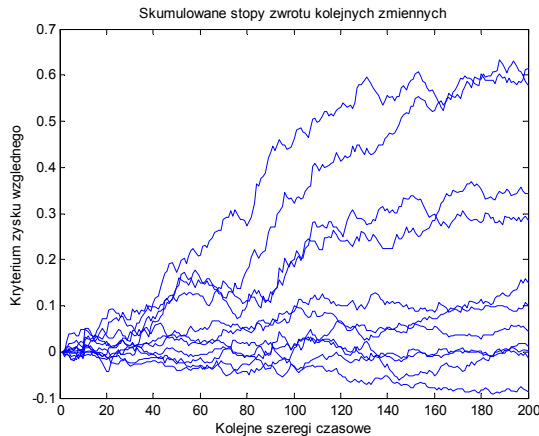
Oznacza to, że uzyskano przewagę zwycięstw nad porażkami wynoszącą 14 zdarzeń (na 200 badanych). Przy zastosowaniu mechanizmu *Stop Loss* uzyskano narastanie kapitału jak na rys. 5.

Kolejnym i najistotniejszym rozpatrywanym tu zadaniem było sprawdzenie przydatności prognostycznej różnych zmiennych obserwowanych, wybieranych kolejno z tej samej macierzy danych. Zadanie to jest określone jako wspomniana predykcja wieloraka. Stawia się tu bowiem hipotezę, że w pewnej skończonej przestrzeni danych pomiarowych mogą być lepsze i gorsze zmienne (kolumny w macierzy danych) w aspekcie ich wykorzystania jako pewnego obiektu opisywanego za pomocą pozostałych danych.

Badania te przeprowadzono dla opisanych wcześniej 12 zmiennych charakteryzujących wybrane papiery wartościowe.

Stosując te same zasady obliczania predykcji potwierdzonej i ten sam sposób obliczania skumulowanej wartości stopy zwrotu co poprzednio dla pojedynczego przypadku (WIG20), na rys. 7 przedstawiono równocześnie narastanie kapitału wszystkich zmiennych.

Tak jak poprzednio dla badań pojedynczej wyróżnionej zmiennej (WIG20) zastosowano tu mechanizm SL (*Stop Loss*) zawsze jednakowy dla każdej rozpatrywanej zmiennej obserwowanej ustalony na poziomie 1% wartości maksymalnej tej zmiennej.



Rysunek 7. Skumulowane stopy zwrotu dla kolejno rozpatrywanych 200 szeregów czasowych dla kolejno rozpatrywanych 12 zmiennych obserwowanych

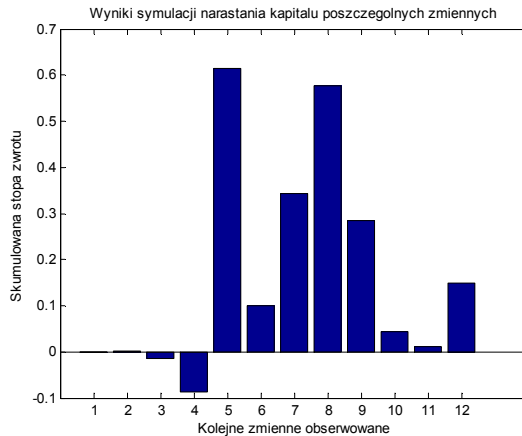
## 6. Uwagi końcowe

Z przedstawionego na rys. 7 przebiegu narastania kapitału dla poszczególnych zmiennych wynika wyraźnie, że istnieje grupa zmiennych, które użyte jako zmienne objaśniane nie rokuja, a w zasadzie wykluczają możliwość wykorzystania ich jako obiektu obserwowanego w praktycznym celu prognostycznym. To co najmniej 8 zmiennych, których wartość skumulowanego zysku oscyluje wokół zera, a w kilku przypadkach jest ujemna. Na tym tle wyróżniają się zmienne, które dają dość wyraźny wynik dodatni. Wyniki końcowe dla badanych 200 przypadków przedstawiono na rys. 8. Dwie spośród nich (na rys. 8) osiągają wartości skumulowanego zysku rzędu 60% wartości maksymalnej zmiennej (to zmienne o indeksach 5 (złoto) i 8 (ropa naftowa)). Dwie inne dają także wyraźny dodatni wynik rzędu 30% wartości maksymalnej (to zmienne nr 7 (Nikkei 225) i 9 (WIG20)).

W zasadzie na totalną porażkę skazane są wszystkie pary walutowe i indeksy amerykańskie. To w zasadzie logiczny rezultat – to one raczej kształtują światowy rynek niż go „słuchają”.

To, że w rozpatrywanej tu predykcji wielorakiej wyraźnie wyróżniają się jako zmienne obserwowane te papiery wartościowe, których miejsce w łańcuchu przyczynowo-skutkowym jest zgodne i z intuicją, i z logiką (często dwiema sprzecznymi kategoriami filozoficznymi), świadczy na korzyść rozpatrywanego tu modelu opisującego nieprzewidywalny rynek (ryunki).

Uzyskane wyniki mogą stać się inspiracją dla rozwoju badań w kierunku modeli nieliniowej predykcji, w kierunku ekstrakcji przesuniętych wektorów w szeregu Gabora-Kolmogorowa, czy w kierunku wielowarstwowości modelu i zastosowania sieci neuronowych dokonujących automatycznego wyboru stopnia złożoności modelu prognostycznego w bieżącym miejscu przestrzeni danych.



Rysunek 8. Końcowe rezultaty symulacji skumulowanej stopy zwrotu kolejnych zmiennych traktowanych jako zmienne objaśniane przez pozostałe

## Bibliografia

- [1] Draper N.R., Smith H. *Applied regression analysis*. Wiley. New York 1981.
- [2] Ivakhnenko A.G., Zaichenko J.P., Dimitrov V.D. *Priniatije resenij na osnove samoorganizaczi*. Sovetskoe Radio, Moskwa 1976
- [3] Madala H.R., Ivakhnenko A. G. *Inductive Learning Algorithms for Complex Systems Modeling*. CRC Press 1994
- [4] Lemke F., Mueller J.-A. *Self-Organizing Data Mining for a Portfolio Trading System*. Journal of Computational Intelligence in Finance, 1997
- [5] Jajuga K., Kuziak K., Markowski P. *Inwestycje finansowe*. Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu, Wrocław 1998.
- [6] Weisstein, E. W. *Moore-Penrose Matrix Inverse*. Wolfram MathWorld [online] <http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html> [dostęp: 02/2008]
- [7] Greshilov A.A., Stakun V.A., Stakun A.A. *Matematiceskije metody postroenija prognozov*. Radio i sviaz. Moskwa 1997.
- [8] Weisstein, E. W. *Time Series Analysis*. Wolfram MathWorld [online] <http://mathworld.wolfram.com/TimeSeriesAnalysis.html> [dostęp: 02/2008]

# O Ukrytych Modelach Markowa i ich zastosowaniu do rozpoznawania ciągów znaków pisma odręcznego

Przemysław Kłęsk

Politechnika Szczecińska, Wydział Informatyki

## Abstract:

*In the paper, selected informations on Hidden Markov Models (also called Hidden Markov Chains) are reminded. Basic notions are defined and algorithms related to these models are shortly presented. The research part of the papers shows results of three conducted experiments entitled: “pork cutlet”, “form sheet” and “poetaster”. The most important experiment “form sheet” gives a good starting point to a practical application of HMMs to the handwriting recognition. The “poetaster” experiment shows possible application of HMMs in so called “artifical creation”.*

## Słowa kluczowe:

*Hidden Markov Models, Viterbi algorithm, handwriting recognition, artifical creation*

## 1. Wprowadzenie (jawne modele Markova)<sup>11</sup>

Zbiór wielkości losowych  $\{q_t\}$ , które zależą tylko od jednego parametru nazywa się *procesem losowym* lub *stochastycznym*. Zwykle parametr  $t$  utożsamiany jest z czasem [2]. Tak naprawdę możemy więc myśleć o ciągu losowym produkowanym przez pewne źródło, które chcemy zamodelować.

Jeżeli zarówno czas jak i zbiór możliwych wartości w ciągu są dyskretne, to proces stochastyczny nazywa się *łańcuchem stochastycznym*. Wyobraźmy sobie zatem system, który w każdej chwili czasowej może przebywać tylko w jednym stanie pochodzącym ze skończonego zbioru stanów  $q_t \in \{S_1, S_2, \dots, S_N\}$ . W ogólności można rozważać najbardziej szeroką rodzinę łańcuchów stochastycznych zawierającą łańcuchy, które do probabilistycznego opisu wymagają w każdym momencie czasu  $t$ : obecnego numeru stanu a także całej historii wszystkich stanów poprzednich. Jeżeli zawiążemy rozważania tylko do takich łańcuchów, dla których prawdziwe jest

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots) = P(q_t = S_j | q_{t-1} = S_i), \quad (1)$$

---

<sup>11</sup> W niniejszym artykule fragmenty opisujące podstawowe pojęcia i algorytmy dotyczące HMM są w dużej mierze powtarzane za pracą [5]: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, której autorem jest Lawrence R. Rabiner. Zachowano identyczną notację.

czyli prawdopodobieństwo przejścia do pewnego stanu zależy tylko od stanu poprzedniego (a nie od całej historii), to rozważamy rodzinę *łańcuchów Markova* [5, 4].

Jeżeli dodatkowo zawężymy rozważania do przypadku, gdzie prawdopodobieństwa przejść są niezmiennie w czasie, tj.:

$$\forall t \quad P(q_t = S_j | q_{t-1} = S_i) = \text{const.} = a_{ij} \quad 1 \leq i, j \leq N, \quad (2)$$

to mówimy o *łańcuchach jednorodnych* lub inaczej *stacjonarnych* [2]. W ten sposób określona jest macierz prawdopodobieństw przejść pomiędzy stanami  $A = \{a_{ij}\}$ , gdzie  $a_{ij} \geq 0$  oraz  $\sum_{j=1}^N a_{ij} = 1$  dla wszystkich  $1 \leq i \leq N$ . Każdą macierz kwadratową spełniającą te własności nazywamy *macierzą stochastyczną* [2, 4].

Do zdefiniowania konkretnego łańcucha Markova, oprócz macierzy  $A$ , potrzebny jest także początkowy rozkład prawdopodobieństwa poszczególnych stanów tj. rozkład dla chwili  $t = 1$ . Oznaczmy taki rozkład przez  $\pi = (\pi_1, \pi_2, \dots, \pi_N)$ . Rozkład  $\pi$  można nazwać rozkładem *a priori* – reprezentuje on naszą wiarę lub wiedzę o prawdopodobieństwie pojawienia się poszczególnych stanów w chwili  $t = 1$ . Łańcuchem Markova (jawnym) możemy zatem nazwać parę  $\lambda = (A, \pi)$  [4].

Interesującym pytaniem jest: „Jaki jest rozkład prawdopodobieństwa na poszczególne stany w chwili o numerze  $T$ ?”, tj. po wykonaniu się  $T - 1$  kroków czasowych. Zauważamy, że po jednym kroku czasowym szukany rozkład wynosi  $\pi \cdot A$ , jako że iloczyn wiersza rozkładu początkowego z każdą  $j$ -tą kolumną macierzy  $A$  daje prawdopodobieństwo całkowite znalezienia się w stanie  $S_j$ :  $\pi_1 a_{1j} + \pi_2 a_{2j} + \dots + \pi_N a_{Nj}$ . Po dwóch krokach czasowych szukany rozkład to  $\pi \cdot A \cdot A$ , zatem dla chwili  $t = T$  mamy:

$$\pi A^{T-1}. \quad (3)$$

Jeszcze innym ciekawym pytaniem jest [5]: „Jakie jest prawdopodobieństwo pozostawania w konkretnym stanie  $S_i$  przez  $d$  kolejnych momentów czasowych, pod warunkiem, że system jest w tym stanie w obecnym momencie czasowym (np.  $t = 1$ ) i pod warunkiem danego modelu  $\lambda$ ?”. Prawdopodobieństwo to maleje wykładniczo i wynosi ono:

$$P\left(O = \underbrace{(S_i, S_i, \dots, S_i)}_d, S_{j \neq i} | q_1 = S_i, \lambda\right) = a_{ii}^{d-1} (1 - a_{ii}). \quad (4)$$

Ile wynosi wartość oczekiwana liczby chwil trwania danego stanu  $S_i$ ? Z definicji wartości oczekiwanej zapisujemy sumę  $\sum_{d=1}^{\infty} d a_{ii}^{d-1} (1 - a_{ii})$ . W pracy [5] stwierdza się, że suma ta wynosi  $\frac{1}{1 - a_{ii}}$ . Poniżej udowodnimy to stosując technikę tzw. „zaburzenia sumy” (zobacz [3]).

$$\sum_{d=1}^{\infty} d a_{ii}^{d-1} (1 - a_{ii}) = (1 - a_{ii}) \lim_{n \rightarrow \infty} \underbrace{\sum_{1 \leq d \leq n} d a_{ii}^{d-1}}_{S_n}. \quad (5)$$



Technika ta polega na wyjściu od równości  $S_{n+1} = S_{n+1}$  i, „zaburzając” ją, uzyskaniu po obu jej stronach składników  $S_n$ , ale z różnymi współczynnikami, tak aby nie skasowały się one do zera:

$$\begin{aligned}
 S_{n+1} &= S_{n+1} \\
 \sum_{1 \leq d \leq n} da_{ii}^{d-1} + (n+1)a_{ii}^n &= 1 \cdot a_{ii}^0 + \sum_{2 \leq d \leq n+1} da_{ii}^{d-1} \\
 S_n + (n+1)a_{ii}^n &= 1 + \sum_{2 \leq d+1 \leq n+1} (d+1)a_{ii}^d \\
 S_n + (n+1)a_{ii}^n &= 1 + \sum_{1 \leq d \leq n} (d+1)a_{ii}^d \\
 S_n + (n+1)a_{ii}^n &= 1 + a_{ii}S_n + \sum_{1 \leq d \leq n} a_{ii}^d \tag{6}
 \end{aligned}$$

Ostatni składnik łatwo obliczyć, jako że jest on sumą ciągu geometrycznego  $\sum_{1 \leq d \leq n} a_{ii}^d = \frac{a_{ii}(1-a_{ii}^n)}{1-a_{ii}}$ , zatem ostatecznie dostajemy:

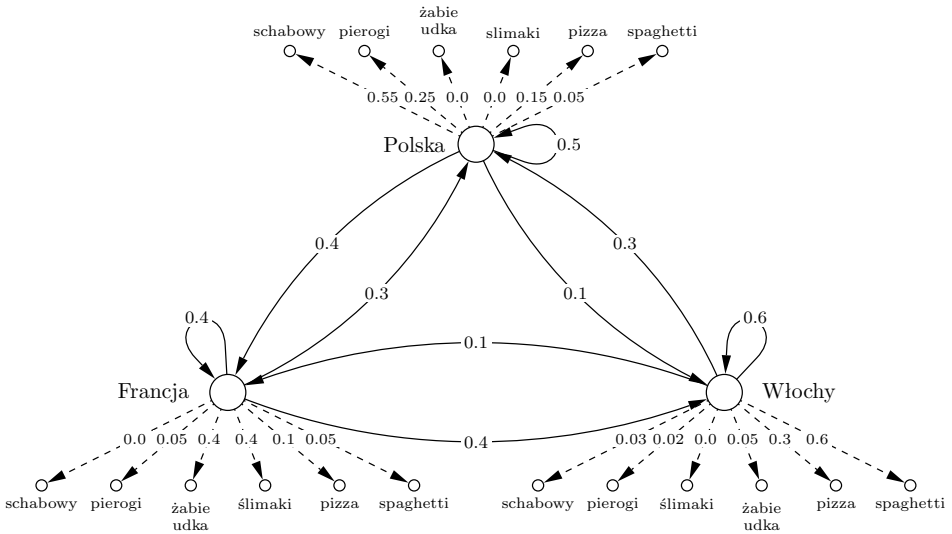
$$S_n = \frac{1}{1-a_{ii}} \left( 1 + \frac{a_{ii}(1-a_{ii}^n)}{1-a_{ii}} - (n+1)a_{ii}^n \right). \tag{7}$$

Wracając do granicy i pamiętając, że  $a_{ii} < 1$  (dla  $a_{ii} = 1$  suma jest rozbieżna – pozostanie w danym stanie jest zdarzeniem pewnym, więc stan trwa nieskończenie długo):

$$\begin{aligned}
 (1-a_{ii}) \lim_{n \rightarrow \infty} S_n &= \lim_{n \rightarrow \infty} \left( 1 + \frac{a_{ii}(1-a_{ii}^n)}{1-a_{ii}} - (n+1)a_{ii}^n \right) \\
 &= 1 + \frac{a_{ii}}{1-a_{ii}} = \frac{1}{1-a_{ii}}. \tag{8}
 \end{aligned}$$

## 2. Ukryte Modele Markowa i trzy podstawowe problemy rozwiązywane w ich ramach

Dotychczas rozpatrywany był jawny model Markowa, w którym obserwacje utożsamia się ze stanami, tzn. stany traktuje się jako pewne obserwowalne fizyczne zdarzenia. Teraz rozszerzmy to na niejako podwójny proces losowy, tzn. taki, gdzie obserwacja jest probabilistyczną funkcją stanu. Jeden proces losowy rządzi przechodzeniem po stanach, a drugi proces losowy rządzi generowaniem obserwacji w poszczególnych stanach. Ścieżki stanów nie widzimy, widzimy tylko sekwencję obserwacji. Rys. 1 dobrze obrazuje tego typu proces losowy – jest to rysunek, który w dalszej części posłuży do eksperymentu o nazwie „schabowy”.



Rysunek 1. Podróże i posiłki – przykładowe źródło Markowskie o zbiorze stanów  $\{\text{Polska, Francja, Włochy}\}$  ( $N = 3$ ) i zbiorze obserwacji  $\{\text{schabowy, pierogi, żabie udka, ślimaki, pizza, spaghetti}\}$  ( $M = 6$ ). Obserwacja jest probabilistyczną funkcją stanu. Ścieżka przejść pomiędzy stanami (trasa podróży) jest ukryta, widoczna jest tylko sekwencja obserwacji (posiłki).

Niech  $\{v_1, v_2, \dots, v_M\}$  oznacza skończony zbiór możliwych obserwacji. Probabilistyczną funkcję stanu definiuje następująca macierz:

$$B = \{b_i(k)\} = \begin{pmatrix} b_1(1) & b_1(2) & \cdots & b_1(M) \\ b_2(1) & b_2(2) & \cdots & b_2(M) \\ \vdots & \vdots & \ddots & \vdots \\ b_N(1) & b_N(2) & \cdots & b_N(M) \end{pmatrix}, \quad (9)$$

gdzie

$$b_i(k) = P(O_t = v_k | q_t = S_i) \quad \forall t \quad (10)$$

jest prawdopodobieństwem zaobserwowania obserwacji  $v_k$  pod warunkiem przebywania w stanie  $S_i$ . Ukrytym łańcuchem Markowa nazywamy trójkę  $\lambda = (A, B, \pi)$  [5, 4].

## 2.1. Trzy podstawowe problemy rozwiązywane w ramach Ukrytych Modeli Markowa

1. Mając daną pewną sekwencję obserwacji  $O = (O_1 O_2 \dots O_T)$  i model  $\lambda = (A, B, \pi)$ , należy obliczyć jakie jest prawdopodobieństwo tej sekwencji obserwacji pod warunkiem modelu, tj.  $P(O|\lambda) = ?$ .

2. Mając daną pewną sekwencję obserwacji  $O = (O_1 O_2 \dots O_T)$  i model  $\lambda = (A, B, \pi)$ , należy znaleźć najbardziej prawdopodobną ścieżkę stanów  $Q = (q_1 q_2 \dots q_T)$ , która towarzyszyła generowaniu danej sekwencji obserwacji.
3. Mając daną pewną sekwencję obserwacji  $O = (O_1 O_2 \dots O_T)$  oraz znając tylko liczby  $M$  i  $N$ , należy nastroić model, tzn. znaleźć możliwie najlepszą zawartość trójki  $\lambda = (A, B, \pi)$ , tak aby maksymalizować  $P(O|\lambda)$ .

Wszystkie powyższe problemy udaje się rozwiązywać w czasie rzędu  $O(TN^2)$  poprzez sformułowanie odpowiednich indukcji i wykorzystaniu struktury *kraty* (ang. *lattice*). Należy wspomnieć, że gdyby chcieć rozwiązywać te problemy w sposób „zachłanny” tj. m.in. poprzez definicyjne zastosowanie wzoru na prawdopodobieństwo całkowite (licząc sumę składników po składniku) natrafiamy na złożoność  $O(TN^T)$ . Jest ona wykładnicza ze względu na długość ścieżki i tym samym nie do przeliczenia nawet dla stosunkowo małych wartości  $T$  i  $N$  np.  $N = 10, T = 50$ , choćby w czasie trwania cywilizacji.

## 2.2. Rozwiązanie problemu nr 1

### Algorytm Forward-Backward

Definiujemy tzw. *zmienną wprzód*:  $\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda)$ , która jest prawdopodobieństwem tego, że wystąpił początkowy fragment obserwacji  $O_1 O_2 \dots O_t$ , oraz że w chwili  $t$  system jest w stanie  $S_i$  pod warunkiem modelu.

$$\text{Inicjalizacja: } \alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N. \quad (11)$$

$$\text{Indukcja: } \alpha_{t+1}(j) = \left( \sum_{1 \leq i \leq N} \alpha_t(i) a_{ij} \right) b_j(O_{t+1}), \quad 1 \leq j \leq N, 1 \leq t \leq T-1. \quad (12)$$

$$\text{Stop: } P(O|\lambda) = \sum_{1 \leq i \leq N} \alpha_T(i). \quad (13)$$

Powyższy algorytm stanowi część *forward* całości i jest to część wystarczająca dla znalezienia  $P(O|\lambda)$ . Poniżej przedstawiamy część *backward*, która nie jest wymagana dla rozwiązania problemu 1, natomiast przydaje się do rozwiązania problemów 2 i 3.

Definiujemy tzw. *zmienną wstecz*:  $\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T | q_t = S_i, \lambda)$  która jest prawdopodobieństwem tego, że wystąpił końcowy fragment obserwacji  $O_{t+1} O_{t+2} \dots O_T$  pod warunkiem tego, że w chwili  $t$  system był w stanie  $S_i$  i pod warunkiem modelu.

$$\text{Inicjalizacja: } \beta_T(i) = 1, \quad 1 \leq i \leq N. \quad (14)$$

$$\text{Indukcja: } \beta_{t-1}(i) = \sum_{1 \leq j \leq N} a_{ij} b_j(O_t) \beta_t(j), \quad 1 \leq i \leq N, t = T, T-1, \dots, 2. \quad (15)$$

$$\text{Stop: } P(O|\lambda) = \sum_{1 \leq i \leq N} \pi_i b_i(O_1) \beta_1(i). \quad (16)$$

### 2.3. Rozwiązanie problemu nr 2

#### Sposób 1

Szukamy ścieżki  $Q^* = (q_1^* q_2^* \dots q_T^*)$  złożonej z indywidualnie najbardziej prawdopodobnych stanów w każdym kroku  $t$ , tj.  $q_t^* = \arg \max_{1 \leq i \leq N} P(q_t = S_i | O, \lambda)$ . Definiujemy zmienną  $\gamma_t(i) = P(q_t = S_i | O, \lambda)$ . Okazuje się, że  $\gamma_t(i)$  daje się przedstawić za pomocą  $\alpha_t(i)$  i  $\beta_t(i)$ :

$$\begin{aligned} \alpha_t(i)\beta_t(i) &= P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) P(O_{t+1} O_{t+2} \dots O_T | q_t = S_i, \lambda) \\ &= P(\underbrace{O_1 \dots O_t}_O, q_t = S_i | \lambda) \\ &= \frac{P(O, q_t = S_i, \lambda)}{P(\lambda)} \frac{P(O, \lambda)}{P(O, \lambda)} = \underbrace{P(q_t = S_i | O, \lambda)}_{\gamma_t(i)} P(O | \lambda), \\ \gamma_t(i) &= \frac{\alpha_t(i)\beta_t(i)}{P(O | \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{1 \leq j \leq N} \alpha_t(j)\beta_t(j)}. \end{aligned} \quad (17)$$

Jak widać  $P(O | \lambda)$ , które pojawiło się w mianowniku, nie musi być konieczne obliczane za pomocą wzorów (13) lub (16). Wystarczy zastosować czynnik normalizujący  $1/\sum_{1 \leq j \leq N} \alpha_t(j)\beta_t(j)$ , aby całość stała się poprawną miarą prawdopodobieństwa<sup>12</sup>.

Optymalną ścieżkę stanów  $Q^* = (q_1^* q_2^* \dots q_T^*)$  znajdujemy ustawiając każde

$$q_t^* = \arg \max_{1 \leq i \leq N} \gamma_t(i). \quad (18)$$

Jest to ścieżka optymalna w tym sensie, że maksymalizuje ona kryterium wartości oczekiwanej liczby poprawnych stanów na poszczególnych pozycjach. Niestety nie gwarantuje to, że cała ścieżka jako całość jest najbardziej prawdopodobną przy danej sekwencji obserwacji  $O$ . Lepszymi kryteriami byłyby więc takie, które maksymalizują oczekiwaną liczbę poprawnych podciągow o długości 2, 3, itd., czy też wreszcie kryterium, które szuka po prostu najbardziej prawdopodobnego całego ciągu stanów.

#### Sposób 2 – Algorytm Viterbi’ego [6]

Definiujemy następującą zmienną:

$$\delta_t(i) = \max_{(q_1 q_2 \dots q_{t-1})} P(q_1 q_2 \dots q_{t-1}, q_t = S_i, O_1 O_2 \dots O_t | \lambda),$$

tzn. wśród wszystkich ścieżek stanów o długości  $t-1$  i takich, że  $q_t = S_i$  wybieramy tę, która maksymalizuje prawdopodobieństwo pojawienia się tej ścieżki stanów wraz z towarzyszącą jej obserwacją  $O_1 O_2 \dots O_t$ . W algorytmie przedstawionym

<sup>12</sup> Widać jednocześnie, że tym sposobem możliwe są różne przedstawienia dla  $P(O | \lambda) = \sum_{1 \leq j \leq N} \alpha_t(j)\beta_t(j)$ , poprzez wstawienie do tego zapisu dowolnego momentu czasowego  $1 \leq t \leq T$ .

poniżej potrzebna będzie dodatkowo pewna struktura  $\psi$  (np. tablica) pozwalająca na śledzenie po kracie  $N$  różnych ścieżek, jako że dopiero w ostatnim kroku indukcji wiadome będzie, która spośród nich jest zwycięska.

$$\text{Inicjalizacja: } \delta_1(i) = \pi_i b_i(O_1), \quad (19)$$

$$\psi_1(i) = 0, \quad 1 \leq i \leq N. \quad (20)$$

$$\text{Indukcja: } \delta_{t+1}(j) = \left( \max_{1 \leq i \leq N} \delta_t(i) a_{ij} \right) b_j(O_{t+1}), \quad (21)$$

$$\psi_t(i) = \arg \max_{1 \leq i \leq N} \delta_t(i) a_{ij}, \quad 1 \leq j \leq N, \quad 1 \leq t \leq T-1. \quad (22)$$

$$\text{Stop: } P^* = \max_{1 \leq i \leq N} \delta_T(i). \quad (23)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i). \quad (24)$$

$$\text{Śledzenie ścieżki wstecz: } q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1. \quad (25)$$

## 2.4. Rozwiązanie problemu nr 3

Zdefiniujmy następującą zmienną:  $\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$ . Jest to prawdopodobieństwo tego, że z chwili  $t$  na chwilę  $t+1$  nastąpi przejście  $S_i \rightarrow S_j$  pod warunkiem danej sekwencji obserwacji i pod warunkiem modelu. Daje się ono wyrazić za pomocą zmiennych *wprzód* i *wstecz*:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\underbrace{\sum_{1 \leq k \leq N} \sum_{1 \leq l \leq N} \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}_{P(O|\lambda)}} \quad (26)$$

Na podstawie zdefiniowania zmiennych  $\gamma$  i  $\xi$  można zauważyć, że:

- wartość oczekiwana liczby pobyków w stanie  $S_i$  w sekwencji o długości  $T$  wynosi:  $\sum_{1 \leq t \leq T} 1 \cdot \gamma_t(i)$ ,
- wartość oczekiwana liczby przejść  $S_i \rightarrow S_j$  w sekwencji o długości  $T$  wynosi:  $\sum_{1 \leq t \leq T-1} 1 \cdot \xi_t(i, j)$ .

## Iteracyjna procedura strojenia modelu Markowa

Rozpoczynamy od losowego zainicjalizowania trójki  $\lambda = (A, B, \pi)$  zachowując tylko własności stochastyczne tych macierzy. Pojedyncza iteracja uzyskania poprawionego modelu  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$  ze starego modelu  $\lambda = (A, B, \pi)$  jest następująca:

$$\begin{aligned}\bar{a}_{ij} &= \frac{\text{wartość oczekiwana liczby przejść } S_i \rightarrow S_j}{\text{wartość oczekiwana liczby pobytów w } S_i} \\ &= \frac{\sum_{1 \leq t \leq T-1} \xi_t(i, j)}{\sum_{1 \leq t \leq T} \gamma_t(i)}, \quad 1 \leq i, j \leq N.\end{aligned}\quad (27)$$

$$\begin{aligned}\bar{b}_i(k) &= \frac{\text{wartość oczekiwana liczby pobytów w } S_i \text{ i zaobserwowania } v_k}{\text{wartość oczekiwana liczby pobytów w } S_i} \\ &= \frac{\sum_{\substack{1 \leq t \leq T \\ O_t = v_k}} \gamma_t(i)}{\sum_{1 \leq t \leq T} \gamma_t(i)}, \quad 1 \leq i \leq N, \quad 1 \leq k \leq M.\end{aligned}\quad (28)$$

$$\bar{\pi}_i = \gamma_1(i), \quad 1 \leq i \leq N. \quad (29)$$

Dowodniono (patrz np. [1]), że w wyniku powyższej procedury iteracyjnej: albo (1) osiągnięto już tzw. punkt krytyczny i  $\lambda = \bar{\lambda}$  lub też (2)  $P(O|\bar{\lambda}) > P(O|\lambda)$  tzn. otrzymaliśmy model, dla którego jest bardziej prawdopodobne, że wyprodukował on daną sekwencję obserwacji. Należy zaznaczyć, że przestrzeń, w której szukamy najlepszego modelu może mieć wiele punktów krytycznych i efekt działania procedury iteracyjnej może zależeć od punktu początkowego.

Ważnym wariantem procedury iteracyjnej jest wariant, który pozwala do strojenia wykorzystać nie tylko pojedynczą sekwencję obserwacji, a cały zbiór różnych takich sekwencji wyprodukowanych przez badane źródło. Oznaczmy ten zbiór jako  $\{O^1, O^2, \dots, O^L\}$ , gdzie  $O^l = (O_1^l, O_2^l, \dots, O_{T_l}^l)$ ,  $1 \leq l \leq L$ . Pojedyncza iteracja przyjmuje w tym wariantcie postać (patrz także [5]):

$$\bar{a}_{ij} = \frac{\sum_{1 \leq l \leq L} \frac{1}{P(O^l|\lambda)} \sum_{1 \leq t \leq T_l-1} \alpha_t^l(i) a_{ij} b_j(O_{t+1}^s) \beta_{t+1}^l(j)}{\sum_{1 \leq l \leq L} \frac{1}{P(O^l|\lambda)} \sum_{1 \leq t \leq T_l} \alpha_t^l(i) \beta_t^l(i)}, \quad 1 \leq i, j \leq N. \quad (30)$$

$$\bar{b}_i(k) = \frac{\sum_{1 \leq l \leq L} \frac{1}{P(O^l|\lambda)} \sum_{\substack{1 \leq t \leq T_l \\ O_t^l = v_k}} \alpha_t^l(i) \beta_t^l(i)}{\sum_{1 \leq l \leq L} \frac{1}{P(O^l|\lambda)} \sum_{1 \leq t \leq T_l} \alpha_t^l(i) \beta_t^l(i)}, \quad 1 \leq i \leq N, \quad 1 \leq k \leq M. \quad (31)$$

$$\bar{\pi}_i = \frac{\sum_{1 \leq l \leq L} \frac{1}{P(O^l|\lambda)} \alpha_1^l(i) \beta_1^l(i)}{\sum_{1 \leq l \leq L} \frac{1}{P(O^l|\lambda)}}, \quad 1 \leq i \leq N. \quad (32)$$

### 3. Możliwe zastosowania modeli Markova

Ukryte Modele Markova są najbardziej znane ze swoich zastosowań w dziedzinie *rozpoznawania wzorców przebiegających w czasie* (ang. *temporal pattern recognition*) takich jak np.: rozpoznawanie mowy, rozpoznawanie pisma odręcznego, rozpoznawanie gestów, śledzenie/generowanie partytury muzycznej (ang. *musical score following*), generowanie akompaniamentu muzycznego, a także kryptoanaliza. W tego typu aplikacjach ważną rolę odgrywa zwykle algorytm Viterbi'ego. Do zbioru obserwacji wprowadza się wówczas różne możliwe zaburzone postacie pewnych obiektów np.: różne sposoby wypowiedzenia danej głoski, różne sposoby odręcznego zapisania danej litery, czy też różne sposoby odegrania przez muzyka danego dźwięku. Natomiast zbiór stanów stanowią te obiekty w rozumieniu precyzyjnym (głoski zapisane transkrypcjami, zapis literowy słów np. w edytorze tekstu, nuty w partyturze). Zadanie polega więc na odkryciu najbardziej prawdopodobnej sekwencji precyzyjnej na podstawie sekwencji zaburzonej.

Znane są także zastosowania modeli Markova w bioinformatyce i genetyce do: predykcji kodowania protein, modelowania rodzin powiązanych sekwencji DNA, predykcji sekwencji proteinowej drugiego rzędu na podstawie sekwencji pierwszego rzędu.

Należy także zwrócić uwagę na to, że po nastrojeniu, model Markova może z powodzeniem służyć do samego symulowania źródła. Prowadzi to do zastosowań w ramach tzw. „*sztucznej twórczości*”. Poza tym sztuczny generator źródła może być przydatny w sytuacjach, gdy nie jest możliwe lub jest zbyt kosztowne obserwowanie źródła prawdziwego.

### 4. Przeprowadzone badania

#### 4.1. Eksperyment „schabowy”

Modelowany był proces losowy, który obrazuje rys. 1. Jest to proces losowy, w którym zbiór stanów to kraje, po których podróżuje pewien podróżnik: {Polska, Francja, Włochy}, a zbiór obserwacji to możliwe posiłki: {schabowy, pierogi, żabie udka, ślimaki, pizza, spaghetti}. Problem – niejako detektywistyczny – polega na wykryciu, gdzie kolejno podróżnik był (ścieżka krajów) na podstawie tego, co jadł (sekwencja posiłków).

Z rysunku można odczytać informacje o zawartości macierzy  $A$  i  $B$ . Jako rozkład początkowy przyjęto  $\pi = (0.6, 0.3, 0.1)$ . Wyniki testów zawiera tabela 1. Każdy wygenerowany zbiór obserwacji obejmował sekwencje o losowej długości od 5 do 50. W przypadkach gdy strojono model (poczynając od losowego), liczba iteracji procedury strojenia była równa 100.

Poniżej przedstawiono: (1) przykładową sekwencję obserwacji, (2) faktyczną ścieżkę stanów, która wytworzyła tę obserwację, (3) ścieżkę stanów wykrytą przez algorytm Viterbi'ego znając prawdziwe  $\lambda$ , (4) ścieżkę stanów wykrytą przez algorytm Viterbi'ego znając nastrojone  $\bar{\lambda}$ .

rozmiar zbioru obserwacji	odsetek w całości poprawnych wykrytych ścieżek (znając prawdziwe $\lambda$ )	odsetek poprawnych stanów w wykrytych ścieżkach (znając prawdziwe $\lambda$ )	odsetek w całości poprawnych wykrytych ścieżek (na podst. $\bar{\lambda}$ )	odsetek poprawnych stanów w wykrytych ścieżkach (na podst. $\bar{\lambda}$ )
50	0.1	0.844	0.04	0.773
100	0.06	0.842	0.06	0.813
500	0.062	0.842	0.054	0.804

Tabela 1. Wyniki testów dla eksperymentu o nazwie „schabowy”.

- (1): (schabowy, spaghetti, spaghetti, spaghetti, schabowy, schabowy, schabowy, schabowy, schabowy, zabie udka, spaghetti, schabowy, pierogi, pizza, pizza).
- (2): (Polska, Włochy, Włochy, Włochy, Polska, Polska, Polska, Polska, Polska, Francja, Włochy, Polska, Polska, Polska, Francja).
- (3): (Polska, Włochy, Włochy, Włochy, Polska, Polska, Polska, Polska, Polska, Francja, Włochy, Polska, Polska, Polska, Polska).
- (4): (Polska, Francja, Włochy, Włochy, Polska, Polska, Polska, Polska, Polska, Francja, Włochy, Polska, Polska, Francja, Włochy).

Należy pamiętać, że nastrojony model, może być bliski prawdziwemu w sensie własności statystycznych, ale może różnić się ze względu na permutację numerów stanów. Aby móc porównywać wyniki działania algorytmu Viterbiego z faktycznymi ścieżkami stanów, należy po nastrojeniu modelu znaleźć także poprawną permutację tych numerów. Do tego celu potrzebne już jest uczenie z nadzorem. Mówiąc w skrócie należy porównywać rozkłady w wierszach znalezionej macierzy  $\bar{B}$  ze zbiorem sekwencji obserwacji, gdzie każda obserwacja opatrzona byłaby numerem stanu, z którego pochodzi (nadzór). Można tu skorzystać np. z kryterium maksymalnej wiarygodności (ang. *maximum likelihood*) patrząc na entropie empiryczne. I tak prawdziwy indeks  $i$  pewnego stanu  $S_i$  można znaleźć jako:

$$i = \arg \max_{1 \leq j \leq N} \sum_{1 \leq l \leq L} \sum_{t \in S_i^l} \frac{1}{\#S_i^l} \log \bar{b}_j(O_t), \quad (33)$$

gdzie  $S_i^l$  oznacza zbiór tych momentów czasowych  $t$  w  $l$ -tej ukrytej ścieżce stanów, w których system był w stanie  $S_i$ , tj.  $S_i^l = \{t: q_t^l = S_i\}$ , zaś  $\#S_i^l$  oznacza moc tego zbioru.

## 4.2. Eksperyment „formularz”

Niniejszy eksperyment może stanowić dobry punkt wyjścia do stworzenia pełnej aplikacji rozpoznawania pisma odręcznego. Nazwa „formularz” ma sugerować pewne uproszczenie zadania, a mianowicie odosobnienie liter – tak jak ma to miejsce, gdy wypełniamy przygotowany arkusz formularza z odrębnymi kratkami dla każdego znaku. Zadania tego typu (nawet uproszczone) pojawiają się w życiu, często w skali masowej i proszą się o zautomatyzowanie np.: wypełnianie deklaracji podatkowych PIT, wypełnianie wniosków o zmianę dokumentów osobistych, itp.



Eksperyment przeprowadzony był następująco. Zbiorem stanów był zbiór wybranych dużych liter:  $\{A, B, C, D, E, I, K, M, N, O, Z\}$ . Macierz  $B$  modelu została zbudowana pod nadzorem w sposób następujący. Dla każdego stanu – litery – przygotowano 50 plików graficznych o wymiarach  $50 \times 30$  zawierających różne odręczne sposoby napisania danej litery. Każdy plik poddano pomniejszeniu do wymiarów  $p \times q = 5 \times 3$  na zasadzie pewnego skrótu rastrowego, tj. decydując czy piksel  $(i, j)$  w pomniejszonym obrazie ma być zaczerniony patrzono na stosunek pikseli czarnych do białych w odpowiadającym mu kwadracie o wymiarach  $10 \times 10$  pochodzącym z obrazu oryginalnego. Jeżeli taki stosunek przekraczał pewną wartość progową, to dany piksel  $(i, j)$  był zaczerniany. Wartość progę w wyniku eksperymentów dobrano na 0.28 (białe tło dominuje nad kolorem czarnym w oryginalnym obrazie). Rysunek 2 ilustruje powyższy opis.



Rysunek 2. Kilka wersji odręcznego napisania litery  $A$  i ich skróty rastrowe.

Ze względu na wymiary skrótów  $5 \times 3$  przestrzeń różnych możliwych obserwacji ma moc  $2^{5 \cdot 3} = 32768$ . Jest oczywiste, że litery pisane odręcznie na różne sposoby w sposób naturalny nie wyczerpią całej tej przestrzeni, stąd też na pewno liczba  $M < 32768$ .

Każdy skrót rastrowy zamieniano na liczbę wg wzoru  $\sum_{i=1}^p \sum_{j=1}^q \text{pixel}(i, j) \cdot 2^{(i-1)q+(j-1)}$ . Dla każdej litery zbiór jej różnych obserwacji (skrótów) przekształconych na liczby można nazwać *książką kodową* danej litery. Dla każdej książki kodowej zapamiętywano także częstości pojawienia się poszczególnych jej elementów. Utożsamiając te częstości z rozkładami prawdopodobieństwa, otrzymujemy praktycznie gotowe wiersze dla macierzy  $B$  modelu Markowa.

Bardzo istotnym szczegółem było wprowadzenie do każdej książki kodowej całego zaczernionego skrótu, który ma sens symbolu nieznanego '?'. Ta sztuczka przydaje się na etapie już samego rozpoznawania całych napisów za pomocą gotowego modelu. W momencie, gdy natrafimy na znak, który nie należy do żadnej książki kodowej, to możemy ten znak utożsamiać z symbolem nieznanym '?', który z takim samym prawdopodobieństwem  $\frac{1}{50}$  może pojawić się w każdym stanie (literze). Dzięki temu dalsze rozstrzygnięcie, jaką literą '?' tak na naprawę jest, odbywa się poprzez algorytm Viterbi'ego na podstawie własności statystycznych języka polskiego zawartych w nastrojonej dwójce  $A$  i  $\pi$ . Chodzi tu o częstsze lub rzadsze pojawianie się w języku polskim pewnych kombinacji liter, czy chociażby o konieczność pojawienia się samogłoski co jakiś czas, itp. Model Markowa poznał

te własności podczas strojenia dzięki uczącemu zbiorowi obserwacji – o nim za chwilę.

Budując książki kodowe, odnotowano następujące fakty:

1. średnia długość książek kodowych wyniosła 22.73, co oznacza, że pomimo dostarczenia 50 różnych dużych obrazów dla każdej litery, przeciętnie dostajemy około 23 różnych skrótów na literę – duże obrazy mogą odwzorowywać się na ten sam skrót (stąd też możemy w ogóle mówić o rozkładzie prawdopodobieństwa skrótów danej litery);
2. zbiór będący sumą mnogościowa wszystkich książek kodowych uzyskał rozmiar 222, czyli  $M = 222$ ;
3. suma długości wszystkich książek kodowych wyniosła 250, co jest większe od  $M$  – zatem istnieją obserwacje, które przynależą do więcej niż jednej książki kodowej.

Punkt 3 wynika z możliwych podobieństw liter (np.  $\mathcal{D}$  i  $\mathcal{O}$ ,  $\mathcal{M}$  i  $\mathcal{N}$ ) i uzyskiwania dla nich tych samych skrótów. Z tego wynika dalej, że niemożliwe jest jednoznaczne rozpoznanie każdego całego napisu, poprzez odrębne rozpoznanie poszczególnych liter.

Zbiorem uczącym, który posłużył do nastrojenia modelu był zbiór złożony z 1000 sekwencji obserwacji, wygenerowany poprzez różne sposoby napisania 50 słów (dla każdego słowa 20 różnych wersji):

{AMOKIEM, BABA, BACA, BACZA, BADA, BAM, BICZ, BIEDA, BIEDNI, BIKINI, BOKIEM, CIEBIE, CIECZ, CIEKNIE, CODZIENNIE, CZAD, CZĘKA, DACTE, DZIADA, DZICZE, DZIĘKA, DZIEKAN, IKONA, KACZKI, KANA, KICZ, KIMONO, KONA, MAMA, MAMBIE, MAMIE, MIMA, MINIE, NADZIENIE, NIANTA, NICI, NIEZNANE, OCZKO, OKIEM, OKNEM, OKNO, ZADA, ZADANIEM, ZAKAZ, ZAZNACZENIE, ZBADAM, ZBOCZENIEC, ZDANIE, ZNACZEK, ZNANE}.

Po nastrojeniu modelu Markova (3 iteracje), testowano jego skuteczność na zbiorze 50 nowych napisów, nie zawartych w zbiorze uczącym i reprezentujących inne wyrazy. Każdy napis odwzorowywano na skrót. Odnotowywano i zliczano znaki nie należące do żadnej książki kodowej, a zatem przeszły one w ‘?’. Następnie wykonywano algorytm Viterbi’ego i porównywano jego wynik działania z oczekiwanym wyrazem. Rysunki 3 i 4 przedstawiają przykładowe rozpoznania.

Rysunek 3. Przykład poprawnego rozpoznania. Po skrócie, znaki o numerach 5 i 8 nie pochodziły z żadnej książki kodowej, więc zostały zamienione na zaczernione prostokąty tj. symbole ‘?’ (ostatni wiersz). Wynikowa ścieżka stanów Viterbi’ego:

*NIEZNACZNI E.*



Rysunek 4. Przykład niepoprawnego rozpoznania. Po skrócie, znaki o numerach 3, 7 i 8 nie pochodziły z żadnej książki kodowej, więc zostały zamienione na zaczernione prostokąty tj. symbole ‘?’. Wynikowa ścieżka stanów Viterbi’ego: *ZIENNIENTIEM*. Znak o numerze 3 został wstawiony poprawnie, znaki 7 i 8 niepoprawnie. Pojawił się też drugi rodzaj błędu, tj. znak o numerze 4 został rozstrzygnięty na podstawie całości ciągu jako  $\mathcal{N}$  raczej niż  $\mathcal{M}$  (znajduje się on w obu książkach kodowych).

Wyniki testów były następujące. Odsetek znaków, które przynależały do przynajmniej jednej książki kodowej wyniósł 75.21%. Należy rozumieć, że wielkość zbliżoną otrzymalibyśmy chcąc rozpoznawać całe napisy na podstawie odrębnego rozpoznawania poszczególnych znaków ze skrótów rastrowych  $5 \times 3$ , odcinając się od jakiegokolwiek wiedzy o języku polskim i nie korzystając z modelu Markova. Zatem względem tej liczby należy oceniać skuteczność rozpoznawania całych napisów przy zastosowaniu modelu Markova i algorytmu Viterbi’ego. Odsetek poprawnych znaków w całych rozpoznawanych wyrazach wyniósł 90.25%, co znacznie przewyższa 75.21%. Widać tym samym zysk z dodania pewnej wiedzy o języku polskim zawartej w modelu Markova. Odsetek ścieżek Viterbi’ego poprawnych w całości wyniósł 60%.

Warto jeszcze wspomnieć o unikaniu zbyt „ostrego” nastrojenia modelu w tego typu aplikacjach. Chodzi o to, że model poznaje tylko skończony zbiór wyrazów podczas strojenia. Nie wszystkie własności języka polskiego mogą ujawnić się w takim zbiorze, a należy pamiętać, że mogą one pojawić się później na etapie rozpoznawania. Po wykonaniu zbyt wielu iteracji uczących, model przydzieli prawdopodobieństwa  $a_{ij}$  bliskie zeru pewnym przejściom, których nie było w tym zbiorze. Np. pojawienie się w wyrazie podciągu *II* model będzie uznawał za bardzo mało prawdopodobne, jeżeli „nie widział” podczas strojenia żadnego wyrazu, który miałby taki podciąg (np. *DANII*). Innym możliwym sposobem na naprawienie przestrojenia (oprócz małej liczby iteracji) jest skorygowanie macierzy  $A$  już po nastrojeniu. Tzn. elementy  $a_{ij}$  bliskie zeru można podnieść o pewną wartość  $\epsilon$  i znormalizować wiersze, tak aby zachowały sens probabilistyczny.

### 4.3. Eksperyment „wierszokleta”

Niniejszy eksperyment dotyczy przypadku jawnego łańcucha Markova, gdzie  $\lambda = (A, \pi)$ , i ukazuje on, jak nastrojony model może pracować jako generator

losowy naśladowujący zamodelowane źródło. Jako sekwencję obserwacyjną wybrano wiersz Kazimierza Przerwy-Tetmajera „A kiedy będziesz moją żoną”:

A kiedy będziesz moją żoną,  
umiłowaną, poślubioną,  
wówczas się ogród nam otworzy,  
ogród świetlisty, pełen zorzy.

Rozdzwonią nam się kwietne sady,  
pachnąc nam będą winogrody,  
i róże śliczne i powoje  
całować będą włosy twoje.

Pójdziemy cisi, zamyśleni,  
wśród żółtych przymgleń i promieni,  
pójdziemy wolno alejami,  
pomiędzy drzewa, cisi, sami.

Gałązki ku nam zwisać będą,  
narcyzy piąć się srebrną grzędą,  
i padnie biały kwiat lipowy  
na rozkochane nasze głowy.

Ubiorę ciebie w błękit kwiatów,  
niezapominajek i bławatów,  
ustroję ciebie w paproć młodą  
i świat rozświetlę twą urodą.

Pójdziemy cisi, zamyśleni,  
wśród złotych przymgleń i promieni,  
pójdziemy w ogród pełen zorzy  
kiedy drzwi miłość nam otworzy.

Jako zbiór stanów obrano zbiór całych słów występujących w wierszu plus znak końca linii (pominięto znaki interpunkcyjne). Liczba stanów  $N = 73$ , podczas gdy długość wiersza  $T = 128$  (bez powtarzalności słów, nastrojony model odtwarzałby po prostu wiersz w formie niezmienionej). „Jawność” łańcucha, można w praktyce zrealizować ustawiając  $M = N$  i macierz  $B$  jako jednostkową. Po nastrojeniu, używano modelu do generowania losowych „wierszy”. Oto przykładowe wybrane z nich:

a kiedy drzwi miłość nam otworzy  
i padnie biały kwiat lipowy  
pójdziemy w paproć młodą  
wśród żółtych przymgleń i świat rozświetlę twą urodą  
pachnąc nam będą włosy twoje  
ubiorę ciebie w błękit kwiatów

a kiedy będziesz moją żoną  
ogród świetlisty pełen zorzy  
niezapominajek i świat rozświetlę twą urodą  
pójdziemy wolno alejami  
pójdziemy cisi zamyśleni  
i bławatów  
pójdziemy cisi sami  
pójdziemy wolno alejami  
pomiędzy drzewa cisi zamyśleni  
i padnie biały kwiat lipowy

Jak widać niektóre fragmenty mogą nie mieć sensu, niemniej jednak w ogólności otrzymujemy całkiem ciekawe sekwencje.

Interesującym eksperymentem byłoby nauczenie dwóch modeli Markova  $\lambda_1$  i  $\lambda_2$  na podstawie dużej liczby wierszy dwóch autorów, a następnie orzekanie dla

nowego wiersza, od którego autora on pochodzi porównując  $P(O|\lambda_1)$  i  $P(O|\lambda_2)$ . W analogiczny sposób można by także sprawdzać autentyczność różnego rodzaju materiałów, dokumentów, itp.

## Bibliografia

- [1] Baum L. E., Sell G. R. *Growth functions for transformations on manifolds*. Pac.J.Math., vol. 73, 1967, pp. 360–363.
- [2] Bronsztejn I. N., Siemiendajew K. A., Musiol G., Mulig H., *Nowoczesne Kompendium Matematyki*. Wydawnictwo Naukowe PWN, Warszawa, Polska, 2004.
- [3] Graham R. L., Knuth D. E., Patashnik O. *Matematyka konkretna*. Wydawnictwo Naukowe PWN SA, Warszawa, Polska, 2002, Tytuł oryginału: *Concrete Mathematics. A foundation for Computer Science*.
- [4] Kemeny J. G., Snell J. L. *Finite markov chains*. Springer Verlag, U.S.A., 1960,1976.
- [5] Rabiner L. R. *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. IEEE, vol. 77, 2 1989, pp. 257–286.
- [6] Viterbi A. J. *Error bounds for convolutional codes and an asymptotically optimal decoding algorithm*. IEEE, Trans. Informat. Theory, vol. IT-13, 1967, pp. 260–269.



# Integracja metod AHP i SWOT w formułowaniu strategii działania przedsiębiorstwa

Luiza Kuśkiewicz, Jarosław Wątróbski

*Politechnika Szczecińska, Wydział Informatyki*

## Abstract:

*W niniejszej pracy podjęto próbę adaptacji aparatu wielokryterialnego wspomaganie decyzji w formułowaniu strategii gospodarczej przedsiębiorstwa. W publikacji przybliżono problematykę formułowania strategii biznesowej organizacji, dokonano syntezy aktualnie stosowanych w tym zakresie metod i technik oraz przedstawiono założenia metodyczne proponowanego rozwiązania własnego.*

## Słowa kluczowe:

*wielokryterialne wspomaganie decyzji, adaptacja aparatu AHP, strategia gospodarcza, analiza SWOT*

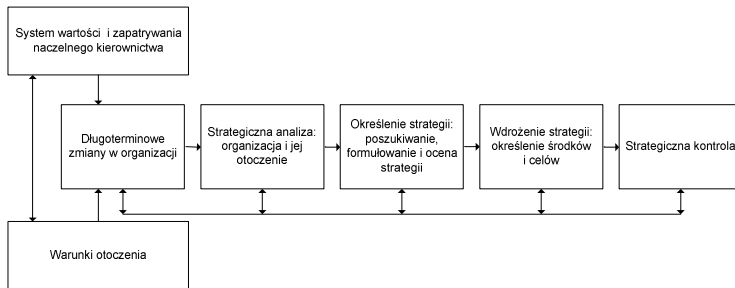
## 1. Wstęp

Pod pojęciem strategii rozumie się *plan pewnego rodzaju świadomie zamierzonego sposobu działania*, albo *wskazówkę (zbiór wskazówek)* pozwalającą radzić sobie w określonej sytuacji. Strategię można postrzegać jako *wzorzec* (a dokładniej *wzorzec w zestawie działań*) wykazujący spójność zamierzoną lub, nie możliwą do osiągnięcia na różne sposoby [1].

Strategię gospodarczą organizacji stanowi układ długookresowych działań, mających na celu realizację przyjętych założeń, misji, celów i długookresowego planu działania [2]. Odgrywa ona podstawową rolę w funkcjonowaniu przedsiębiorstw. Celem strategii jest tworzenie przewagi konkurencyjnej [3].

Istotną rolę w formułowaniu i wdrażaniu strategii organizacji odgrywa jej metodyka [2, 3, 5]. Przyjęta metodyka wskazuje sposób i etapy opracowania strategii gospodarczej. Analiza literatury przedmiotu wskazuje na szereg dostępnych metod i technik doboru odpowiedniej strategii gospodarczej. Do ważniejszych zaliczyć można: metodę SWOT, SPACE, SMART, PEST, metodę Czynników Krytycznych Sukcesu, metodę Siatki Audytu Systemów, czy metody Portfelowe, zwane Analizą BCG.

— Analiza SWOT stanowi efektywną metodę identyfikacji kategorii słabych i silnych stron oraz szans i zagrożeń. Przedmiotem analizy może być przedsiębiorstwo, inwestycja lub dowolna organizacja [3, 2, 7]. Głównym celem analizy SWOT jest określenie aktualnej i perspektywicznej pozycji przedmiotu analizy SWOT oraz prognoza strategii postępowania.



Rysunek 1. Podstawowy model strategicznego planowania w organizacji. Źródło: Kubiak B. F., 2003 (por. Kreikebaum H., 1996).

— Analiza SPACE (ang. *Strategic Position and Action Evaluation* – analiza pozycji strategicznej i ocena działalności firmy) to jedna z metod badania ogólnej zdolności rozwojowej firmy [12]. Ułatwia ona podejmowanie decyzji dotyczących dywersyfikacji działalności firmy i ocenę poszczególnych jej domen.

— Analiza SMART jest to metodyka konstruowania celów organizacyjnych, charakteryzująca właściwie sformułowane cele. Oparta jest na rozwinięciu akronimu SMART, czyli reguły mnemotechnicznej ukazującej najistotniejsze kryteria, które spełniają cele.

— Analiza PEST jest bardzo często wykorzystywana przez eksporterów do oceny środowiska makroekonomicznego. Dotyczy kraju, w którym eksporter prowadzi sprzedaż swoich produktów/usług lub kraju, na obszarze którego eksporter dopiero zamierza zacząć sprzedawać swoje produkty/usługi.

— Metoda krytycznych czynników sukcesu służąca do analizy zasobów oraz umiejętności analizowanego przedsiębiorstwa. Głównym założeniem jest wyszczególnienie najważniejszych kryteriów dla analizowanego przedsiębiorstwa. W tej metodzie należy podkreślić różnorodność kluczowych czynników sukcesu, przez wzgląd na rodzaj działalności i branżę, w jakiej działa przedsiębiorstwo.

— Analiza portfelowa BCG jest rodzajem pozycjonowania produktu (produktów) firmy na tle konkurencyjnych marek. Pozycjonowania strategicznych jednostek biznesu dokonuje się w metodzie BCG, w odróżnieniu od pozycjonowania drogą badań marketingowych i analizą benchmarkingową [7, 11].

Dokonując podsumowania prezentowanych podejść należy zaznaczyć, że dominujący jest jakościowy charakter zaprezentowanych metod. Przedstawione procedury zarządzania strategicznego, cechuje nieformalny charakter, a wskazania końcowe posiadają ogólny i niekwantyfikowalny zakres. Istnieje zatem realna potrzeba opracowania spójnej strategii, rozpatrującej wyniki w sposób bardziej formalny, ilościowy i metodyczny.

## 2. Proponowane rozwiązanie

Badania ukierunkowano na opracowanie spójnej procedury generowania wskaźników strategicznych, a w konsekwencji planu strategicznego przedsiębiorstwa, w po-

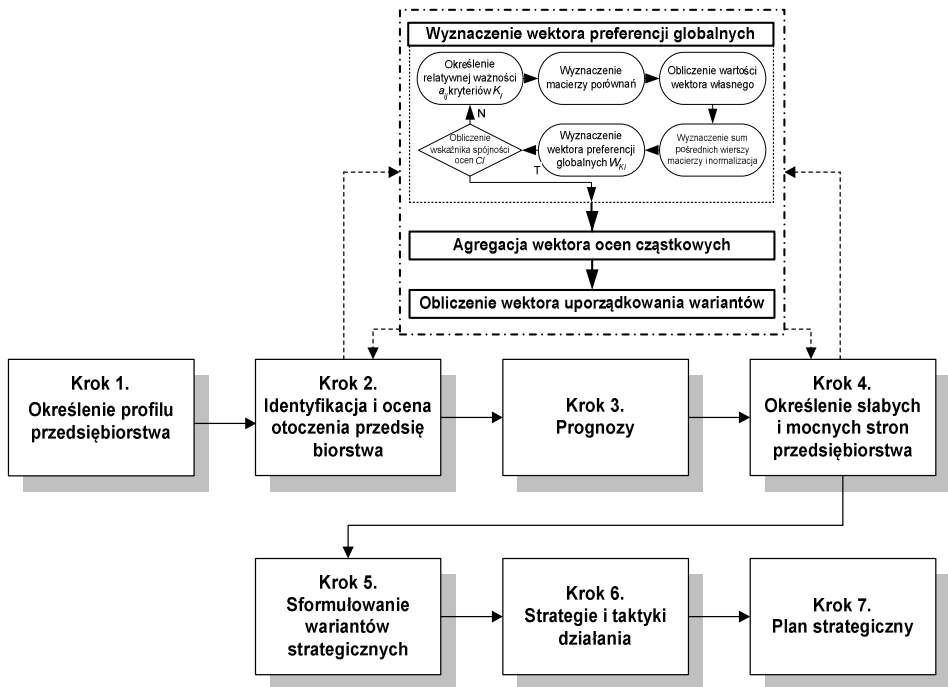


łączeniu z ich ilościową reprezentacją. W tym celu podjęto próbę połączenia procesu formalnych mechanizmów agregacji, opartych na metodzie wielokryterialnego wspomaganie decyzji z technikami doboru strategii gospodarczej. Zaproponowano zintegrowane podejście metodyczne (połączenie metody AHP [8] i analiza SWOT) pozwalające na kwantyfikację ilościową szczególnych wyników strategii gospodarczych.

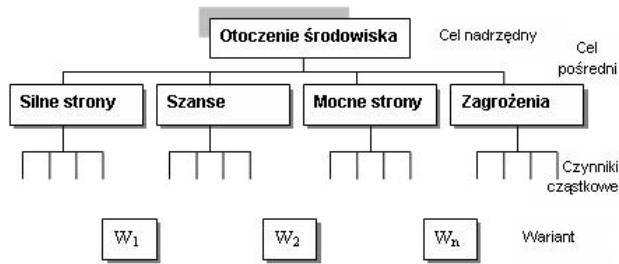
W podejściu niniejszym zawarto klasyczne kroki analizy SWOT (Rys. 2). Zawarte w proponowanym rozwiązaniu kroki analizy strategicznej są szeroko udokumentowane w literaturze przedmiotu [2, 7, 11]. Kolejno skoncentrowano się na przybliżeniu podstaw metodycznych zastosowanego dodatkowo aparatu badawczego.

W etapie drugim (analiza Szans i Zagrożeń) i etapie czwartym (analiza Słabych i Mocnych Stron) włączony został proces pozyskiwania preferencji, ustalania wartości oraz ich agregacji. Proces w szczególności oparto na konstrukcji wektora preferencji globalnych, jak również wyznaczenia użyteczności cząstkowych, każdego z czynników formułowanych strategii. W konsekwencji pozwala to na przekształcanie opisywanych strategii składowych do formy ilościowej.

Zastosowana metoda hierarchicznej analizy problemu (ang. AHP – *Analytic Hierarchy Process*), wspomaga podejmowanie decyzji w obecności wielu kryteriów



Rysunek 2. Procedura generowania wskazań strategicznych. Źródło: Opracowanie własne.



Rysunek 3. Schemat struktury hierarchicznej zadania w metodzie Saaty'ego. Źródło: Opracowanie własne.

dla dyskretnych problemów decyzyjnych [9]. Jak wskazują O. Downarowicz i in. (2000) metoda AHP ujmuje podejście wielokryterialne, oparte na kompensacyjnej strategii modelowania preferencji i przy założeniu porównywalności wariantów, uwzględniając specyfikę psychologicznych procesów wartościowania o charakterze hierarchicznym i relacyjnym. Jednocześnie metoda wspiera artykułowanie preferencji decydenta (lub oceniających) oraz wykorzystanie ich w procesie agregacji ocen [13]. Preferencje oceniających (decydujące o subiektywności ocen) stanowią istotę podejścia wielokryterialnego i są zjawiskiem naturalnym dla ocen dokonywanych przez człowieka, w odróżnieniu od pomiarów o charakterze obiektywnym [8, 9]. Wyniki otrzymane za pomocą tej metody są wartościami liczbowymi, określającymi hierarchię możliwych wariantów wg zbiorczej (wielokryterialnej) jakościowej oceny decydentów.

Występujące w modelu czynniki ryzyka, stanowią kryteria oceny wariantów pod względem wewnętrznych i zewnętrznych stron środowiska przedsiębiorstwa. Wariantami decyzyjnymi są projektowe lub realizacyjne rozwiązania, dotyczące SWOT, który w efekcie zapewnia odpowiednie zastosowanie strategii przedsiębiorstwa.

W ramach metody AHP winna być dokonana dekompozycja problemu decyzyjnego, w postaci hierarchicznej struktury decyzyjnej (Rys.3): cel nadrzędny (otoczenie środowiska), cele pośrednie (główne czynniki składające się na analizę SWOT), czynniki cząstkowe (wybrane kryteria) i warianty decyzyjne (rozwiązanie projektowe lub realizacyjne).

Zapewnia to odpowiednią ilościową i jakościową porównywalność kryteriów z tego samego poziomu, jak również poprawność logiczną i relacyjną na wszystkich poziomach struktury hierarchicznej.

### 3. Weryfikacja empiryczna proponowanej procedury

Proponowaną procedurę przedstawiono w niniejszej pracy na przykładzie przedsiębiorstwa poligraficznego. W tym celu zastosowano wyżej wymienione rozwiązanie, na podstawie przeprowadzonej serii eksperymentów badawczych, gdzie ujawniono celowość prezentowanej multimetody.

Tabela 1. Macierz SWOT alternatyw strategicznych. Źródło: opracowanie własne.

Mocne strony				Słabe strony			
	Ocena 1-5	Waga	Ocena ważona		Ocena 1-5	Waga	Ocena ważona
Doświadczenie pracowników	4	0,2	0,6	Brak znaczących inwestycji	3	0,2	0,45
Bardzo atrakcyjna oferta podtrzymująca rozpoznawalność marki na rynku	2	0,2	0,34	Brak zintegrowanego systemu informatycznego	3	0,1	0,36
Szeroka sieć odbiorców krajowych i istotny udział eksportu w dostarczaniu usług	4	0,2	0,6	Niekorzystne położenie zakładów z punktu widzenia dystrybucji wyrobów w kraju	2	0,2	0,3
Rezerwy powierzchni posiadane zakład w Szczecinie dające możliwość późniejszej rozbudowy	2	0,1	0,1	Brak możliwości rozwoju oddziału w Szczecinie z uwagi na ograniczenia powierzchni zakładu	2	0,1	0,16
Dobra, stabilna sytuacja finansowa, dająca możliwość rozwoju	3	0,1	0,23	Zbyt duże zapasy-zwiększenie kosztów magazynowania	2	0,1	0,16
Dobra opinia u klientów	3	0,1	0,39	Trudności ze ściąganiem należności	3	0,2	0,51
Zwiększenie samofinansowania przedsiębiorstwa	4	0,2	0,8	Obciążenie finansowe spowodowane koniecznością spłacania zaciągniętych kredytów	4	0,2	0,8
Wysoka jakość dostępnych ofert	3	0,1	0,23	Ograniczone środki finansowe przeznaczone na działalność promocyjną	2	0,1	0,1
<b>Suma</b>		<b>1</b>	<b>3,28</b>	<b>Suma</b>		<b>1</b>	<b>0,44</b>

Szanse				Zagrożenia			
	Ocena 1-5	Waga	Ocena ważona		Ocena 1-5	Waga	Ocena ważona
Wzrost gospodarczy	3	0,1	0,3	Zwiększenie wysokości stóp podatkowych	3	0,2	0,54
Zwiększenie popytu krajowego	3	0,2	0,51	Zwiększenie poziomu inflacji	2	0,1	0,2
Wzrost dynamiki rynku inwestycyjnego i budowlanego	2	0,1	0,26	Zwiększenie norm związanych z ochroną środowiska	2	0,1	0,2
Z momentem wejścia do Unii, możliwość pozyskania nowych rynków zbytu	5	0,2	1,1	Zmniejszenie się dynamiki rozwoju rynku poligraficznego i inwestycyjnego	3	0,2	0,51
wzrost popytu na wszystkie usługi oferowane w Polsce	4	0,2	0,8	Zmniejszenie konsumpcji krajowej i zagranicznej	4	0,2	0,8
Obniżka stóp procentowych oraz bogatsza oferta kredytowa i pożyczkowa banków	3	0,2	0,54	Napływ zagranicznej konkurencji	4	0,3	1
<b>Suma</b>		<b>1</b>	<b>3,51</b>	<b>Suma</b>		<b>1</b>	<b>3,25</b>

Tabela 2. Analiza zasobów przedsiębiorstwa przy użyciu analizy SWOT i wybranych kryteriów. Źródło: opracowanie własne.

Lp.	Wybrane kryteria	Przyznana waga (0-1)	RANKING ANALIZY SWOT = WAGA x ilość przyznaczonych pkt. (1-9)							
			Mocne Strony		Słabe Strony		Szanse		Zagrożenia	
			Pkt		Pkt.		Pkt.		Pkt.	
1.	Lokalizacja	0,15	7	1,05	3	0,45	5	0,75	5	0,75
2.	Marketing	0,2	5	1	4	0,8	5	1	3	0,6
3.	Jakość	0,3	6	1,8	4	1,2	6	1,8	6	1,8
4.	Promocja	0,35	6	2,1	5	1,75	6	2,1	5	1,75
<b>5.</b>	<b>SUMA</b>	<b>1</b>		<b>5,95</b>		<b>4,2</b>		<b>5,65</b>		<b>4,9</b>

W praktyce sklasyfikowano wszystkie czynniki mające wpływ na bieżącą i przyszłą pozycję organizacji. Istnieją dwa kryteria, które zostały sklasyfikowane: zewnętrzne w stosunku do organizacji lub mające charakter uwarunkowań wewnętrznych oraz wywierające wpływ negatywny lub pozytywny na organizację. Ze skrzyżowania tych dwóch podziałów powstają cztery kategorie SWOT (Tabela 1).

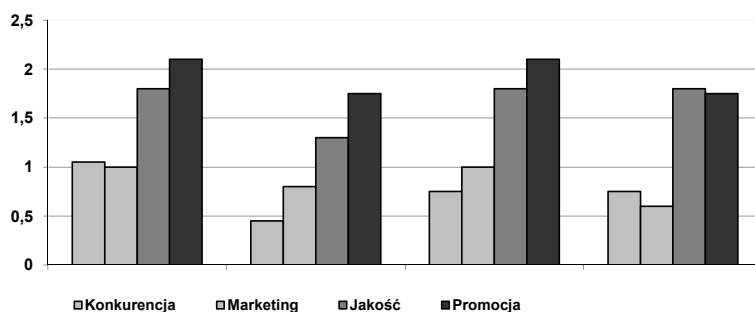
Kolejno zdefiniowano strukturę hierarchii, odzwierciedlającej problem. Do oceny analizy SWOT przedsiębiorstwa poligraficznego wystarcza przyjęcie hierarchicznego modelu o podstawowej strukturze, składającej się z najwyżej czterech poziomów czyli celu nadrzędnego, celu głównego, czynników cząstkowych oraz wariantów decyzyjnych.

W modelu hierarchicznym otoczenie środowiska przedsiębiorstwa jest skutkiem zgodnym z określonymi celami cząstkowymi, jakie zostały opisane przez zdefiniowane atrybuty oraz związane z nimi kryteria oceny. System kryteriów oceny całego otoczenia środowiska, w którym zostało przedstawione podejście analityczne poprzez metodę SWOT ma zatem strukturę hierarchiczną.

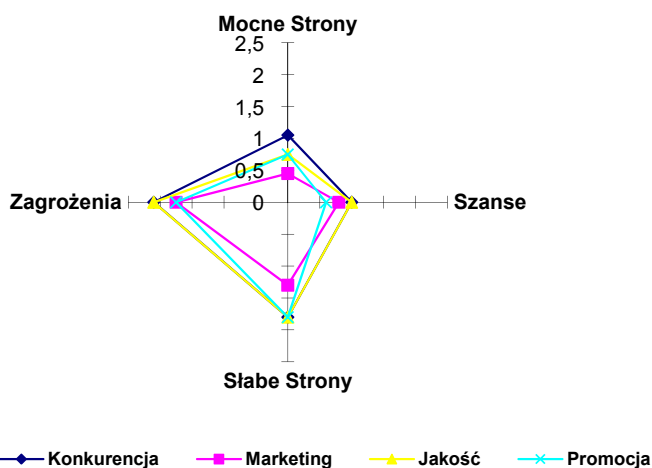
Dla celów kolejnych analiz wyznaczono ważność wybranych alternatywnych wariantów strategii gospodarczej. Wyznaczony tak wektor priorytetów kolejno znormalizowano, co pozwoliło na opracowanie ilościowej reprezentacji i rankingu w metodzie SWOT. Wyniki badań przedstawiono w Tabeli 1 i Rysunku 4 i 5.

Przez agregacja ocen w ramach struktury hierarchicznej analizy SWOT w odpowiednim otoczeniu środowiskowym stanowi podstawę do uogólnienia wyników w postaci tabeli danych zagregowanych (Tabela 2), która jest postawą utworzenia zredukowanej struktury wykresu, poddającej oceny końcowe wyznaczone dla kryteriów głównych, czynników i wariantów [2], 13].

Interpretując wyniki powyższych analiz można stwierdzić, że w większości przypadków firma kontaktuje się ze środowiskiem neutralnym. Może to prowadzić do wniosków, że aby osiągnąć wszystkie założone cele, będzie potrzebny odpowiedni nakład pracy i środków. Z przeprowadzonej analizy przedsiębiorstwa poligraficznego wynika, że firma powinna zastosować odpowiednie strategie: koncentracji, rozwoju rynku, rozwoju produktu oraz innowacji. Organizacja może przyjąć jedną



Rysunek 4. Ogólne pierwszeństwo alternatywnych strategii w analizie SWOT. Źródło: opracowanie własne.



Rysunek 5. Ukazanie czynników wybranych kryteriów za pomocą pierwszeństwa wag w przedsiębiorstwie poligraficznym. Źródło: opracowanie własne.

z tych strategii lub zastosować wszystkie naraz, gdyż uzupełniają się one nawzajem. Stosowanie się do powyższych strategii pozwoli firmie zdobyć lepszą pozycję rynkową. Środowiska ekonomiczne, społeczne i technologiczne nie będą w stanie pozytywnie wpłynąć na działalność firmy. Jak wynika z danych, firma jest zorganizowana w sposób zapewniający jej sprawne istnienie na rynku, a także jest odpowiednio zabezpieczona przed konkurencją.

#### 4. Podsumowanie

Przedstawiona w niniejszej pracy procedura badawcza obejmuje analizę SWOT w przedsiębiorstwie. Aparat wielokryterialnego wspomaganie decyzji, stanowiąc część diagnostyczną projektowanych strategii innowacyjności analizy SWOT. Zastosowana metoda AHP posłużyła do kwantyfikacji ilościowej poszczególnych

składników strategii, co stanowi bardzo intuicyjny i wygodny aparat matematyczny. Warto jednocześnie zaznaczyć, że prezentowany zestaw metod cechuje komplementarny charakter. Wykorzystana w pracy koncepcja definiowania preferencji pozwala na łatwe powiązanie wyników badań z określeniem wartości preferencji porównywanych wariantów według przyjętych kryteriów. Uogólniając wyniki badań należy wskazać, że prezentowane rozwiązanie może stanowić efektywne narzędzie analityczne wykorzystywane w procesie zarządzania strategicznego.

## Bibliografia

- [1] Bengt K. *Strategia biznesu. Koncepcje i modele*. Przewodnik. Biblioteka Menadżerska i Bankowa Warszawa 1994.
- [2] Gierszewska G., Romanowska M. *Analiza strategiczna przedsiębiorstwa*. PWE, Warszawa 1995.
- [3] Koźmiński A.K., Piotrowski W. *Zarządzanie – teoria i praktyka*. PWN, Warszawa 1996.
- [4] Koziński J. *Podjęcie decyzji*. PWN, Warszawa 1995.
- [5] Machaczka J. *Podstawy zarządzania*. AE Kraków, Kraków 2001.
- [6] Nogalski B., Rybicki J., Gacek-Bielec J. *Modele analizy portfelowej. Teoria i praktyka*. Oficyna Wydawnicza OPO, Bydgoszcz 1996.
- [7] Pierścionek Z. *Strategie rozwoju firmy*. PWN, Warszawa, 2007
- [8] Saaty, T.L. *The Analytic Hierarchy Process*. MacGraw Hill 1980.
- [9] Saaty, T.L., Vargas, L.G. *The Analytic Hierarchy Process Series*. University of Pittsburgh 1990.
- [10] Stabryła A. *Podstawy zarządzania firmą*. PWN, Warszawa 1995.
- [11] Stabryła A. *Zarządzanie strategiczne w teorii i praktyce firmy*. PWN, Warszawa 2000.
- [12] Tyszka T. *Analiza decyzyjna i psychologia decyzji*. PWN, Warszawa 1986.

# Środowisko programistyczne dla rozproszonych minisystemów kontrolno-pomiarowych

Dariusz Rzońca, Jan Sadolewski, Andrzej Stec, Zbigniew Świder,  
Bartosz Trybus, Leszek Trybus

*Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki*

## Abstract:

*An CPDev environment for programming small distributed control-and-measurement systems in ST language (Structured Text) of IEC 61131-3 standard is presented. The environment involves ST compiler, configurer of hardware resources and simulator. The compiler generates an universal code that can be executed on different platforms by Java-like virtual machines. The environment is open in terms of software and hardware, what means that the programmer can create his own libraries, and I/O and communication drivers can be written by hardware designers and linked with the universal code. It is possible to select a number of data types for particular applications. The environment has been created in C# on MS .NET Framework platform. First application for programming a distributed mini-system from LUMEL Zielona Góra is described.*

## Słowa kluczowe:

*compiler development, ST language, IEC 61131-3 standard, distributed system, simulation, control-and-measurement*

## 1. Wprowadzenie

Zakres produkcji krajowego przemysłu aparatury kontrolno-pomiarowej obejmuje przetworniki pomiarowe, urządzenia wykonawcze, falowniki, regulatory PID, sterowniki wielofunkcyjne i dedykowane, rejestratory, wyświetlacze HMI itd. Po połączeniu w systemy rozproszone mogą one służyć do automatyzacji procesów średniej wielkości. Narzędzia inżynierskie przeznaczone do programowania tych urządzeń są jednak dość ubogie i nie odpowiadają normie IEC 61131-3 obowiązującej w Polsce od 2004 r. [1]. Ogranicza to skuteczność konkurencji z aparaturą importowaną. Problem ten można byłoby częściowo rozwiązać opracowując uniwersalne, otwarte środowisko inżynierskie<sup>13</sup> przeznaczone do programowania urządzeń kontrolno-pomiarowych, szczególnie niewielkich sterowników i regulatorów, według normy IEC 61131-3 (dalej oznaczanej skrótem IEC).

---

<sup>13</sup> Praca realizowana w ramach projektu rozwojowego MNiSzW nr R02 058 03.

Norma IEC definiuje pięć języków programowania – LD, IL, FBD, ST i SFC, pozwalając dobrać język odpowiadający danemu problemowi z uwzględnieniem umiejętności projektanta. Lista rozkazów IL oraz tekst strukturalny ST są językami tekstowymi, a schematy drabinkowy LD, blokowy FBD oraz sekwencyjny SFC graficznymi (SFC nie jest językiem samodzielnym, bo wymaga fragmentów napisanych w innych językach). Relatywnie proste języki LD i IL stosuje się w mniejszych aplikacjach. FBD, ST i SFC występują w aplikacjach średnich i dużych. Dobrym źródłem dotyczącym programowania według normy IEC jest niedawno wydana książka Kasprzyka [2].

Tekst strukturalny ST jest językiem wysokiego poziomu wywodzącym się z Pascala, Ady i C, nadającym się szczególnie do zapisu skomplikowanych algorytmów (np. automatycznego strojenia pętli regulacyjnych). W ST można napisać odpowiednik każdego programu utworzonego w którymkolwiek z pozostałych języków, ale nie odwrotnie. W większości pakietów inżynierskich ST jest więc językiem „domyślnym” do programowania własnych bloków funkcjonalnych użytkownika. Z tych względów przyjęto, że w środowisku programistycznym omawianym tutaj język ST będzie językiem bazowym. Po uwzględnieniu pewnych ograniczeń [2], do ST będą konwertowane programy napisane w pozostałych językach (trwają prace nad edytorem języka FBD).

Artykuł niniejszy dokumentuje aktualny stopień zaawansowania prac nad prototypem środowiska do programowania sterowników i innych urządzeń kontrolno-pomiarowych według normy IEC. Pierwsze informacje na ten temat autorzy przedstawili w referatach [3, 4]. Środowisko to o nazwie CPDev (*Control Program Developer*) jest budowane na platformie Microsoft .NET Framework 2.0 [5]. Podstawowym założeniem postawionym na początku było, aby środowisko CPDev było:

- *uniwersalne* w sensie przeznaczenia dla różnych platform sprzętowych,
- *otwarte* zarówno dla projektantów oprogramowania jak i konstruktorów urządzeń,
- *elastyczne* poprzez swobodny dobór typów danych dla konkretnych aplikacji.

Wymagania te sprecyzowano bliżej w sekcji 2 w ramach charakterystyki środowiska CPDev. Po krótkim wprowadzeniu do języka ST, w sekcji 3 przedstawiono środowisko, a w tym interfejs projektanta z przykładowym programem, zmienne globalne oraz funkcje i bloki biblioteczne. Składniki kompilatora, tj. skaner, parser i generator kodu, są omówione w sekcji 4 wraz z klasami obiektów występujących w kompilacji. Przykład kompilacji programu standardowego licznika poczynając od kodu źródłowego w ST, a kończąc na wykonywalnym kodzie binarnym zawiera sekcja 5. W sekcji 6 przedstawiono realizację maszyny wirtualnej oraz pierwszą aplikację CPDev, jaką jest minisystem rozproszony z Zakładów LUMEL w Zielonej Górze. Konfiguracja komunikacji w tym systemie oraz symulator są opisane w sekcji 7.



## 2. Charakterystyka środowiska CPDev

### 2.1. Cechy ogólne

Podstawowym zadaniem pakietów inżynierskich do programowania systemów kontrolno-pomiarowych jest kompilacja programu napisanego w jednym z języków IEC na kod maszynowy przeznaczony dla konkretnej platformy sprzętowej (procesora). Zmiana platformy wymaga nowego kompilatora. Natomiast pakiet, jeżeli ma być *uniwersalny*, powinien generować kod wykonywalny na różnych platformach, przede wszystkim takich jak AVR, MCS-51, ARM czy PC, gdy chodzi o urządzenia krajowe. Wykonywanie takiego uniwersalnego kodu musi jednak prowadzić interpreter zastępujący instrukcje kodu rozkazami języka maszynowego. Każda z platform powinna mieć własny interpreter. Uniwersalny kod wykonywalny należy więc rozumieć jako pewien kod pośredni, na który najpierw jest tłumaczony oryginalny program napisany w języku IEC. Ogólnie biorąc odpowiada to koncepcji maszyn wirtualnych Javy wykonujących programy na różnym sprzęcie [6]. Dlatego interpretery uniwersalnego kodu, o którym mowa w dalszej części, będą tutaj nazywane *maszynami wirtualnymi*.

Środowisko inżynierskie można uznać za *otwarte* pod względem programowym i sprzętowym, jeżeli dostępne są:

- narzędzia do tworzenia własnych bloków funkcjonalnych będących jednostkami oprogramowania przeznaczonymi do wielokrotnego użycia,
- specyfikacje interfejsów wejścia/wyjścia i komunikacyjnych definiujące ogólne prototypy procedur obsługi interfejsów dla różnych platform i rozwiązań układowych.

Procedury te mogą oczywiście różnić się wewnętrzną treścią odpowiednio do platformy i rozwiązań układowych, ale formy ich wywołań powinny pozostać identyczne.

O *elastyczności* środowiska mówi się wówczas, jeżeli spośród wszystkich typów danych zdefiniowanych w normie IEC można swobodnie wybrać te, które odpowiadają danej aplikacji. Nieco innych typów wymagają bowiem regulator PID, mikrosterownik PLC i programowalna tablica świetlna.

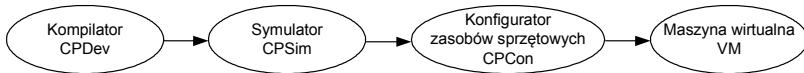
Tak rozumiane wymagania zostały pierwotnie postawione przed środowiskiem CPDev.

### 2.2. Składniki środowiska

W skład środowiska CPDev, nazywanego także pakietem, wchodzi cztery programy pokazane na rysunku 1. Po stronie komputera PC (czyli stacji inżynierskiej) są to:

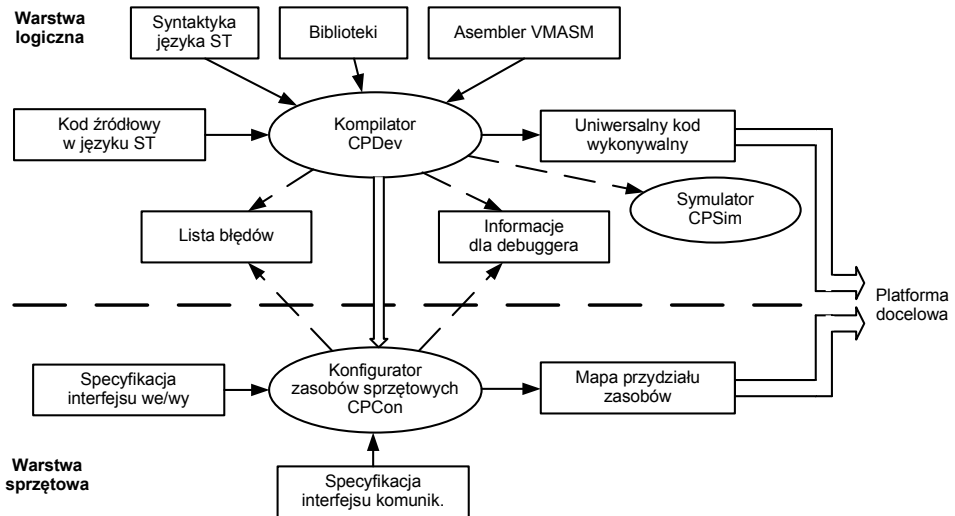
- kompilator CPDev języka ST,
- symulator oprogramowania CPSim,
- konfigurator zasobów sprzętowych CPCCon.

Programy wymieniają dane w formie plików. Kompilator CPDev (jest to zarazem nazwa całego środowiska) generuje uniwersalny kod wykonywalny, który po stronie sterownika wykonuje maszyna wirtualna VM (*Virtual Machine*), pracująca jako interpreter. Kod wykonywalny jest listą elementarnych instrukcji języka tej maszyny nazywanego asemblerem VMASM (*VM Assembler*). Nie jest on związany z żadnym procesorem, choć bliski nieco rozszerzonemu typowemu asemblerowi. Krótką charakterystykę VMASM podano w sekcjach 4, 5. Symulator CPSim zawiera również maszynę wirtualną, a więc w tym przypadku funkcjonuje ona po stronie komputera PC.



Rysunek 1. Składniki środowiska CPDev

Powiązanie programów CPDev, CPSim i CPCCon ilustruje rysunek 2 z wyróżnionymi warstwami logiczną i sprzętową. Rozdzielenie kompilacji programu od konfiguracji sprzętu ułatwia generowanie kodu dla różnych platform. W warstwie logicznej program napisany w języku ST jest kompilowany na uniwersalny kod wykonywalny. Wykorzystuje się do tego reguły syntaktyczne języka ST, listę instrukcji elementarnych asemblera VMASM oraz funkcje, bloki funkcjonalne i programy zawarte w bibliotekach. Oprócz uniwersalnego kodu tworzone są informacje służące do śledzenia i symulacji programu. Adresy zmiennych deklarowane w ST są na razie adresami względnymi, dalej nazywanymi lokalnymi. Wystarczają one symulatorowi CPSim do symulacji samego algorytmu.

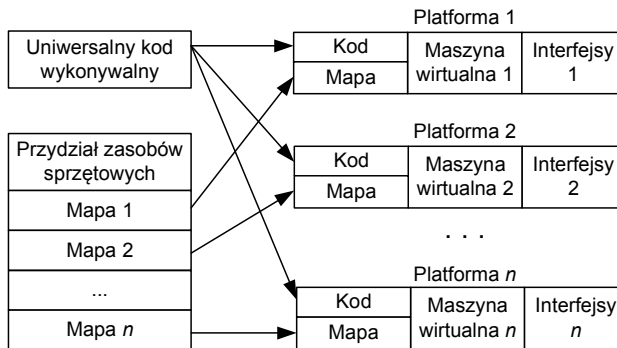


Rysunek 2. Powiązanie składników środowiska CPDev

Konfiguracja zasobów sprzętowych konfiguratorem CPCon stanowiąca treść warstwy drugiej dotyczy pamięci oraz interfejsów wejścia/wyjścia i komunikacyjnych. Specyfikacja obejmuje typy i obszary pamięci, liczby i typy wejść/wyjść oraz kanałów komunikacyjnych, adresy fizyczne, sygnalizację poprawności/niepoprawności itp. Przydział zasobów sprzętowych ma postać *mapy* wiążącej adresy lokalne z adresami fizycznymi (jest to pewnego rodzaju tabela). Maszyna wirtualna platformy docelowej otrzymawszy kod i mapę przydziału zasobów jest w stanie prowadzić wymagane obliczenia.

### 2.3. Tworzenie oprogramowania dla różnych platform

Z punktu widzenia środowiska CPDev odmienne platformy sprzętowe różnią się faktycznie tylko mapą przydziału zasobów, a nie kodem wykonywalnym. Może on pozostać taki sam<sup>14</sup> i dlatego jest nazywany uniwersalnym. Tryb tworzenia oprogramowania dla różnych platform przedstawia rysunek 3. Maszyna wirtualna funkcjonująca jako interpreter jest dedykowana dla konkretnego procesora. Na razie opracowano maszyny dla platform AVR, MCS-51 i PC (jest ona zasadniczą częścią symulatora CPSim). Jako następną rozpatruje się ARM 7. Tworząc maszyny dołożono starań, aby narzut czasowy na interpretację kodu wykonywalnego był możliwie niewielki. Wynika to z podobieństwa asemblera VMASM do języków maszynowych oraz mechanizmów indeksowania zastosowanych do interpretacji instrukcji [3].



Rysunek 3. Tryb tworzenia oprogramowania dla różnych platform sprzętowych

## 3. Środowisko programistyczne

### 3.1. Przegląd języka ST

Jak wspomniano na początku, język ST jest najwszechstronniejszym spośród pięciu języków normy IEC. Wspólnymi elementami języków są nazwy (identyfika-

<sup>14</sup> Chyba, że dokonuje się optymalizacji na dany procesor (podsekcja 5.3).

tory), typy danych, literały (stałe) i zmienne. Nazwy mają postać standardową, ale nie rozróżnia się między małymi, a dużymi literami. Typy elementarne, których jest 20, są wymienione w tabeli 1 wraz z rozmiarami pamięci zarezerwowanymi w pakiecie CPDev i z zakresami wartości (dla połowy). Najczęściej stosowanymi typami są `BOOL`, `INT`, `REAL` i `TIME`. Przykładami stałych tych typów są `FALSE`, `13`, `-4.1415`, `T#1m3s` (rozdzielczością `TIME` jest `1ms`).

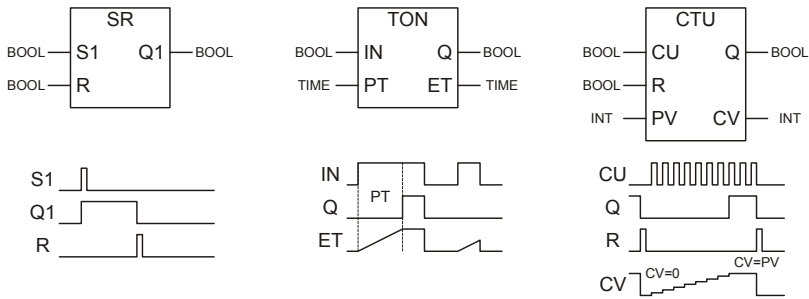
Norma definiuje trzy zakresy dostępności zmiennych – `LOCAL`, `GLOBAL` i `ACCESS`. Zmienne `LOCAL` są dostępne tylko w ramach programu lub w bloku funkcjonalnym. Zmienne `GLOBAL` można używać w całym projekcie z tym, że w programie muszą być zadeklarowane jako `EXTERNAL`. Zmienne `ACCESS` są wymieniane z innymi systemami (na razie nie ma ich w CPDev).

Według normy IEC podstawowymi elementami strukturalnie czytelnego oprogramowania są funkcje, bloki funkcjonalne i programy. Kluczowe znaczenie mają bloki funkcjonalne przeznaczone do wielokrotnego użycia w różnych częściach programu (*reuse*). Blok zawiera zmienne wejściowe i wyjściowe o odpowiedniej strukturze, a ponadto pamięć na dane z poprzednich wykonań. Norma definiuje niewielki zestaw standardowych bloków, którymi są elementy dwustanowe, detektory zbrocza, czasomierze i liczniki (tabela 3 dalej). Trzy z nich pokazano na rysunku 4.

Programy napisane w ST rozpoczynają się od deklaracji zmiennych i egzemplarzy bloków funkcjonalnych (instancji) zawartych między `VAR` a `END_VAR`. Po nich następuje lista instrukcji. Instrukcje zawierają wyrażenia, które po obliczeniu dają wartość w jednym ze zdefiniowanych typów – elementarnym lub pochodnym, np. strukturalnym. Operatorami występującymi w wyrażeniach są (według ma-

Tabela 1. Elementarne typy danych normy IEC 61131-3 oraz ich implementacja w pakiecie CPDev

Typ	Rozmiar i zakres	Typ	Rozmiar
SINT	1B (-128 .. 127)	BOOL	1B (0, 1)
INT	2B (-32768 .. 32767)	BYTE	1B
DINT	4B (-231 .. 231 - 1)	WORD	2B
LINT	8B (-263 .. 263 - 1)	DWORD	4B
USINT	1B (0 .. 255)	LWORD	8B
UINT	2B (0 .. 65535)	TIME	4B
UDINT	4B (0 .. 232 - 1)	DATE	4B
ULINT	8B (0 .. 264 - 1)	TIME_OF_DAY	4B
REAL	4B, format IEEE-754	DATE_AND_TIME	8B
LREAL	8B, format IEEE-754	STRING	Zmienna długość



Rysunek 4. Przykłady bloków standardowych wraz z przebiegami dynamicznymi: przerzutnik SR, czasomierz włączający TON, licznik inkrementujący CTU

lejących priorytetów): nawiasy, obliczenie wartości funkcji, negacje, potęgowanie, operatory arytmetyczne, porównania, operatory boolowskie.

Język ST udostępnia pięć rodzajów instrukcji:

- podstawienie `:=` (symbol z Pascala),
- wybór `IF`, `CASE`,
- pętle `FOR`, `WHILE`, `REPEAT`,
- sterowanie `RETURN`, `EXIT`, `END`,
- wywołanie bloku funkcjonalnego.

W tabeli 2 podano tekst programu dla licznika CTU z rysunku 4. CU jest wejściem inkrementującym (*Count Up*), a R zerującym (*Reset*). PV reprezentuje wartość zadaną (*Preset Value*), po osiągnięciu której wyjście Q jest ustawiane na TRUE. Wyjście CV reprezentuje aktualną wartość licznika (*Current Value*). Deklarując LICZNIK: CTU jako egzemplarz bloku CTU wywołuje się go w programie i wykorzystuje wyjścia za pomocą następujących przykładowych instrukcji:

```
LICZNIK (CU:=INKREMENTUJ,R:=ZERUJ,PV:=GRANICA);
PELNY := LICZNIK.Q;
STAN := LICZNIK.CV;
```

### 3.2. Interfejs projektanta

Na rysunku 5 pokazano główne okno interfejsu projektanta w pakiecie CPDev. Składa się ono z trzech pól:

- drzewo struktury projektu, po lewej stronie,
- program w języku ST, główne środkowe pole,
- lista komunikatów, na dole.

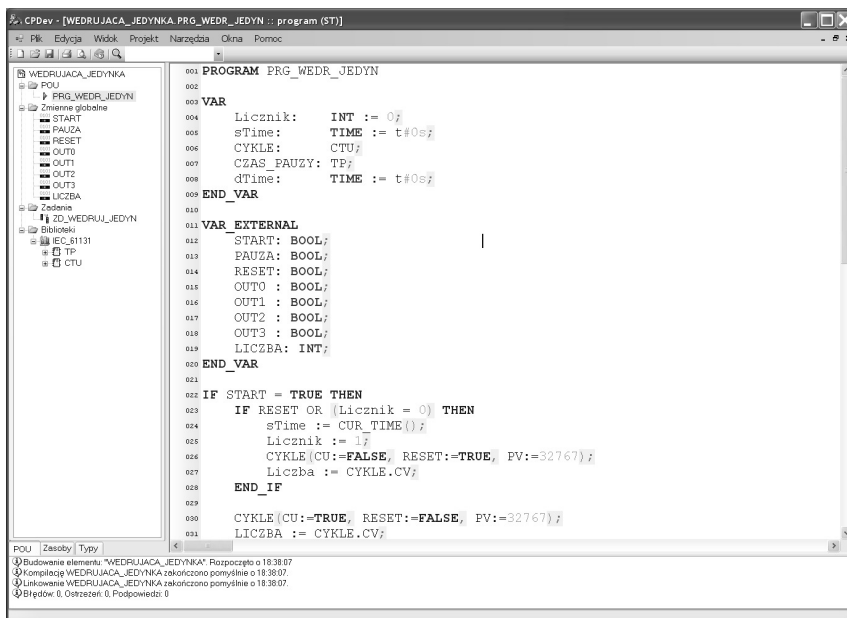
Drzewo projektu WEDRUJACA\_JEDYNKA z rysunku 5 zawiera jednostkę POU – program PRG\_WEDR\_JEDYN, osiem zmiennych globalnych START, PAUZA, RESET, OUT0, OUT1, OUT2, OUT3 i LICZBA, zadanie ZD\_WEDR\_JEDYN oraz dwa bloki funkcjonalne TP i CTU ze standardowej biblioteki IEC\_61131.

Tabela 2. Program standardowego licznika CTU w języku ST

```

FUNCTION_BLOCK CTU          (* CTU - Licznik "w górę" *)
VAR_INPUT
    CU : BOOL;              (* wejście inkrementujące *)
    RESET : BOOL;          (* zerowanie *)
    PV : INT;               (* wartość zadana *)
END_VAR
VAR_OUTPUT
    Q : BOOL;              (* wyjście sygnalizacyjne *)
    CV : INT;              (* wartość bieżąca *)
END_VAR
VAR
    CUp : BOOL := FALSE;   (* poprzednie wejście CU *)
END_VAR
IF RESET THEN              (* jeśli zerowanie *)
    CV := 0;
ELSE
    IF (CU AND NOT CUp) THEN (* jeśli zbocze narastające na wejściu CU *)
        IF (CV < PV) THEN
            CV := CV + 1;
        END_IF
    END_IF
END_IF
Q := CV >= PV;             (* Q := TRUE, jeśli CV >= PV *)
CUp := CU;                 (* poprzednia wartość CU *)
END_FUNCTION_BLOCK

```

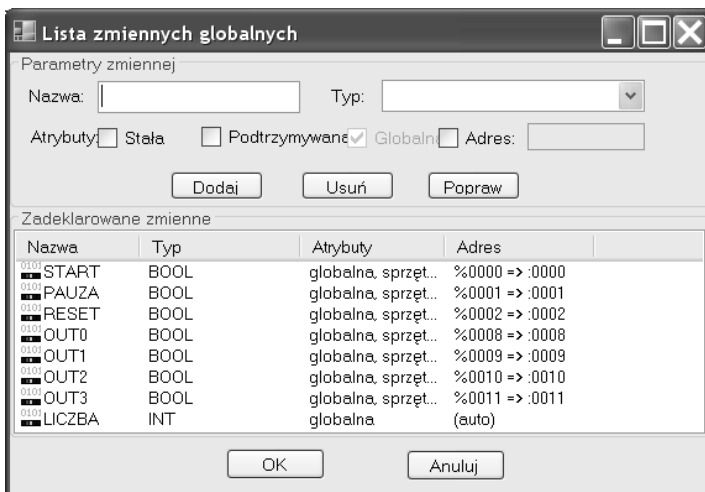


Rysunek 5. Interfejs projektanta w pakiecie CPDev – projekt WEDRUJACA\_JEDYNKA

Program PRG\_WEDR\_JEDYN widoczny w głównym polu jest napisany według reguł syntaktycznych języka ST. Pierwszą część stanowią deklaracje zmiennych lokalnych Licznik, sTime i dTime oraz egzemplarze bloków funkcjonalnych CYKLE:CTU i CZAS\_PAUZY:TP. Drugą częścią są deklaracje zmiennych globalnych (EXTERNAL), a trzecią instrukcje programu. Instrukcje nie będą tu jednak szczegółowo omawiane. Wyjaśnimy tylko, że po pojawieniu się sygnału START, co 2 sekundy na kolejnych wyjściach OUT0 do OUT3 pojawia się wartość TRUE (*wędrująca jedynka*). Zmienna Licznik wskazuje, o które z wyjść chodzi. Pojawienie się sygnału PAUZA wstrzymuje wędrowanie jedynki na 10 sekund, ale po ich upływie wędrowanie zostaje wznowione. Przerwę 10-sekundową odmierza blok CZAS\_PAUZY. Licznik CYKLE o wyjściu LICZBA zlicza liczbę cykli (obiegów) od momentu startu. Pojawienie się sygnału STOP powoduje powrót do stanu wyjściowego (łącznie z zerowaniem licznika).

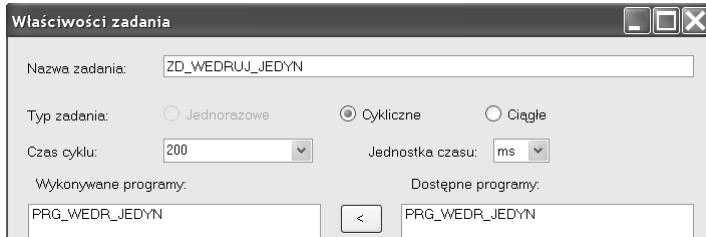
### 3.3. Zmienne globalne, zadanie

Zmienne globalne deklaruje się w CPDev bądź za pomocą indywidualnych okien, bądź zbiorczo na liście zmiennych. Listę tę dla projektu WEDRUJACA\_JEDYNKA pokazano na rysunku 6. Atrybuty Stała i Podtrzymywana reprezentują CONSTANT i RETAIN z normy IEC. Adresy dotyczą tzw. *zmiennych reprezentowanych bezpośrednio* [2] i określają względną lokalizację w pamięci sterownika (deklarując zmienną w indywidualnym oknie używa się słowa kluczowego AT). Adresy te są tutaj nazywane adresami *lokalnymi*. Jak podano poprzednio, powiązania adresów lokalnych z fizycznymi dokonuje mapa przydziału zasobów generowana przez konfigurator CPCon (podsekcja 7.1). Zmienne, którym nie przypisano adresów, rozmieszczane są przez kompilator automatycznie.



Rysunek 6. Lista zmiennych globalnych projektu WEDRUJACA\_JEDYNKA

Okno deklaracji zadania ZD\_START\_STOP pokazano na rysunku 7. Zadanie może być wykonywane jednorazowo, cyklicznie lub w sposób ciągły (tzn. uruchamiane natychmiast po zakończeniu, jak w małych sterownikach PLC). Nie ma ograniczeń na liczbę programów przyporządkowanych zadaniu, ale dany program można przyporządkować tylko raz.



Rysunek 7. Deklaracja zadania ZD\_WEDRUJ\_JEDYN

Tekst projektu reprezentowanego przez drzewo z rysunku 5 jest zapisywany w pliku tekstowym w formacie XML. Kompilację przeprowadza się wywołując z menu głównego **Projekt**→**Buďuj**. Komunikaty pojawiają się w dolnym polu. Jeźeli nie ma błędów, projekt jest zapisywany w plikach w postaci skompilowanej (podsekcja 4.1).

### 3.4. Funkcje i bloki biblioteczne

Kompilator CPDev udostępnia większość standardowych funkcji zdefiniowanych w normie IEC. W lewej części tabeli 3 podano grupy tych funkcji łącznie z przykładami. Argumentami większości funkcji liczbowych, a także funkcji wyboru i porównania, może być dowolny z typów liczbowych, czyli INT, DINT, UINT i REAL (nazywany ANY\_NUM w IEC [2]).

Praktycznie w każdym programie napisanym w języku ST występują wywołania bloków funkcjonalnych, przy czym wyjścia bloków poprzednich są wejściami następných. W pakiecie CPDev dostępne są dwie biblioteki:

- biblioteka standardu IEC\_61131,
- biblioteka Basic\_blocks z prostymi blokami uzupełniającymi standard.

Bloki biblioteki IEC\_61131 podano w prawej części tabeli 3. W drugiej bibliotece znajdują się bloki o podobnym stopniu komplikacji jak bloki standardowe, z grubsza podobne do tych, jakie można spotkać w małych wielofunkcyjnych sterownikach krajowych i zagranicznych (ZPDA Ostrów Wlkp., Siemens, ABB).

W uzupełnieniu tych dwu bibliotek projektant może tworzyć własne funkcje, bloki funkcjonalne i programy gromadząc je w swoich bibliotekach. Tworzenie bloku funkcjonalnego rozpoczyna się od wyboru z menu odpowiedniej pozycji, w wyniku czego pojawia się okno szablonu z napisem FUNCTION\_BLOCK. Tworząc funkcję lub blok można korzystać ze wszystkich dołączonych bibliotek.



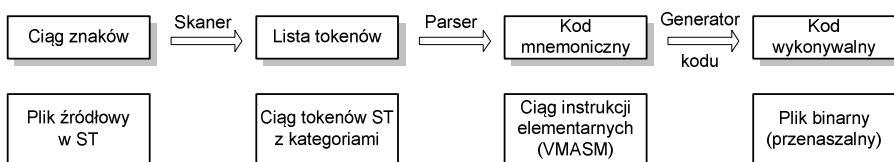
Tabela 3. Standardowe funkcje i bloki normy IEC 61131-3 dostępne w pakiecie CPDev

Funkcje	Bloki
<i>Konwersje typów</i> INT_TO_REAL, TIME_TO_DINT, TRUNC	<i>Dwustanowe</i> przerzutniki RS, SR semafor SEMA
<i>Funkcje liczbowe</i> ADD, SUB, MUL, DIV, SQRT, ABS, LN	<i>Liczniki</i> w górę CTU w dół CTD zamiennie CTUD
<i>Funkcje boolowskie i bitowe</i> AND, OR, NOT, SHL, ROR	
<i>Wybór i porównanie</i> SEL, MAX, LIMIT, MUX, GE, EQ, LT	<i>Detektory zbrocza</i> narastające R_TRIG opadające F_TRIG
<i>Operacje na ciągach znaków</i> LEN, LEFT, CONCAT, INSERT	<i>Czasomierze</i> generator impulsu TP opóźnione załącz., wyłącz. TON, TOF zegar czasu rzeczywistego RTC
<i>Operacje na typach czasowych</i> ADD, SUB, MUL, DIV	

## 4. Składniki kompilatora

### 4.1. Skaner, parser i generator kodu

Zadaniem kompilatora CPDev jest zamiana pliku XML z programem projektu w języku ST na plik z uniwersalnym kodem wykonywalnym w postaci binarnej. Ogólny schemat funkcjonowania kompilatora, w skład którego wchodzi skaner, parser i generator kodu pokazano na rysunku 8.



Rysunek 8. Schemat funkcjonowania kompilatora

Skaner (analyzer leksykalny) analizuje ciąg znaków źródłowego pliku ST dzieląc go na podstawowe jednostki leksykalne – tokeny (leksemy). Każdy token jest kwalifikowany do odpowiedniej kategorii, z których najważniejsze podano w tabeli 4. Tokeny wraz ze swymi kategoriami są gromadzone na liście przekazywanej do parsera.

Parser funkcjonuje według metody zstępującej z translacją sterowaną składnią [7]. Korzystając z reguł języka ST rozpoznaje kolejne konstrukcje tokenowe w otrzymanej liście. Pomija białe znaki i komentarze. Rozpoznawszy kolejną konstrukcję zastępuje ją zestawem instrukcji kodu mnemonicznego assemblera VMASM

Tabela 4. Kategorie tokenów rozpoznawane przez skaner

Kategoria	Przykład	Kategoria	Przykład
identyfikator	PRG1	stała całkowita	50
słowo kluczowe	FUNCTION	operator	+
stała typowana	DINT#1722211	ogranicznik	,
komentarz	(* zerowanie *)	dyrektywa	(*\$READ*)
stała rzeczywista	18.32	biały znak	
stała łańcuchowa	'Temperatura'	błędny znak	\

maszyny wirtualnej. Korzysta przy tym z wbudowanych typów elementarnych języka ST (tabela 1) oraz listy instrukcji elementarnych asemblera VMASM. Przykłady tych instrukcji podano w tabeli 5. Niektóre przekształcenia wymagają wprowadzenia zmiennych pomocniczych i etykiet (podsekcja 5.2).

Pochodne typy danych, funkcje i bloki funkcjonalne dołącza się z bibliotecznymi plikami XML. Kod mnemoniczny skompilowanego programu jest zapisywany w specjalnym formacie tekstowym. Może on zostać skonsolidowany z innymi kodami mnemonicznymi.

W trzecim kroku generator kodu przetwarza skonsolidowany kod na kod wykonywalny w postaci binarnej. Polega to na zamianie mnemoniców instrukcji elementarnych oraz nazw zmiennych i etykiet przez odpowiednie identyfikatory liczbowe. W tym celu generator korzysta z bibliotecznego pliku konfiguracyjnego LCF

Tabela 5. Podstawowe instrukcje elementarne maszyny wirtualnej

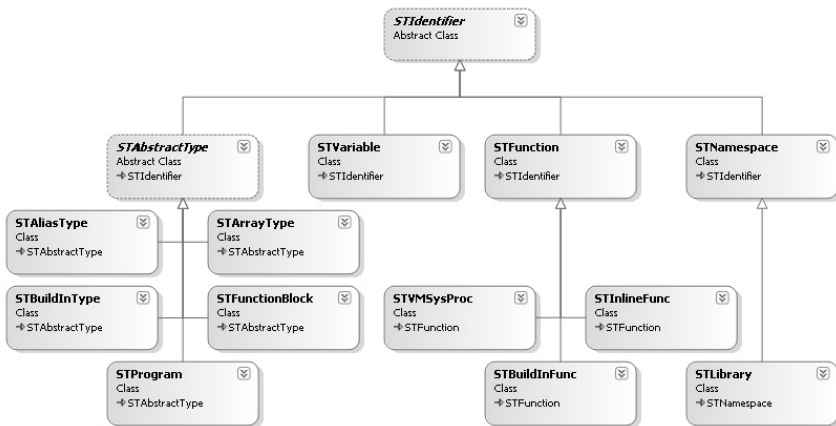
Instrukcja	Znaczenie	Oper. ST	Instrukcja	Znaczenie
ADD	suma	+	OR	suma logiczna
SUB	różnica	-	XOR	różnica symetryczna
MUL	iloczyn	*	NOT	negacja bitowa
DIV	iloraz	/	MCD	inicjacja stałej wartością
NEG	negacja	-	MEMCP	podstawienie (przypis.)
EXPT	potęga	**	SHL/SHR	przesunięcie bitowe
GT	większy	>	ROL/ROR	obrót bitowy
LT	mniejszy	<	JMP	skok bezwarunkowy
GE	większy lub rów.	>=	JZ	skok warunkowy
LE	mniej. lub rów.	<=	JNZ	skok warunkowy
EQ	równy	=	JR	skok bezwar. relatywny
NE	różny	<>	JRN	skok war. relatywny
CONCAT	konkatenacja	+	JRZ	skok war. relatywny
AND	iloczyn logiczny	&	RETURN	powrót z funkcji

(*Library Configuration File*, format XML), w którym zawarte są liczbowe identyfikatory instrukcji, liczby i typy operandów oraz sposoby ich przekazywania. Identyfikator operandu może oznaczać wskaźnik do zmiennej lub wartość bezpośrednią. Każdą implementację maszyny wirtualnej określa odpowiedni plik konfiguracyjny LCF.

Oprócz binarnego pliku z kodem wykonywalnym (rozszerzenie `*.xcp`), kompilator generuje także plik tekstowy (`*.dcp`) z kodem mnemonicznym, dodatkowymi informacjami dla symulatora (nazwy zmiennych) oraz raportem z kompilacji.

## 4.2. Klasy parsera i generatora kodu

Zasadnicze elementy kompilatora zostały zdefiniowane jako klasy języka C# [8, 9]. Każdy token wyodrębniony przez skaner stanowi obiekt odpowiedniej klasy dziedziczącej z abstrakcyjnej klasy bazowej `STIdentifier` (rysunek 9). Podczas kompilacji identyfikatory obiektów są gromadzone na listach. Tworzona jest więc lista identyfikatorów globalnych oraz lokalne listy dla funkcji, bloków funkcjonalnych i programów. Dla odnalezienia identyfikatorów na listach wykorzystywane są predykaty, co pozwoliło zrezygnować z pracochłonnego tworzenia tablic skrótów (tzw. *hash tables*).



Rysunek 9. Klasy obiektowej reprezentacji elementów języka ST

Identyfikatory na listach weryfikowane są pod względem unikalności. Pojawienie się dwu jednakowych identyfikatorów przerywa kompilację sygnalizując błąd. Przesłonięcie identyfikatora globalnego identyfikatorem lokalnym generuje komunikat w raporcie końcowym (dolne pole na rysunku 5).

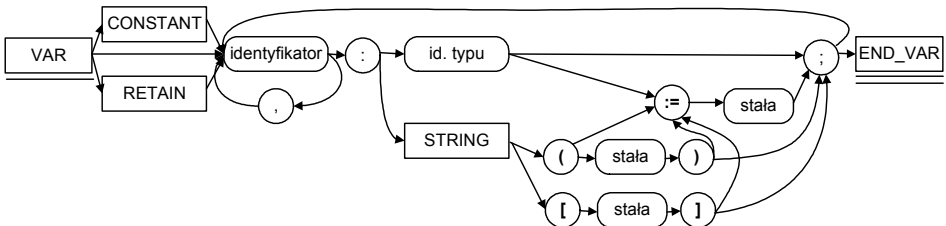
Instrukcje maszyny wirtualnej przekazywane z parsera do generatora kodu składają się z mnemoników i listy operandów. Każda instrukcja jest przechowywana w postaci instancji klasy `VMInstruction`. Lista operandów, jako obiekt klasy `VMOperand`, jest jednym z pól tej klasy. Wykorzystanie list zarówno przez ska-

ner, parser jak i generator kodu eliminuje problemy typowe dla tablic o stałych rozmiarach.

## 5. Kompilacja standardowego licznika CTU

### 5.1. Definicja bloku, deklaracje zmiennych

Etapy generowania kodu binarnego zostaną teraz szczegółowo omówione na przykładzie licznika CTU o kodzie źródłowym podanym w tabeli 2. Jak podano, w pierwszym etapie skaner dzieli znaki tego kodu na tokeny dokonując kategoryzacji. Pierwszym tokenem jest `FUNCTION_BLOCK` kwalifikowany do kategorii słów kluczowych. Następny token `CTU` zostanie więc potraktowany jako identyfikator nowego obiektu typu `STFunctionBlock` (rysunek 9) i zarejestrowany na liście identyfikatorów globalnych. Kolejnym tokenem jest słowo kluczowe `VAR_INPUT` rozpoczynające klauzulę deklaracyjną. Zgodnie ze składnią języka ST klauzule deklaracyjne muszą być zakończone przez `END_VAR`. Interpretacja klauzul zaczynających się od `VAR`, a więc `VAR_INPUT`, `VAR_OUTPUT` i `VAR` w CTU, odbywa się zgodnie z diagramem leksykalnym na rysunku 10. Odpowiednio do deklarowanych typów (tutaj `BOOL` i `INT`) następuje wyznaczenie rozmiarów oraz nadanie nazw obszarom pamięci przeznaczonym na zmienne wejściowe, wyjściowe i lokalne. Rozmiary te posłużą do określenia adresów w wynikowym kodzie binarnym (podsekcja 5.3).

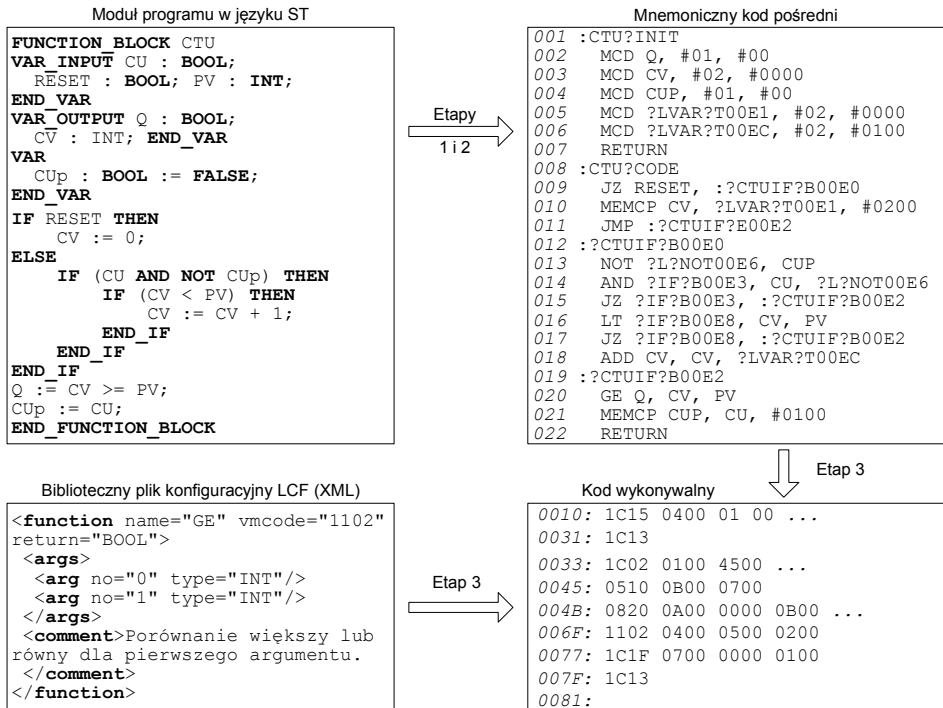


Rysunek 10. Diagram leksykalny klauzuli VAR

### 5.2. Kompilacja instrukcji, inicjacja instancji

Jeżeli kolejny token nie odpowiada żadnej klauzuli deklaracyjnej, rozpoczyna się kompilacja zasadniczego kodu reprezentującego działanie bloku. Kod licznika CTU powtórzono dla wygody w lewej górnej części rysunku 11. Najpierw generowany jest kod „ciała”, a następnie kod generujący instancję bloku, tzn. nadający wartości początkowe zmiennym wyjściowym i lokalnym, oraz inicjujący inne funkcje i bloki, jeżeli w tym bloku zostały użyte. Pierwszą instrukcją ciała jest tutaj instrukcja warunkowa rozpoczyna się słowem kluczowym `IF`. Powoduje to zinterpretowanie następnego identyfikatora `RESET` jako

wyrażenia logicznego i wygenerowanie instrukcji skoku JZ podanej w linii 009, w prawej górnej części rysunku 11 (listę instrukcji podano poprzednio w tabeli 5). Z linii tej następuje skok do fragmentu kodu oznaczonego etykietą :?CTUIF?B00E0, gdy RESET przyjmuje wartość FALSE (klauzula ELSE). Należy wyjaśnić, że nazwy zmiennych pomocniczych i etykiet tworzonych automatycznie podczas kompilacji instrukcji warunkowych, pętli oraz złożonych wyrażeń matematycznych zawierają pytajniki ?, co odróżnia je od nazw w ST (nie mogących zawierać pytajników).



Rysunek 11. Etapy kompilacji programu licznika CTU

Instrukcja MEMCP w linii 010 kopiuje dwa bajty (#0200) do zmiennej CV ze zmiennej pomocniczej ?LVAR?T00E1, która jest reprezentacją stałej 0 (zobacz niżej). JMP w linii 011 powoduje bezwarunkowy skok do etykiety :?CTUIF?E00E2 oznaczającej koniec kompilacji instrukcji IF. W liniach 013 i 014 znajdują się instrukcje wyznaczające wartość wyrażenia warunkowego CU AND NOT CUp z drugiej instrukcji IF. Jest ono obliczane z uwzględnieniem priorytetów działań i instrukcji języka ST zdefiniowanych w normie IEC. Linia 018 odpowiada wyrażeniu CV:=CV+1; z najbardziej zagnieżdżonego IF. Zmienna ?LVAR?T00EC reprezentuje stałą 1. Dwie instrukcje podstawień znajdujące się po ostatnim END\_IF zostały przetłumaczone do kodu mnemonicznego jako linie 020 i 021. Funkcja elementarna GE zastępuje operator >=. Kompilację ciała bloku kończy instrukcja RETURN.

Po zakończeniu przetwarzania instrukcji ciała kompilator przystępuje do tworzenia kodu inicjującego instancję bloku. Jest on pokazany w liniach 001 do 007 i zawiera pięć instrukcji MCD inicjujących zmienne wejściowe, lokalne i pomocnicze wartościami stałymi. Są to więc:

- zmienne boolowskie Q, CUp inicjowane wartością FALSE (#00),
- zmienna całkowita CV inicjowana wartością 0 (#0000),
- zmienne pomocnicze ?LVAR?T00E1 i ?LVAR?T00EC inicjowane odpowiednio wartościami 0 i 1 (#0100) wynikającymi z kodu.

Jak widać, wartości dwubajtowe są zapisywane w postaci *Little Endian*. Na zakończenie sekcji inicjującej jest generowane RETURN.

### 5.3. Kod binarny

W trzecim etapie powstaje wynikowy kod dla maszyny wirtualnej. W tym celu łączone są kody wszystkich modułów i następuje translacja kodu mnemonicznego do kodu binarnego. Zamieniane są zatem:

- mnemoniki instrukcji na odpowiadające im identyfikatory liczbowe,
- nazwy zmiennych na adresy w pamięci danych,
- nazwy etykiet na adresy kodu (bezwzględne lub relatywne).

Zamieniając mnemoniki korzysta się z identyfikatorów liczbowych zawartych w wspomnianym bibliotecznym pliku konfiguracyjnym LCF. Fragment tego pliku dotyczący instrukcji GE (linia 20 poprzednio) pokazano w lewej dolnej części rysunku 11. Atrybutem *vmcode* jest identyfikator liczbowy 1102, który pojawia się dalej w kodzie wykonywalnym pod adresem 006F (prawa dolna część rysunku 11).

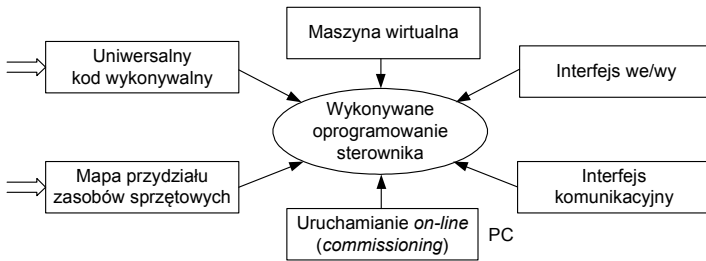
Określenie adresów lokalnych dla zmiennych następuje według kolejności deklaracji odpowiednio do wyznaczonych poprzednio rozmiarów (podsekcja 5.1), przy czym pierwszym adresem jest 0000. W przypadku licznika CTU deklarowane są kolejno CU, RESET, PV, Q, CV i CUp (rysunek 11). BOOL zajmuje jeden bajt, a INT dwa. Stąd adresami lokalnymi tych zmiennych są kolejno 0000, 0001, 0002, 0004, 0005 i 0007. Adresy zmiennych Q, CV i PV z instrukcji GE występują pod adresem 006F w kodzie binarnym (w postaci *Little Endian*).

Jak wspomniano, plik konfiguracyjny LCF jest tworzony indywidualnie dla każdej maszyny wirtualnej. W jednym przypadku mnemonikowi GE może odpowiadać identyfikator 1102, a w drugim inny. Wówczas instrukcję GE obydwie maszyny będą wykonywały odmiennie. Zróźnicowanie atrybutów *vmcode* jest drogą do *optymalizacji* kodu wykonywalnego dla konkretnego procesora. W pliku LCF za pomocą elementów *<deny-type>* (w sekcji TYPES) można również ograniczyć liczbę dostępnych typów danych. Zapewnia to sygnalizowaną na początku *elastyczność* środowiska CPDev, które pozwala tworzyć dedykowane kompilatory różniące się tylko plikami LCF. W ten sposób m.in. ograniczono do 10 liczbę typów danych w sterowniku SMC, o którym mowa dalej.

## 6. Realizacja maszyny wirtualnej

### 6.1. Maszyna wirtualna

Proces tworzenia oprogramowania wykonywanego na platformie docelowej ilustruje rysunek 12. Plik binarny z kodem uniwersalnym oraz mapa przydziału zasobów (drugi plik, podsekcja 7.1) są przesyłane do sterownika, gdzie przetwarza je maszyna wirtualna. Główne cechy tego przetwarzania scharakteryzowano poniżej.

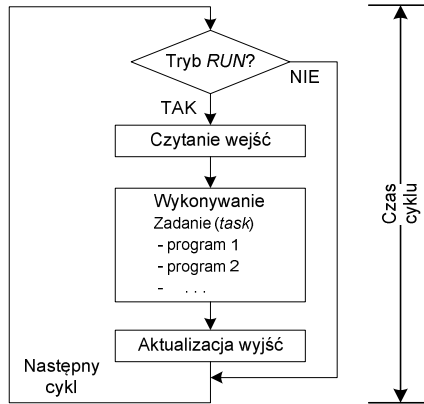


Rysunek 12. Tworzenie wykonywanego oprogramowania sterownika

Maszyna wirtualna jest automatem funkcjonującym według schematu z rysunku 13. W skład zadania wchodzi kolejno uruchamianie programy. Interpretowany kod binarny zawiera identyfikatory instrukcji oraz adresy operandów (rysunek 11, prawa dolna część). Maszyna wirtualna, podobnie jak typowy procesor, prowadzi licznik programu zawierający adres instrukcji do wykonania oraz aktualny adres bazowy obszaru z operandami (właściwy dla funkcji, bloku funkcjonalnego lub programu). Dysponując adresem instrukcji maszyna pobiera jej kod, dekoduje, pobiera operandy i wykonuje instrukcję. Emulacja stosu i aktualizacja adresów bazowych pozwala na wielokrotne, współbieżne wywoływanie funkcji, bloków i programów.

Organizację pamięci maszyny wirtualnej mieszczącej oprogramowanie utworzone w opisywany sposób pokazano na rysunku 14a. Instrukcje są ulokowane w segmencie kodu (pamięć typu *read only*). Segment danych zawiera zmienne globalne, lokalne i pomocnicze, z tego część z wartościami stałymi (porównaj podsekcję 5.2). Adresowanie tego segmentu może być bezpośrednie lub pośrednie za pomocą specjalnych rejestrów. Pamięć wewnętrzna maszyny zawiera kod interpretera oraz stosy i rejestry. Do pamięci tej nie można się w żaden sposób przedostać z poziomu programu.

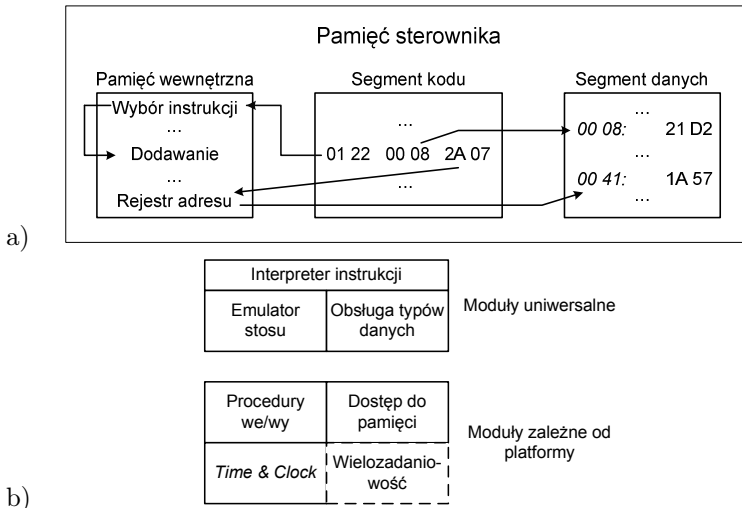
Jak pokazano na rysunku 14b, maszyna wirtualna ma strukturę modułową, aby ułatwić implementację na różnych platformach. Moduły uniwersalne pozostają niezmiennione (jeśli pominiąć to, że kod źródłowy musi być skompilowany na dany procesor). Moduły zależne od platformy należy przepisać lub zmodyfikować (ale bez zmiany sposobu wywołania, zobacz niżej). Właściwy dla danego sprzętu będzie np. moduł *Time & Clock* wykorzystujący przerwania zegarowe w operacjach z typem *TIME*. Indywidualnej obsługi wymaga także układ zegara czasu rzeczywistego.



Rysunek 13. Fazy cyklu maszyny wirtualnej

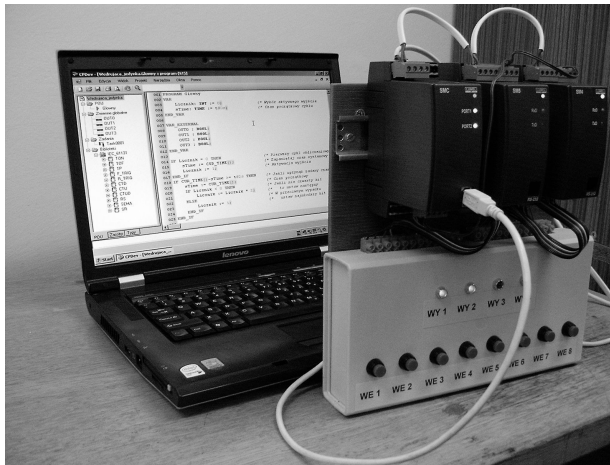
stego RTC warunkujący dostępność typu DATE\_AND\_TIME. Wielozadaniowość jest modulem opcjonalnym, ponieważ korzysta z mechanizmów konkretnego systemu operacyjnego. Instancje zadań tworzą wtedy własne kopie zmiennych globalnych celem uniknięcia konfliktów (tzw. *process image*).

Część uniwersalną maszyny wirtualnej napisano w standardowym języku ANSI C, aby móc ją wprost wykorzystać w różnych procesorach. Jak podano wcześniej, liczbę typów danych oraz sposób wykonywania instrukcji elementarnych w konkretnym procesorze określa plik konfiguracyjny LCF. Dla obsługi układów procesora (przerwania, RTC) oraz interfejsów zewnętrznych zostały opracowane specyfikacje ogólne w formie prototypów odpowiednich procedur (nazwy, typy



Rysunek 14. Maszyna wirtualna: (a) organizacja pamięci, (b) struktura modułowa





Rysunek 15. Testowy minisystem rozproszony ze sterownikiem SMC i modułami SM z LUMELu Zielona Góra

wejść i zwracanych wyjść). Specyfikacje te nie zależą od procesora i rozwiązań układowych. Plik z prototypami tych procedur (\*.h) jest kompilowany wraz z uniwersalnymi modułami maszyny wirtualnej. Treści tych procedur mogą być opracowane niezależnie i w formie pliku binarnego (np. \*.obj) skonsolidowane ze skompilowanym kodem dając całość oprogramowania maszyny wirtualnej dla konkretnej platformy. Treści te zależą oczywiście od procesora i rozwiązań układowych, ale mogą je opracować sami konstruktorzy danego sterownika. Przez powierzenie programowania interfejsów konstruktorom urządzeń, pakiet CPDev staje się *otwarty* również pod względem sprzętowym.

## 6.2. Minisystem rozproszony ze sterownikiem SMC

Pierwszym urządzeniem, do programowania którego wykorzystuje się pakiet CPDev jest sterownik SMC wprowadzony ostatnio przez Zakłady LUMEL w Zielonej Górze [10]. SMC ma dwa porty szeregowo RS-485, RS-232/485 oraz interfejs USB, ale brak w nim własnych wejść i wyjść obiektowych. Z tego względu pełni rolę modułu centralnego w minisystemie rozproszonym przeznaczonym do sterowania, regulacji i monitorowania niewielkich procesów technologicznych. Komunikacja jest prowadzona według protokołu Modbus RTU [11].

Na rysunku 15 pokazano testową realizację minisystemu ze sterownikiem SMC (górny po lewej), modułem wejść binarnych SM5 (w środku) oraz modułem wyjść binarnych SM4 (po prawej). Widoczny poniżej pulpit z przyciskami i LEDami służy do testowania prostych programów logicznych i sekwencyjnych. Komputer PC korzysta z pakietu CPDev (programowanie) i pakietu SCADA (wizualizacja). PC i SMC są połączone przez port USB traktowany jako wirtualny port szeregowy.

Śśród 20 typów danych dostępnych w pakiecie CPDev do implementacji w SMC wybrano 10, tj. BOOL, INT, DINT, UINT, REAL, WORD, DWORD, STRING, TIME

oraz DATE\_AND\_TIME. Konfigurator zasobów sprzętowych CPCCon, będący trzecim składnikiem pakietu CPDev, służy tutaj do konfiguracji komunikacji. Jest on opisany w następnym punkcie.

Jednostką centralną sterownika SMC jest 8-bitowy mikrokontroler ATmega 128 z rodziny Atmel AVR [12]. Jest to nowoczesny procesor typu RISC wykonujący większość instrukcji w jednym cyklu zegara, z osobnymi magistralami dla pamięci programu i danych. Część uniwersalną maszyny wirtualnej sterownika, tzn. interpreter instrukcji, emulator stosu i obsługę typów danych (rysunek 14b), skompilowano kompilatorem *avr-gcc* z pakietu WinAVR [13]. Część zależną od sprzętu, tzn. przerwania, zegar RTC i interfejsy komunikacyjne oprogramowali konstruktorzy z LUMELu i przesłali w formie pliku binarnego. Po konsolidacji powstała wynikowa maszyna wirtualna, która umieszczona w pamięci *flash* sterownika wykonuje programy napisane w języku ST.

## 7. Konfigurator i symulator

### 7.1. Konfigurator komunikacji CPCCon

W strukturze funkcjonalnej pakietu CPDev konfigurator CPCCon występuje w warstwie sprzętowej, ponieważ tworzy mapę przydziału zasobów (rysunek 2). W przypadku minisystemu rozproszonego ze sterownikiem SMC, CPCCon służy do konfiguracji komunikacji. Kompilator CPDev przekazuje mu plik binarny z kodem wykonywalnym (\*.xcp) oraz plik tekstowy z kodem mnemonicznym i nazwami zmiennych (\*.dcp).

Funkcjami konfiguratora CPCCon w odniesieniu do systemu testowego z rysunku 15 są:

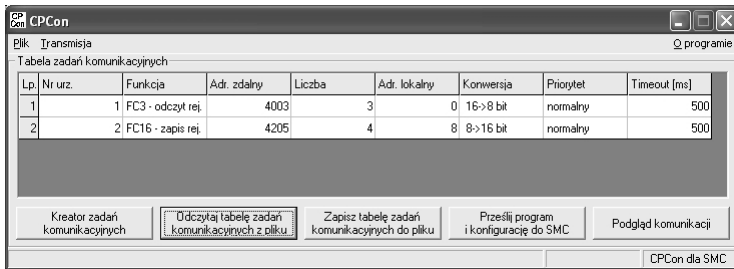
- konfiguracja komunikacji SMC-moduły SM,
- utworzenie pliku z mapą konfiguracji sprzętu (\*.xmc),
- przesłanie plików z kodem wykonywalnym (\*.xcp) i mapą (\*.xmc) do sterownika SMC.

Zazwyczaj po utworzeniu mapy, a przed przesłaniem plików, przeprowadzana jest jeszcze symulacja systemu za pomocą symulatora CPSim (zobacz niżej).

Główne okno konfiguratora CPCCon pokazano na rysunku 16. W zakładce **Transmisja** ustawia się parametry transmisji PC $\leftrightarrow$ SMC (prędkość, parzystość, bity stopu) oraz SMC $\leftrightarrow$ SM. Takie same parametry muszą być oczywiście wcześniej ustawione w porcie USB sterownika oraz portach szeregowych modułów SM.

Treścią okna CPCCon jest przede wszystkim **Tabela zadań komunikacyjnych** określająca, jakie transakcje typu pytanie $\leftrightarrow$ odpowiedź i polecenie $\leftrightarrow$ potwierdzenie przeprowadza sterownik SMC (*master*) z modułami wejścia/wyjścia (*slaves*). Transakcje są tutaj nazywane *zadaniami komunikacyjnymi*, a reprezentują je kolejne wiersze tabeli. Konfiguracja systemu rozproszonego sprowadza się do wypełnienia odpowiedniej liczby wierszy dokonywanego albo bezpośrednio, albo za pomocą kilku okien **Kreatora zadań komunikacyjnych** (przycisk aktywujący).

Pierwszy wiersz tabeli na rysunku 16 reprezentuje komunikację sterownika SMC z modulem wejść binarnych SM5 obsługującym przyciski pulpitu (rysunek 15), które w przypadku projektu WEDRUJACA\_JEDYNKA (z podsekcji 3.2) ustawiają zmienne START, PAUZA i RESET. Zmienne te po stronie SMC mają kolejne adresy począwszy od 0000 (rysunek 6). Moduł SM5 umieszcza wejścia w kolejnych 16-bitowych rejestrach od adresu 4003. Dla sterownika SMC są to adresy w urządzeniu *zdalnym*. Ponieważ adresy zdalne są adresami kolejnymi, więc trzy wejścia modułu można odczytać jedną transakcją Modbusa z kodem FC3 (odczyt grupy rejestrów [11]). Koniecznie należy jednak wskazać, że w 16-bitowych rejestrach mieszczą się wartości binarne, a nie np. liczby całkowite. Ponieważ typ **BOOL** zajmuje w pakiecie CPDev jeden bajt, więc interfejs maszyny wirtualnej w SMC powinien najpierw dokonać konwersji 16→8 bit.



Rysunek 16. Konfiguracja komunikacji projektu WEDRUJACA\_JEDYNKA

Zadania komunikacyjne (transakcje) obsługiwane są w przerwach, jakie po wykonaniu właściwego programu pozostają do końca cyklu. W systemie z rysunku 15 jedna transakcja zajmuje zazwyczaj 20 do 50 ms zależnie od prędkości (max. 115.2 kbit/s). Jeżeli przerwa jest znaczna, zadanie może być wykonane kilkakrotnie. Na razie dla prostoty przyjęto, że zadanie o priorytecie *normalnym* ma być wykonywane dwukrotnie rzadziej niż zadanie o priorytecie *wysokim*, a zadanie o priorytecie *niskim* trzykrotnie rzadziej. Jak widać z rysunku 16, komunikacja z modulem SM5 ma normalny priorytet, a czas *timeout*, w ciągu którego transakcja powinna się zakończyć wynosi 500 ms.

Podobnie wygląda drugi wiersz tabeli zadań definiujący komunikację SMC z modulem wyjść binarnych SM4 sterującym LEDami. Cztery kolejne zmienne OUT0 do OUT3, pierwsza o adresie lokalnym 0008, przesyłane są do SM4 komunikatem o kodzie FC16 (zapis grupy rejestrów) na adresy zdalne począwszy od 4205. Tym razem potrzebna jest konwersja 8→16 bit.

## 7.2. Symulator CPSim

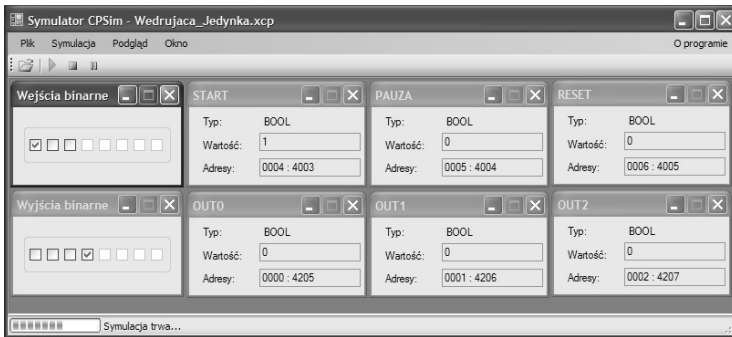
Celem symulacji jest sprawdzenie funkcjonowania skompilowanego projektu przed zapisaniem w sterowniku, aby wykryć i usunąć ewentualne usterki. Symulację można przeprowadzić w dwu trybach:

- przed konfiguracją zasobów sprzętowych (symulacja algorytmu),

— po konfiguracji zasobów (symulacja całego systemu).

Pierwszy tryb dotyczy wyłącznie warstwy logicznej środowiska CPDev według podziału z rysunku 2. Symulator CPSim funkcjonuje wtedy w PC jako maszyna wirtualna korzystająca z uniwersalnego kodu wykonywalnego (pliki \*.xcp i \*.dcp). Weryfikuje się w ten sposób poprawność samego algorytmu. Tak sprawdzony kod może być przenoszony na różne platformy (porównaj rysunek 3).

W drugim trybie symuluje się projekt po skonfigurowaniu zasobów sprzętowych, które w przypadku minisystemu przedstawionego powyżej przeprowadza konfigurator komunikacji CPCon tworzący mapę przydziału zasobów (\*.xmc). Symulator korzysta teraz zarówno z kodu jak i mapy, aby poziom symulacji dostosować jak najbardziej do sprzętu. Fragment okna CPSim podczas symulacji projektu WEDRUJACA\_JEDYNKA pokazano na rysunku 17. Dwa grupowe pola po lewej stronie przedstawiają kolorami wartości trzech zmiennych wejściowych i czterech wyjściowych. Adresy w polach indywidualnych są odpowiednio adresami lokalnym i zdalnym. Wybór pól w oknie (grupowe, indywidualne) oraz przypisanie im zmiennych należy do użytkownika. Wartości zmiennych można ustawiać w obydwu typach pól.



Rysunek 17. Symulacja projektu WEDRUJACA\_JEDYNKA

Na razie okno z rysunku 17 służy tylko do symulacji. Trwają jednak prace, aby wykorzystywać je również do uruchamiania oprogramowania wprost w sterowniku w trybie *on-line* (tzw. *commissioning*, rysunek 12).

## 8. Podsumowanie

Przedstawiono stan prac nad prototypowym środowiskiem CPDev przeznaczonym do programowania sterowników i innych urządzeń kontrolno-pomiarowych według normy IEC 61131-3. Na razie programować można tylko w języku ST. Środowisko CPDev jest uniwersalne, tzn. skompilowany kod można wykonywać na różnych platformach sprzętowych. W wykonywaniu tego kodu muszą jednak pośredniczyć maszyny wirtualne właściwe dla zastosowanych procesorów. Odpowiada to koncepcji maszyn wirtualnych Javy. Obsługę interfejsów zewnętrznych

konfiguruje się odpowiednio do wykorzystywanego sprzętu. CPDev jest środowiskiem otwartym umożliwiając tworzenie własnych bloków funkcjonalnych i bibliotek oraz programowanie procedur obsługi interfejsów przez konstruktorów urządzeń. Jest także środowiskiem elastycznym, pozwalając dobierać typy danych odpowiednio dla konkretnej aplikacji. W miarę szczegółowo omówiono proces kompilacji prowadzony przez skaner, parser i generator kodu binarnego oraz realizację maszyny wirtualnej. Przedstawiono także pierwszą aplikację CPDev, tzn. pakiet do programowania minisystemów rozproszonych z urządzeniami z zielonogórskiego LUMELu.

Zakłada się, że do ST będą w przyszłości konwertowane programy napisane w innych językach normy IEC (przede wszystkim FBD). Format wymiany danych między językami został już zdefiniowany przez międzynarodowe stowarzyszenie PLCopen [14].

## Bibliografia

- [1] PN-EN 61131-3 – *Sterowniki programowalne. Część 3: Języki programowania*. Warszawa, 2004.
- [2] Kasprzyk J. *Programowanie sterowników przemysłowych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2006.
- [3] Rzońca D., Sadolewski J., Trybus B. *Kompilator języka ST normy IEC 61131-3 na uniwersalny kod wykonywalny*. W: *Systemy Czasu Rzeczywistego* (red. Z. Huzar, Z. Mazur). Wydawnictwa Komunikacji i Łączności, Warszawa, 2007, s. 189–198.
- [4] Stec A., Świder Z., Trybus L. *Charakterystyka funkcjonalna prototypowego systemu do programowania systemów wbudowanych według normy IEC 61131-3*. W: *Systemy Czasu Rzeczywistego* (red. Z. Huzar, Z. Mazur). Wydawnictwa Komunikacji i Łączności, Warszawa, 2007, s. 179–188.
- [5] Microsoft *.NET Framework Developer's Guide* [online]  
<http://msdn2.microsoft.com/en-US/library/aa720433.aspx>.
- [6] Lindholm T., Yellim F. *Java Virtual Machine Specification*. Second ed., Java Software, Sun Microsystems Inc., 2004.
- [7] Cooper K., Torczon L. *Engineering a Compiler*. Morgan Kaufmann, San Francisco, 2003.
- [8] *C# Language Specification* [online]  
<http://msdn2.microsoft.com/en-us/vcsharp/aa336809.aspx>.
- [9] Appel A., Palsberg J. *Modern compiler implementation in Java*. Second ed. Cambridge University Press, 2002.
- [10] Lubuskie Zakłady Aparatów Elektrycznych LUMEL S.A., Zielona Góra – <http://www.lumel.com.pl>
- [11] *Modicon MODBUS Protocol Reference Guide*. Modicon Inc., Massachusetts 1996 [online] [http://www.modbus.org/docs/PI\\_MBUS\\_300.pdf](http://www.modbus.org/docs/PI_MBUS_300.pdf).
- [12] *ATmega 128 Datasheet*. Atmel 2007 [online]  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc2467.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf)
- [13] *WinAVR Development Tools for the Atmel AVR* [online]  
<http://winavr.sourceforge.net>
- [14] *XML Formats for IEC 61131-3 ver. 1.0 – Official Release*. [online]  
<http://www.plcopen.org>





