

# **Studia i Materiały Informatyki Stosowanej**



# **Studia i Materiały Informatyki Stosowanej**

Czasopismo młodych pracowników  
naukowych, doktorantów i studentów

Tom 11, Nr 1, 2019

Bydgoszcz 2019

**Studia i Materiały Informatyki Stosowanej**  
**Czasopismo młodych pracowników naukowych, doktorantów**  
**i studentów**

**© Copyright 2019 by Uniwersytet Kazimierza Wielkiego**

**Patronat naukowy:**

Instytut Informatyki  
Uniwersytet Kazimierza Wielkiego  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
tel. (052) 325 76 11  
e-mail: [simis@ukw.edu.pl](mailto:simis@ukw.edu.pl)

**ISSN 1689-6300**

**Projekt okładki: Łukasz Zawadzki (StudioStrzelec.pl)**  
**DTP: Dawid Ewald**

**Wydawca:**

Instytut Informatyki  
Uniwersytet Kazimierza Wielkiego  
Dyrektor:  
dr hab. inż. Izabela Rojek, prof. uczelni  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
tel. +48 52 325 76 11  
*email: [izarojek@ukw.edu.pl](mailto:izarojek@ukw.edu.pl)*

**Kontakt:**

dr inż. Jacek Czerniak, prof. uczelni  
dr hab. inż. Marek Macko, prof. uczelni  
Uniwersytet Kazimierza Wielkiego  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
*e-mail: [jczerniak@ukw.edu.pl](mailto:jczerniak@ukw.edu.pl)*  
*[mackomar@ukw.edu.pl](mailto:mackomar@ukw.edu.pl)*

**Druk (ze środków sponsora):**  
**Oficina Wydawnicza MW**

**Nakład 250 egz.**

**Bydgoszcz 2019**

# **Studies and Materials in Applied Computer Science**

Journal of young researchers,  
PhD students and students

Vol. 11, No.1, 2019

Bydgoszcz 2019

**Studies and Materials in Applied Computer Science**  
**Journal of young researchers, PhD students and students**

© Copyright 2019 by Kazimierz Wielki University in Bydgoszcz

**Scientific patronage:**

Institute of Informatics  
Kazimierz Wielki University  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
tel. +48 52 325 76 11  
*e-mail: simis@ukw.edu.pl*

**ISSN 1689-6300**

**Cover designed by: Łukasz Zawadzki (StudioStrzelec.pl)**

**DTP by: Dawid Ewald**

**Publisher:**

Institute of Informatics  
Kazimierz Wielki University  
Head:  
Izabela Rojek, PhD DSc Eng Assoc. Prof.  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
tel. + 48 52 325 76 11  
*e-mail: izarojek@ukw.edu.pl*

**Contact:**

Jacek Czerniak, PhD. Eng. Assoc. Prof.  
Marek Macko, PhD. DSc. Eng., Assoc Prof  
Kazimierz Wielki University  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
*e-mail: jczerniak@ukw.edu.pl*  
*mackomar@ukw.edu.pl*

**Printing (funded from non-profit programme):**  
**Oficyna Wydawnicza MW**

**Edition of 250 copies**

**Bydgoszcz 2019**

# Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników naukowych, doktorantów i studentów

patronat: Polskie Towarzystwo Informatyczne



## Przewodniczący Rady Naukowej

prof. dr hab. inż. czł. rzec. PAN Janusz Aleksander Kacprzyk, IBS PAN

## Redaktorzy Naczelni

dr inż. Jacek Czerniak, UKW

dr hab. inż. Marek Macko, prof. nadzw.

## Redaktor Zarządzający

dr inż. Łukasz Apiecionek, UKW

## Redaktor Statystyczny

dr Iwona Filipowicz, UKW

## Komitet Redakcyjny

dr inż. Mariusz Dramski, AM

dr inż. Hubert Zarzycki, WWSIS

dr inż. Marcin Łukasiewicz, UTP

dr inż. Piotr Dziurzański, ZUT

dr inż. Tomasz Kałaczyński, UTP

dr hab. inż. Grzegorz Domek, prof. nadzw.

dr Piotr Prokopowicz, UKW

## Redaktor Tematyczny (Informatyka)

prof. dr inż. Rafał A. Angryk, GSU

## Redaktor Tematyczny (Mechatronika)

prof. dr.h.c.mult. Peter Kopacek, VUT

## Redaktor Tematyczny (Metody numeryczne)

dr hab. Petro Filevych, LNUVB

## Redaktor Językowy (j.polski)

dr Małgorzata Kempieńska, FRM

## Redaktor Językowy (j.angielski)

Andrew Gill, Reed Elsevier, UK

## Rada Naukowa

|                                            |                  |                                                                     |
|--------------------------------------------|------------------|---------------------------------------------------------------------|
| dr hab. Stanisław                          | Ambroszkiewicz   | Instytut Podstaw Informatyki PAN                                    |
| prof. dr inż. Rafał A.                     | Angryk           | Georgia State University, USA                                       |
| dr hab. Zenon                              | Biniek           | Wyższa Szkoła Technologii Informatycznych                           |
| prof. dr hab. inż. Ryszard                 | Budziński        | Zachodniopomorski Uniwersytet Technologiczny                        |
| dr inż. Joanna                             | Chimiak-Opoka    | University of Innsbruck, Austria                                    |
| prof. dr hab. inż. Ryszard                 | Choraś           | Uniwersytet Technologiczno-Przyrodniczy                             |
| dr hab. Petro                              | Filevych         | Lviv National University of Veterinary and Biotechnologies, Ukraina |
| prof. dr hab. inż. Piotr                   | Gajewski         | Wojskowa Akademia Techniczna                                        |
| dr inż. Marek                              | Holyński         | Prezes Polskiego Towarzystwa Informatycznego                        |
| prof. dr hab. inż. czł. rzec. PAN Janusz   | Kacprzyk         | Instytut Badań Systemowych PAN                                      |
| dr hab. inż. Andrzej                       | Kobyliński       | Szkoła Główna Handlowa                                              |
| prof. dr hab. inż. czł. koresp. PAN Józef  | Kopacek          | Vienna University of Technology, Austria                            |
| prof. dr hab. inż. Jacek                   | Korbicz          | Uniwersytet Zielonogórski                                           |
| prof. dr hab. inż. Marek                   | Koronacki        | Instytut Podstaw Informatyki PAN                                    |
| prof. dr hab. inż. Halina                  | Kurzyński        | Politechnika Wrocławska                                             |
| prof. dr Mirosław                          | Kwaśnicka        | Politechnika Wrocławska                                             |
| dr inż. Dariusz                            | Majewski         | New York Institute of Technology, United Arab Emirates              |
| prof. dr hab. Andrzej                      | Mikolajewski     | Uniwersytet Kazimierza Wielkiego                                    |
| dr hab. Marcin                             | Marciniak        | Politechnika Poznańska                                              |
| prof. dr hab. inż. czł. PAN Witold         | Paprzycki        | Instytut Badań Systemowych PAN                                      |
| prof. dr hab. inż. Andrzej                 | Pedrycz          | University of Alberta, Canada                                       |
| prof. dr hab. inż. Andrzej                 | Pieगत            | Zachodniopomorski Uniwersytet Technologiczny                        |
| prof. dr hab. inż. Orest                   | Polański         | Politechnika Śląska                                                 |
| prof. dr inż. George                       | Popov            | Zachodniopomorski Uniwersytet Technologiczny                        |
| dr hab. inż. prof. nadzw. Izabela          | Przybył Einstein | College of Medicine, USAT Montserrat                                |
| prof. dr hab. inż. Danuta                  | Rojek            | Uniwersytet Kazimierza Wielkiego                                    |
| prof. dr hab. inż. czł. koresp. PAN Leszek | Rutkowska        | Politechnika Częstochowska                                          |
| prof. dr hab. inż. Milan                   | Rutkowski        | Politechnika Częstochowska                                          |
| prof. dr hab. inż. czł. rzec. PAN Roman    | Sága             | Žilinská Univerzita, Słowacja                                       |
| prof. dr hab. inż. Włodzimierz             | Słowiński        | Instytut Badań Systemowych PAN, Politechnika Poznańska              |
| prof. dr hab. inż. Andrzej                 | Sosnowski        | Uniwersytet Kazimierza Wielkiego, IPPT PAN                          |
| dr hab. inż. Jan                           | Stateczny        | Akademia Morska w Szczecinie                                        |
| prof. dr hab. Tomasz                       | Studziński       | Instytut Badań Systemowych PAN                                      |
| prof. dr hab. Janusz                       | Szapiro          | Szkoła Główna Handlowa                                              |
| prof. dr hab. inż. czł. rzec. PAN Ryszard  | Szczepański      | Uniwersytet Kazimierza Wielkiego, IPPT PAN                          |
| prof. zw. dr hab. inż. czł. rzec. PAN Jan  | Tadeusiewicz     | Akademia Górniczo-Hutnicza                                          |
| prof. dr hab. inż. Sławomir                | Węglarz          | Instytut Chemii Bioorganicznej PAN, Politechnika Poznańska          |
| prof. dr hab. inż. Antoni                  | Wierchoń         | Instytut Podstaw Informatyki PAN                                    |
| dr hab. inż. Andrzej                       | Wiliński         | Zachodniopomorski Uniwersytet Technologiczny                        |
| dr hab. inż. Ryszard                       | Wiśniewski       | Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach                  |
| prof. dr hab. Sławomir                     | Wojtyna          | Uniwersytet Technologiczno-Przyrodniczy                             |
| prof. dr inż. Milan                        | Zadrozny         | Instytut Badań Systemowych PAN                                      |
| prof. dr hab. Zenon                        | Żmindać          | Žilinská Univerzita, Słowacja                                       |
|                                            | Zwierzewicz      | Akademia Morska w Szczecinie                                        |



## SPIS TREŚCI

|                                                                                            |          |
|--------------------------------------------------------------------------------------------|----------|
| <b>Od Redakcji .....</b>                                                                   | <b>4</b> |
| <b>Hurtownia danych w planowaniu produkcji</b>                                             |          |
| Izabela Rojek, Zbyszko Królikowski, Mariola Marciniak .....                                | 5        |
| <b>Zapis logiczny układu kombinacyjnego metodą reprezentacji binarnych</b>                 |          |
| Janusz Łukowski .....                                                                      | 11       |
| <b>Wybrane metody sztucznej inteligencji zaimplementowane w języku programowania C/C++</b> |          |
| Aleksandra Mreła .....                                                                     | 14       |
| <b>Implementacja środowiska testowego w języku R</b>                                       |          |
| Jan Baumgart .....                                                                         | 19       |



## OD REDAKCJI

Szanowni Czytelnicy,

*Na progu Nowego 2019 roku przedstawiamy Czytelnikom kolejny numer naszego czasopisma poświęcony szerokiemu przekrojowi zagadnień z obszaru informatyki: od reprezentacji binarnych, poprzez zagadnienia programistyczne w językach C/C++ i R aż do przemysłowego wykorzystania hurtowni danych. Pozwala to spojrzeć na zagadnienia informatyczne jako rozwiązywanie szerokiego spektrum problemów: od czysto teoretycznych aż po złożone kwestie praktyczne generowane przez otoczenie społeczno-gospodarcze uczelni. Potencjał rozwojowy informatyki jest ogromny, a jej zastosowania mogą zarówno dawać satysfakcje naukową, jak i przyczyniać się do poprawy kondycji gospodarczej, a szerzej - jakości życia ludzi.*

Redaktorzy Naczelni SiMIS,  
dr inż. Jacek Czerniak,  
dr hab. inż. Marek Macko, prof. nadzw.

# HURTOWNIA DANYCH W PLANOWANIU PRODUKCJI

Izabela Rojek<sup>\*1</sup>, Zbyszko Królikowski<sup>2</sup>, Mariola Marciniak<sup>3</sup>

<sup>1</sup> Uniwersytet Kazimierza Wielkiego, Instytut Mechaniki i Informatyki Stosowanej, Chodkiewicza 30, 85-064 Bydgoszcz

<sup>2</sup> Politechnika Poznańska, Wydział Informatyki, Pl. Marii Skłodowskiej-Curie 5, 60-965 Poznań

<sup>3</sup> Uniwersytet Kazimierza Wielkiego, Instytut Matematyki, Chodkiewicza 30, 85-064 Bydgoszcz  
e-mail: izarojek@ukw.edu.pl

**Streszczenie:** W artykule scharakteryzowano hurtownie danych w planowaniu produkcji. W szczególności omówiono cechy charakterystyczne planowania produkcji oraz przepływ informacji i decyzji w tym planowaniu. Zdefiniowano hurtownie danych oraz przedstawiono architektury systemów hurtowni danych. Jako przypadek szczególny omówiono możliwość zastosowania hurtowni danych w planowaniu produkcji w wybranym przedsiębiorstwie produkcyjnym. Artykuł ten jest pierwszym z cyklu, wprowadzającym do wybranej tematyki.

**Słowa kluczowe:** Hurtownia danych, przedsiębiorstwo, planowanie produkcji

## Data warehouse in production planning

**Abstract:** The article describes data warehouses in production planning. In particular, the characteristics of production planning and the flow of information and decisions in this planning were discussed. Data warehouses were defined and the architecture of data warehouse systems was presented. As a special case, the possibility of using a data warehouse in production planning in a selected production company was discussed. This article is the first in the introductory cycle to the chosen topic.

**Key words:** Data warehouse, enterprise, production planning

### 1. Wprowadzenie

Dane, informacje i wiedza stanowią zasób niematerialny każdego przedsiębiorstwa, którego użyteczność zależy od skuteczności ich przetworzenia i przekazania użytkownikom w formie najlepiej zaspokajającej ich wymagania. Tworzą one zasoby informacyjne przedsiębiorstwa, które funkcjonują w jego systemach informacyjnych w postaci baz danych, hurtowni danych, jak też w bardziej zaawansowanych systemach informacyjnych jako bazy wiedzy. Zasoby danych utrzymywane są w systemach informacyjnych przedsiębiorstwa dzięki odpowiedniemu oprogramowaniu i za jego pośrednictwem są udostępniane użytkownikom. Zasoby te potrzebne do podejmowania decyzji strategicznych przeważnie nie wymagają uaktualnienia z dokładnością do minuty, ale powinny zawierać historię przedmiotu. Wobec tego hurtownia danych wydaje się bardzo dobrym rozwiązaniem, które można zastosować w systemach wspomagania decyzji strategicznych, ponieważ jest ona zasilana, według określonego planu, danymi pochodzącymi z innych systemów zewnętrznych. Informacje w systemie są zatem stałe i przeważnie nie pozwalają użytkownikom na ich uaktualnianie i kasowanie. Dzięki takiemu rozwiązaniu dane przechowywane w systemie nie są

najświeższe, ale są aktualne na dany moment czasu, nawet wtedy gdy w bazie transakcyjnej uległy one zmianie [1].

Artykuł przedstawia pierwszy etap badań autorów nad projektowaniem hurtowni danych w planowaniu produkcji.

### 2. Planowanie produkcji

W każdym przedsiębiorstwie produkcyjnym centralnym punktem są zasoby informacyjne dotyczące produkcji, jej planowania, sterowania i rozliczania. Centralnym procesem każdego systemu zarządzania produkcją jest planowanie, którego zadaniem staje się wygenerowanie właściwego planu oraz rozłożenie go w czasie [2]. Planowanie produkcji to złożony proces. Obejmuje zarówno tworzenie planów strategicznych, taktycznych jak i operacyjnych. Jego nieodłączną częścią jest podejmowanie decyzji w oparciu o posiadane informacje. Plan strategiczny obejmuje misję oraz cele strategiczne przedsiębiorstwa, analizę otoczenia, strategię ogólną i strategię funkcjonalną. W przedsiębiorstwie produkcyjnym planowanie strategiczne obejmuje następujące obszary: strategię produktu, zagregowany plan produkcji, strategię rozwoju zdolności produkcyjnych i inwestycji, strategię finansową oraz strategię pracy i zatrudnienia.

Podstawą do tworzenia planów strategicznych są badania marketingowe oraz prognozy (ekonomiczne, techniczne, zapotrzebowania i sprzedaży). Prognozy indywidualne dla każdego wyrobu przedsiębiorstwa są w praktyce obciążone znacznymi błędami. Dokładniejsze prognozy można opracować dla grupy wyrobów. Metoda mająca doprowadzić do osiągnięcia celów strategicznych nazywana jest taktyką, stąd też po planowaniu strategicznym następuje planowanie taktyczne. Podstawowym zadaniem planowania taktycznego jest obliczanie planów produkcji poszczególnych wyrobów na podstawie planów produkcji łącznej oraz ich podziału w czasie (np. rozbić planów miesięcznych na tygodniowe). Planowanie taktyczne obejmuje w przedsiębiorstwie następujące działy: badanie, rozwój i przygotowanie produkcji, obsługę serwisową, finanse oraz dystrybucję. Na poziomie planowania taktycznego tworzone są plany: taktycznego zapotrzebowania materiałowego (MRP I), taktycznego zapotrzebowania na zasoby produkcyjne (MRP II), sterowania jakością produktów, utrzymania ruchu i niezawodności urządzeń oraz sieci i instalacji produkcyjnych. Planowanie taktyczne stanowi wytyczne dla planów operacyjnych. Planowanie operacyjne jest to krótkookresowy plan produkcji, którego podstawą jest rozdział zadań wytwórczych oraz gospodarka zapasami. Rozdział zadań produkcyjnych sprowadza się do wyznaczenia konkretnych operacji, terminów ich realizacji oraz wykonawców. Planowanie operacyjne polega na: przydziale zleceń produkcyjnych, dostarczaniu wyposażenia narzędziowego, materiałach i instrukcji wykonywanych, zapewnieniu obsługi poszczególnych stanowisk pracy, ustaleniu kolejności wykonywanych zadań, określeniu terminu rozpoczęcia i zakończenia zadań, badaniu jakości wykonania oraz aktualizacji stanu zaawansowania prac.

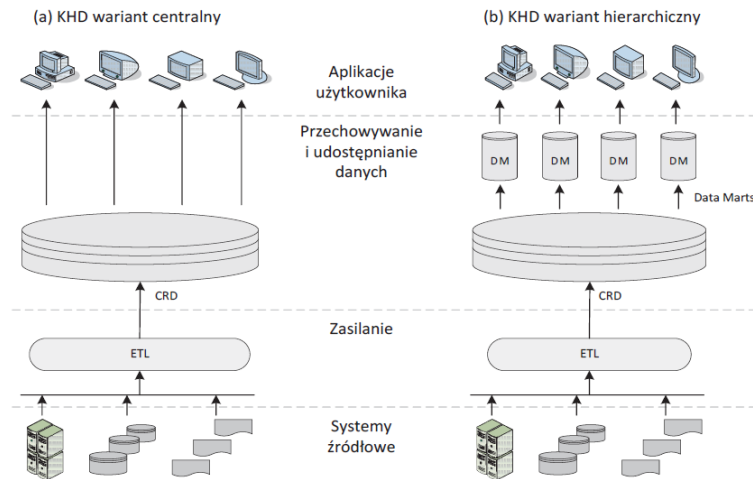


**Rysunek 1.** Przepływ informacji i decyzji w procesie planowania [3].

### 3. Hurtownia danych

W odpowiedzi na zapotrzebowanie dużych korporacji w zakresie przetwarzania danych na potrzeby biznesowe powstała koncepcja hurtowni danych (DW - *data warehouse*) – struktury pozwalającej decydentom na podejmowanie optymalnych decyzji na podstawie danych z różnych źródeł. Według definicji Williama Inmona istotą hurtowni danych jest „skoncentrowane na temat, zintegrowane, zmienne w czasie, nieulotne gromadzenie danych wykorzystywane do podejmowania decyzji strategicznych” [4,5,6,7]. Hurtownia danych jest wyższym szczeblem abstrakcji niż zwykła relacyjna baza danych (choć do jej tworzenia używane są także podobne technologie). Dane przechowywane w hurtowni danych przeznaczone są raczej do odczytu, zagregowane na wielu poziomach szczegółowości i zorientowane tematycznie (np. hurtownia danych klientów). W praktyce hurtownie są bazami danych integrującymi dane ze wszystkich pozostałych systemów bazodanowych w przedsiębiorstwie. Dostęp do tych danych może być problematyczny gdyż dane takie często są rozproszone i mają heterogeniczny charakter. Ponadto przetwarzanie danych w systemach bazodanowych przedsiębiorstwa wspomagające bieżące funkcjonowanie firmy odbywa się zwykle w trybie transakcyjnym (OLTP - *On-line Transaction Processing*) podczas gdy proces wspomagania decyzji (DSS - *Decision Support System*) wymaga podejścia analitycznego (OLAP- *OnLine Analytical Processing*). Hurtownia danych musi zatem obok danych bieżących także dane historyczne dzięki czemu możliwa jest analiza zmienności w czasie, wyszukiwanie trendów i formułowanie prognoz. Oznacza to, że w praktyce hurtownia danych jest bardzo dużą bazą danych. Architektura hurtowni danych może mieć różny charakter. Początkowo dominowała architektura systemu oparta na centralnym repozytorium danych (CRD – *Central Data Repository*), zwana także korporacyjną hurtownią danych (KHD). Koncepcja ta obejmuje fizyczne kopiowanie danych z systemów źródłowych do centralnego repozytorium danych. Zasilanie odbywa się poprzez procesy ETL (*Extraction, Transforming and Loading*) odpowiedzialne za odczyt danych ze źródeł (*Extraction*), transformację do modelu używanego w repozytorium danych ze sprawdzeniem poprawności i zapewnieniem spójności danych (*Transforming*) oraz wczytanie danych do repozytorium (*Loading*). Następnie dane te są udostępniane aplikacjom użytkowników (rys. 2a). Procesy ETL odpowiedzialne są także za wykrywanie zmian w źródłach poprzez mechanizmy wyzwalaczy (źródła aktywne – *active sources*), dzienniki operacji (*logged sources*), mechanizmy przepytывania (*queryable sources*) oraz mechanizmy migawek (*snapshot sources*).

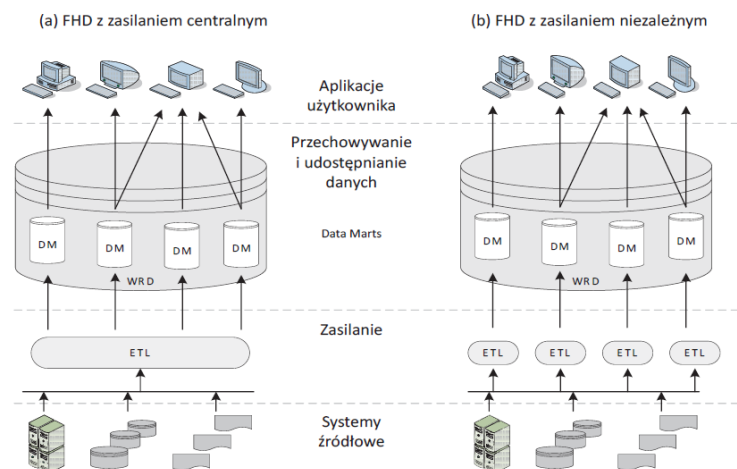
Bardziej rozbudowaną wersją KHD jest wariant hierarchiczny wzbogacony o warstwę hurtowni tematycznych (DM – *Data Marts*) (rys. 2b).



**Rysunek 2.** Warianty korporacyjnej hurtowni danych [8].

Podobnie jak w przypadku KHD można wyróżnić dwa warianty FHD: wariant ze wspólnymi procesami zasilania (rys. 3a) oraz wariant z niezależnymi procesami zasilania (rys. 3b). W tym przypadku centralne repozytorium danych zastąpione jest przez jego formę wirtualną (WRD),

umożliwiającą na dostęp do danych zlokalizowanych fizycznie w hurtowniach tematycznych. W koncepcji federacji hurtowni danych hurtownie tematyczne zasilane niezależnie są traktowane jako niezależne korporacyjne hurtownie danych.



**Rysunek 3.** Warianty federacyjnej hurtowni danych [8].

Dla potrzeb analizy danych opracowano systemy OLAP. W architekturze OLAP najczęściej stosuje się [9]:

- MOLAP (*Multidimensional On-Line Analytic Processing*)- do prowadzenia analiz wykorzystuje serwer wielowymiarowej bazy danych,
- DOLAP (*OLAP Desktop*) to MOLAP operujący na plikach lub bazach danych zainstalowanych lokalnie, na stacjach klienta (brak serwera),
- ROLAP (*Relational On-Line Analytic Processing*) do prowadzenia analiz i wielowymiarowej

prezentacji korzysta z danych przechowywanych w relacyjnej bazie danych. Dostęp do danych jest realizowany na poziomie zwykle złożonych zapytań SQL,

- HOLAP (*Hybrid OLAP*) – to architektura mieszana pozwalająca użytkownikom na korzystanie z dwóch baz danych: wielowymiarowej lub relacyjnej.

#### 4. Case study - Hurtownia danych w planowaniu produkcji

Tworzenie hurtowni danych może być oparte na następujących etapach (por. [1]):

- określenie wymagań przedsiębiorstwa,
- określenie fizycznego projektu hurtowni danych,
- analiza, utworzenie i integracja potrzebnych źródeł danych,
- utworzenie hurtowni danych,
- wybór i budowa narzędzi dostępowych dla użytkowników,
- testowanie i ewentualne poprawki.

Wybrane przedsiębiorstwo zajmuje się działalnością produkcyjną w obszarze produkcji wyrobów z tworzyw sztucznych w produkcji mieszanej na zamówienie klienta. Dlatego wymagana jest analiza danych w środowisku systemów wspomagania decyzji dotycząca klientów oraz sprzedawanych i produkowanych wyrobów. Informacje te potrzebne są do zminimalizowania kosztów produkcji i lepszego zrozumienia potrzeb rynków, dotyczących najbardziej popularnych produktów firmy, jak również do zoptymalizowania oferty produkowanych wyrobów poprzez ograniczenie najmniej opłacalnej produkcji, i skupienie się na wyrobach generujących większy zysk.

Planowanie strategiczne (zgrubne) pobiera dane z harmonogramów sprzedaży pochodzących od klientów. Następnie na podstawie tych harmonogramów generowany jest plan główny w przedsiębiorstwie. Wersje planu głównego mogą być wykorzystywane do definiowania różnych prognoz i scenariuszy oraz do symulacji w powiązaniu z planowaniem głównym i zgrubnym planowania zdolności produkcyjnych. Głównym celem zgrubnego planowania zasobów produkcyjnych jest zrównoważenie zasobów, które zmieniają się w długim okresie, z istniejącym popytem. Typowymi zasobami

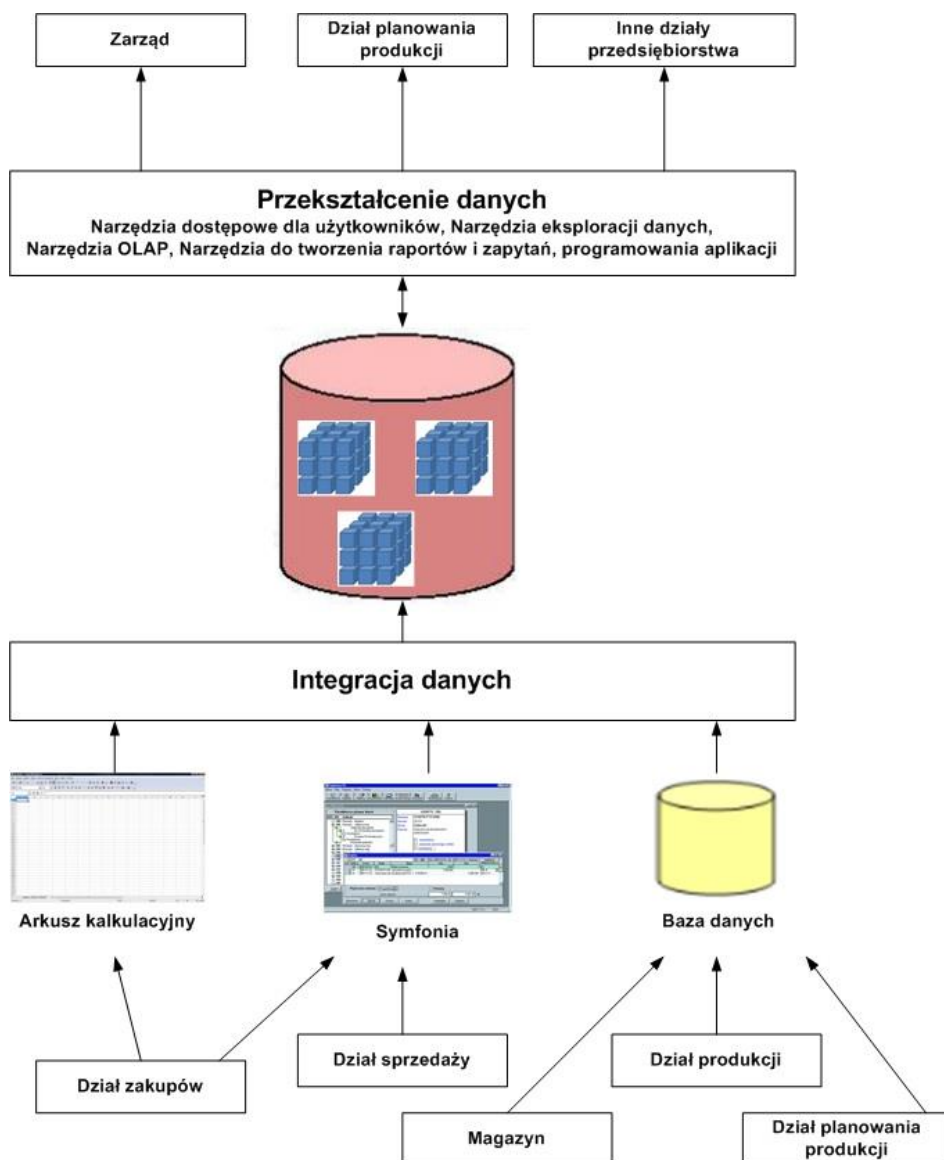
są: zdolności produkcyjne maszyn, powierzchnie magazynowe, personel produkcyjny (koszty bezpośrednie, pośrednie), czy zdolności transportowe.

Idąc dalej planowanie z kilkunastu tygodni przychodzi na planowanie do 4 tygodni, wykorzystując mechanizmy MRP (planowanie potrzeb materiałowych) i CRP (planowanie zdolności produkcyjnych). Bardzo ważnym elementem jest kalkulacja kosztów produkcji.

Szczegółowe wymagania w obszarze planowania produkcji mogą zostać zrealizowane przez utworzenie trzech wielowymiarowych kostek danych, zawierających sprzedaż wyrobów, zakupy materiałów i kalkulację kosztów. Wymaga to zebrania danych z działów firmy: działu sprzedaży, magazynów, produkcji, zakupów i również planowania produkcji. Schemat źródła informacji hurtowni danych wraz z ich przeznaczeniem przedstawia rysunek 4.

#### 5. Podsumowanie

System hurtowni danych jest nie tylko prostym narzędziem do raportowania. Jest to środowisko gromadzące dane z wielu źródeł w przedsiębiorstwie. W hurtowni danych przechowywane są dane historyczne, charakteryzujące się wysoką jakością i zoptymalizowane w taki sposób, aby dostarczyć wymagane informacje w jak najkrótszym czasie. Takie informacje stanowią doskonałe źródło na temat działania przedsiębiorstwa. Ponadto są niezbędne do podejmowania właściwych decyzji, zapewniając monitorowanie i kontrolę realizowanych przez przedsiębiorstwo zadań w szczególności w planowaniu produkcji.



**Rysunek 4.** Źródła informacji hurtowni danych oraz ich przeznaczenie.

## Literatura

1. Poe V., Klauer P., Brobst S. Tworzenie hurtowni danych. Wyd. WNT, Warszawa 2000.
2. Brzeziński M. Sterowanie produkcją, Wyd. Politechniki Lubelskiej, Lublin 2009.
3. Pająk E. Zarządzanie produkcją. Produkt, technologia organizacja, Wyd. PWN, Warszawa 2006.
4. Inmon W., Hackathorn R. Using the data warehouse. Wyd. John Wiley & Sons, New York 1994.
5. Królikowski Z. Hurtownie danych logiczne i fizyczne struktury danych, Wyd. Politechniki Poznańskiej, Poznań 2007.
6. Pelikant A. Hurtownie danych: od przetwarzania analitycznego do raportowania, Wyd. Helion, Warszawa 2011.
7. Chodkowska-Gyurics A. Hurtownie danych: teoria i praktyka, Wyd. PWN, Warszawa 2017.
8. Dymek D., Komnata W., Kotulski L. Federacyjna hurtownia danych w dostępie do informacji poufnej. Kolegium Analiz Ekonomicznych, Szkoła Główna Handlowa w Warszawie, 2014; 33.
9. Gorawski M. Ocena efektywności architektur OLAP. e-Informatica Software Engineering Journal, <https://www.e-informatyka.pl/index.php/pimio/ocena-systemow/ocena-efektywnosci-architektur-olap/>, (dostęp: 25.09.2019).



# ZAPIS LOGICZNY UKŁADU KOMBINACYJNEGO METODĄ REPREZENTACJI BINARNYCH

Janusz Łukowski

Uniwersytet Kazimierza Wielkiego  
Instytut Mechaniki i Informatyki Stosowanej  
ul.Kopernika 1 p.207a, 85-074 Bydgoszcz  
e-mail: januszl@ukw.edu.pl

**Streszczenie:** W artykule przedstawiono nową metodę uproszczania zapisu logicznego układu kombinacyjnego z wykorzystaniem tabeli prawdy podstawowych funkcyj logicznych lub/i reprezentacji binarnej kombinacji wejść. Metoda reprezentacji binarnych stanowi alternatywny sposób konstrukcji uproszczonego zapisu funkcji wyjściowej układu kombinacyjnego w odniesieniu do metody przekształceń formalnych, metody tablic Karnaugh'a czy metody Quine'a–McCluskeya.

**Słowa kluczowe:** Tabela prawdy układu kombinacyjnego, uproszczony zapis logiczny układu kombinacyjnego, metoda reprezentacji binarnych

## Logical description of a combinational system by the binary representation method

**Abstract:** The article presents a new method for simplifying the logical description of a combinational system using truth tables of basic logic functors or / and binary representation of an input combination. The binary representation method is an alternative way of constructing a simplified description of the output function of a combinational system in relation to the method of formal transformations, the Karnaugh table or the Quine-McCluskey methods.

**Key words:** Truth table of combinational system, reduced logical description of combinational system, method of binary

### 1. Wprowadzenie

Układy kombinacyjne [1, 2, 3, 4, 5] należą do układów logicznych, w których każda kombinacja wartości zmiennych wejściowych jednoznacznie określa kombinację wartości zmiennych wyjściowych. Działanie układu kombinacyjnego opisuje funkcja logiczna (funkcja boolowska będąca matematycznym modelem układu kombinacyjnego), w której zmienne i one sama przyjmują wartości ze zbioru  $\{0,1\}$ . Funkcje logiczne mogą być przedstawione w postaci opisu słownego, tabeli prawdy w sposób analityczny. Funkcje logiczne można upraszczać korzystając z podstawowych własności algebry Boole'a (metoda przekształceń formalnych [1,2,4,8]), wykorzystując metodę tablic Karnaugh'a [7], metodę Quine'a–McCluskeya [6] lub metodę bezpośredniego przeszukiwania [4]. Do najprostszych układów kombinacyjnych należą bramki logiczne (digital logic gates) realizujące funkcje logiczne jednej lub wielu zmiennych. W pracy przedstawiono nową metodę uproszczania postaci analitycznej funkcji logicznej wykorzystującej funkcje logiczne bramek logicznych oraz tabelę prawdy układu kombinacyjnego. Metodę tę określono mianem metody reprezentacji binarnych.

### 2. Koncepcja metody reprezentacji binarnych

Elementarna funkcja logiczna jest funkcją dwóch zmiennych wejściowych (z wyjątkiem funkcji logicznej negacji, która jest jednoargumentowa) i jest realizowana fizycznie przez funkcyj logiczne koniunkcji (AND), alternatywy (OR) oraz negacji (NOT). Stabelaryzowana zależność funkcji wyjściowej  $y$  funkcyj logicznych w funkcji zmiennych wejściowych  $x_1x_0$  (Tabela 1) stanowi podstawowy zestaw 4-rech kombinacji zmiennych wejściowych i tworzący bazowy element metody reprezentacji binarnych.

Ilość zestawów oblicza się według formuły:

$$il\_zestaw = 2^n/4,$$

gdzie:  $n$  - ilość wejść układu.

Każdy zestaw 4-ech kolejnych kombinacji (stanowiących jeden zestaw) może być rozpatrywany indywidualnie w procesie tworzenia zminimalizowanej postaci funkcji logicznej lub jako grupa łączona  $2^1, 2^2, \dots, 2^n$  zestawów.

Tabela 1. Tabela prawdy funkcji logicznych

| $x_1$            | $x_0$         | y |                 |     |      |     |     |      |
|------------------|---------------|---|-----------------|-----|------|-----|-----|------|
| 0                | 0             | 0 | 0               | 0   | 1    | 1   | 0   | 1    |
| 0                | 1             | 0 | 1               | 1   | 1    | 0   | 1   | 0    |
| 1                | 0             | 0 | 1               |     | 1    | 0   | 1   | 0    |
| 1                | 1             | 1 | 1               |     | 0    | 0   | 0   | 1    |
| funkcja logiczna | $y = x_1 x_0$ |   | $y = x_1 + x_0$ |     |      |     |     |      |
|                  | AND           |   | OR              | NOT | NAND | NOR | XOR | XNOR |

Istotą proponowanej metody zapisu wyrażenia boolowskiego dla funkcji logicznej wielu zmiennych jest:

- podział wszystkich kombinacji funkcji logicznej wielow wejściowej na zestawy zawierające po 4 kolejne kombinacje układu kombinacyjnego;
- możliwość wykorzystania do jego konstrukcji zapisu funkcji logicznej jednego z podstawowych dwuargumentowych funktorów logicznych lub konfiguracji jednej ze zmiennych wejściowych w postaci tabeli prawdy (rys.1);
- możliwość łączenia zestawów w grupy rozpatrywane jako jeden zestaw.

W przypadku wystąpienia identycznej postaci analizowanej funkcji wyjściowej w ramach zestawu jak postać podstawowego funktora logicznego czy też układu kombinacji jednej zmiennej wejściowej celem uwzględnienia pozostałych zmiennych w konstrukcji wyrażenia boolowskiego traktujemy zestaw jako taki, w którym stan wyjścia przyjmuje stan wysoki dla wszystkich 4-ech kombinacji.

Przykład 1.

| $x_2$ | $x_1$ | $x_0$ | y |
|-------|-------|-------|---|
| 0     | 0     | 0     | 0 |
| 0     | 0     | 1     | 1 |
| 0     | 1     | 0     | 1 |
| 0     | 1     | 1     | 1 |
| 1     | 0     | 0     | 0 |
| 1     | 0     | 1     | 0 |
| 1     | 1     | 0     | 1 |
| 1     | 1     | 1     | 1 |

W przykładzie 1 dla kombinacyjnego układu trójwejściowego ( $x_2x_1x_0$ ) funkcja wyjściowa y ma postać: .Tabela prawdy funkcji logicznej y posiada dwa zestawy 4-ech kombinacji. Postać pierwszego zestawu jest taka sama jak tablica prawdy funkcji OR dla zmiennych, zaś drugiego jest identyczna jak układ kombinacji zmiennej wejściowej. W celu uwzględnienia pozostałych zmiennych wejściowej w zapisie wyrażenia boolowskiego przyjmujemy stan wysoki na wyjściu układu dla wszystkich kombinacji (dwóch zestawów), dzięki czemu dla zestawu pierwszego w zapisie wprowadzamy zmienną  $x_2$  zanegowaną (stan przeciwny do stanu wysokiego wyjścia), a dla zestawu drugiego  $x_2$  w postaci prostej.

Brak zmiennej wejściowej  $x_0$  w zapisie funkcji boolowskiej dla drugiego zestawu wynika z faktu, iż zmienna ta posiada dwie różne wartości w obrębie rozpatrywanego zestawu.

### 3. Procedura konstrukcji uproszczonej postaci wyrażenia boolowskiego

W przypadku zgodności tabeli prawdy funkcji wyjściowej z postacią tabeli prawdy funktora logicznego lub układu kombinacji jednej ze zmiennych wejściowych procedura konstrukcji postaci uproszczonej ma następującą strukturę:

- podział kombinacji układu na zestawy po 4 kombinacje każdy;
- identyfikacja zgodności tabeli prawdy podstawowej funkcji boolowskiej lub układu kombinacji jednej ze zmiennych wejściowych z postacią funkcji wyjściowej w obrębie zestawu;
- w przypadku zgodności uwzględnienie pozostałych zmiennych wejściowych w zapisie logicznym funkcji, przy założeniu stanu wysokiego funkcji wyjściowej dla każdej kombinacji w obrębie zestawu.

W przypadku braku zgodności tabeli prawdy funkcji wyjściowej z postacią tabeli prawdy funktora logicznego lub układu kombinacji jednej ze zmiennych wejściowych dokonujemy podziału kombinacji układu na zestawy po 4 kombinacje każdy. W obrębie 1-go zestawu (Przykład 2) zapis funkcji logicznej (w postaci kanonicznej lub dysjunkcyjnej) może być konstruowany dla łącznej ilości kombinacji zgodnie z wagami binarnego zapisu pozycyjnego tzn.:  $2^0$ ,  $2^1$ ,  $2^2$ . Konstrukcja funkcji logicznej z wagą  $2^0$ , wymaga konstrukcji wyrażenia boolowskiego dla jednej kombinacji zmiennych wejściowych w ramach jednego zestawu. Konstrukcja wyrażenia boolowskiego z wagą  $2^1$ , pozwala na łączne rozpatrywanie 2-ch dowolnych kombinacji w obrębie jednego zestawu o identycznej wartości stanu wyjściowego. Zaś w przypadku wystąpienia 3 identycznych stanów wyjściowych, to konstrukcja wyrażenia boolowskiego jest realizowana w konfiguracji  $2^1 + 2^0$ . W przypadku łącznego rozpatrywania kombinacji funkcja logiczna zawiera tylko te zmienne wejściowe, których wartość jest identyczna lub zanegowana w odniesieniu do wartości analizowanego stanu wyjściowego.

Przykład 2.

| $x_1$ | $x_0$ | $y$ |
|-------|-------|-----|
| 0     | 0     | 0   |
| 0     | 1     | 1   |
| 1     | 0     | 0   |
| 1     | 1     | 1   |

#### 4. Case study – Rozproszone systemy baz danych w planowaniu produkcji

Przedstawiona nowa metoda uproszczonego zapisu logicznego układu kombinacyjnego umożliwia bezpośrednią konstrukcję zapisu logicznego w oparciu o tabelę prawdy funktorów logicznych lub układ kombinacji binarnej jednego z wejść układu. W przypadku niezgodności postaci tabeli prawdy podstawowych funktorów logicznych lub układu kombinacji binarnej wejść przedstawiono technikę łączenia kombinacji układu w ramach zestawu.

#### Literatura

1. Kalisz J. Podstawy elektroniki cyfrowej. WKŁ, Warszawa 1993.
2. Majewski W. Układy logiczne, WNT, Warszawa 1993.
3. Mano M. M. Computer engineering: hardware design. Prentice-Hall, Upper Sadle River 1988.
4. Traczyk W. Układy cyfrowe. Podstawy teoretyczne i metody syntezy. WNT, Warszawa 1986.
5. Bromirski J. Teoria automatów. WNT, Warszawa 1969.
6. McCluskey E. Logic design principles, Prentice-Hall, Upper Sadle River 1986.
7. Kamionka-Mikuła H., Małysiak H., Pochopień B. Synteza i analiza układów cyfrowych. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2009.
8. Adamowicz A., Zbierski P. Logic of mathematics. A modern course of classical logic. A Wiley-Interscience Publication. John Wiley & Sons, Inc., Nowy York 1997, seria: Pure and Applied Mathematics.

# WYBRANE METODY SZTUCZNEJ INTELIGENCJI ZAIMPLEMENTOWANE W JĘZYKU PROGRAMOWANIA C/C++

Aleksandra Mrela

Uniwersytet Kazimierza Wielkiego  
Instytut Informatyki  
ul. Mikołaja Kopernika 1, 85-074 Bydgoszcz  
e-mail: aleksandra.mrela@ukw.edu.pl

**Streszczenie:** Obecnie coraz częściej wykorzystuje się metody sztucznej inteligencji (AI) do budowania systemów ekspertowych, czy opartych na wiedzy. Jednakże, bardzo często podczas zajęć w szkole podstawowej i średniej podczas zajęć z programowania, lub ogólniej kształcenia myślenia komputacyjnego, uczniowie uczą się algorytmów rozwiązujących problemy w warunkach pewności. Wobec tego większość młodych ludzi nie ma okazji rozważania rozwiązań problemów w zakresie logiki rozmytej. Aby nauczyciele informatyki rozważali z uczniami metody sztucznej inteligencji, należy przygotować proste przykłady algorytmów AI. W artykule przedstawiono kilka prostych przykładów zbiorów i relacji rozmytych oraz prostego systemu wnioskującego zakodowanych w C++.

Słowa kluczowe: Edukacja, logika rozmyta, systemy wnioskujące, C++

## Selected methods of artificial intelligence implemented in C/C++ programming language

**Abstract:** Currently, artificial intelligence (AI) methods are increasingly used to build expert or knowledge-based systems. However, very often during elementary and high school classes while programming classes, or more general, computational thinking training, students learn algorithms that solve problems in conditions of certainty. Therefore, most young people have no opportunity to consider solutions to fuzzy logic problems. For IT teachers to discuss artificial intelligence methods with students, simple AI algorithms should be prepared. The article presents some simple examples of fuzzy sets and relations as well as a simple inference system coded in C++.

Keywords: Education, fuzzy logic, inference systems, C++

### 1. Wprowadzenie

Nauka programowania w szkole koncentruje się nauce myślenia komputacyjnego służącego do rozwiązywania różnych problemów, których rozwiązanie polega na znalezieniu algorytmu do ich rozwiązania [1]. Uczniowie, po opracowaniu algorytmu lub jego fragmentu, kodują je korzystając z wybranej technologii. Zgodnie z podstawą programową, uczeń w szkole średniej powinien stosować algorytmy dotyczące m.in. badania czy dana liczba jest pierwsza, wyznaczania NWD i NWW, szyfrowania tekstu metodą Cezara, porządkowania ciągu liczb, wydawania reszty czy obliczania wartości ciągu metodą iteracyjną czy rekurencyjną [2].

Obecnie, w dobie gwałtownego rozwoju sztucznej inteligencji, uczniowie powinni uczyć się także rozwiązywać zadania w warunkach niepewności. Jak twierdzi M. M. Sysło, „przybliżenie sztucznej inteligencji kręgom

edukacyjnym z przekonaniem, że ta dziedzina czeka na uwzględnienie jej w realizacji powszechnego kształcenia wszystkich uczniów” [3]. W artykule omówiono przykłady, zakodowane w C++, zbiorów i relacji rozmytych oraz prostego systemu wnioskującego.

### 2. Zbiory rozmyte

Idea zbiorów rozmytych została wprowadzona i opracowana przez L. A. Zadeha w 1965 r. [5].

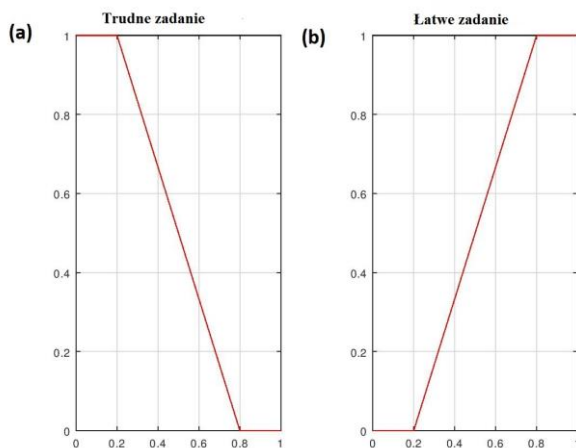
Zbiorem rozmytym  $A$  w przestrzeni  $X$  jest zbiór par uporządkowanych  $\{(x, \mu_A(x)), x \in X\}$ , gdzie  $\mu_A(x): X \rightarrow [0,1]$  jest funkcją przynależności.

Wartość  $\mu_A(x)$  określa stopień przynależności elementu  $x$  do zbioru  $A$ . Powyższa definicja jest z pewnością trudna do zrozumienia dla uczniów, ale poprzez różne przykłady można ją im przybliżyć. Rozważmy przykład zbioru rozmytego z dziedziny bliskiej uczniom, a mianowicie zbioru „trudne

zadanie”. Niech  $X = [0,1]$  i niech funkcja przynależności będzie określona w następujący sposób:

$$\mu_{\text{trudne zadanie}}(x) = \begin{cases} 1 & x < 0,2 \\ -\left(\frac{5}{3}\right)x + \frac{4}{3} & 0,2 \leq x \leq 0,8 \\ 0 & x > 0,8. \end{cases}$$

Wykres funkcji przynależności zbioru rozmytego „trudne zadanie” jest przedstawiony na rys. 1 (a).



Rysunek. 1 Wykresy funkcji przynależności zbiorów rozmytych: (a) trudne zadanie, (b) łatwe zadanie.

Po przeanalizowaniu wzoru funkcji i ewentualnie wykresu, można zakodować jej formułę w C++ następująco:

```
double x;
double trudne_zadanie;
if (x < 0.2)
    trudne_zadanie=1;
else if (x < 0.8)
    trudne_zadanie = -(5/3) * x + 4/3;
else
    trudne_zadanie = 0;
```

Zatem uczniowie mogą, podając różne wartości  $x$ , wyznaczać wartości funkcji przynależności. Podobnie, można zdefiniować funkcję przynależności zbioru rozmytego „łatwe zadanie” wzorem

$$\mu_{\text{łatwe zadanie}}(x) = \begin{cases} 0 & x < 0,2 \\ \left(\frac{5}{3}\right)x - \frac{1}{3} & 0,2 \leq x \leq 0,8 \\ 1 & x > 0,8. \end{cases}$$

Wykres funkcji jest przedstawiony na rys. 1 (b), a kod w C++ jest zapisany w następujący sposób:

```
double x;
double latwe_zadanie;
if (x < 0.2)
    latwe_zadanie = 0;
else if (x < 0.8)
    latwe_zadanie = (5/3) * x - (1/3);
else
    latwe_zadanie = 1;
```

Wobec tego dla różnych wartości  $x$  uczniowie mogą określać stopień przynależności do zbiorów rozmytych „zadanie łatwe” i „zadanie trudne”.

Uczniowie oczywiście będą pytać jak wyznaczyć wartość  $x$ . Rozważmy przykład. W klasie jest 30 uczniów, którzy rozwiązywali zadanie z matematyki. Przyjmijmy, że  $x$  jest równe stosunkowi uczniów, którzy rozwiązali poprawnie wybrane zadanie, do wszystkich uczniów. Zatem, jeśli 10 studentów rozwiązało to zadanie, to

$$x = \frac{10}{30} = 0,3333,$$

więc  $\mu_{\text{trudne zadanie}}(0,3333) = 0,7778$  oraz

$$\mu_{\text{łatwe zadanie}}(0,3333) = 0,2222.$$

Wobec tego powyższe zadanie jest trudne w 78% i łatwe w 22%. Oczywiście, pierwsze spotkanie ze zbiorami rozmytymi może być trudne dla uczniów, to po rozważeniu kilku przykładów, w tym zakodowaniu kilku algorytmów pojęcie zbioru rozmytego stanie się łatwe.

### 3. Relacje rozmyte

Jeśli rozważaną przestrzenią jest iloczyn kartezjański  $X \times Y$ , to zbiór rozmyty nazywamy relacją rozmytą. Rozważmy, bliską uczniom, relację rozmytą  $R_1 = \text{„Wiedza } \times \text{ Zadanie”}$ , gdzie  $X = \text{„Wiedza”} = \{X_1 - \text{Miejsca zerowe funkcji liniowych}, X_2 - \text{Wartości funkcji liniowych}, X_3 - \text{Wykres funkcji liniowych}\}$  i  $Y = \text{Zadanie} = \{Z_1, Z_2, Z_3, Z_4\}$ . Niech relacja  $R_1$  będzie zdefiniowana w następujący sposób

$$R_1 = \begin{bmatrix} 0,8 & 0,3 & 0,2 & 0,4 \\ 0,5 & 0,6 & 0,7 & 0,8 \\ 0,9 & 0,5 & 0,6 & 0,7 \end{bmatrix}$$

Niech zadanie 1 ( $Z_1$ ) będzie określone następująco: Niech  $y = 2x + 1$ . Wyznaczyć miejsca zerowe funkcji  $y$ , obliczyć  $y(0)$  oraz narysować wykres tej funkcji. Na tej podstawie przyjęto, że

$R_1(\text{Miejsca zerowe}, Z_1) = 0,8;$   
 $R_1(\text{Wartości funkcji}, Z_1) = 0,5;$   
 $R_1(\text{Wykres}, Z_1) = 0,9.$

Podczas dyskusji z uczniami, można ustalić inne wartości rozważanej relacji. Relację  $R_1$ , wraz z operacją wyświetlenia wartości tej relacji na ekranie monitora, można zakodować w C++ następująco:

```
double R1[3][4];
R1[0][0]=0.8; R1[0][1]=0.3;
R1[0][2]=0.2; R1[0][3]=0.4;
R1[1][0]=0.5; R1[1][1]=0.6;
R1[1][2]=0.7; R1[1][3]=0.8;
R1[2][0]=0.9; R1[2][1]=0.5;
R1[2][2]=0.6; R1[2][3]=0.7;
for (int i=0; i<3; i++)
{
    for (int j=0; j<4; j++)
    {
        cout << "R1[" << i << ", " << j << "]=" <<
        relacja_R1[i][j] << endl;
    }
}
```

Zbudujemy także relację rozmytą  $R_2$  między zadaniami  $Y$  a uczniami  $Z = \{U_1, U_2, U_3, U_4, U_5\}$ , która przyjmuje dwie wartości: **1** oznacza, że uczeń rozwiązał zadanie poprawnie, a **0**, że uczeń nie rozwiązał zadania. Niech relacja  $R_2$  będzie określona następująco

$$R_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$

którą można zakodować w C++ w następujący sposób

```
double R2[4][5];
R2[0][0]=1; R2[0][1]=0; R2[0][2]=1;
R2[0][3]=1; R2[0][4]=0;
R2[1][0]=1; R2[1][1]=1; R2[1][2]=0;
R2[1][3]=1; R2[1][4]=1;
R2[2][0]=1; R2[2][1]=1; R2[2][2]=1;
R2[2][3]=0; R2[2][4]=1;
R2[3][0]=0; R2[3][1]=0; R2[3][2]=1;
R2[3][3]=1; R2[3][4]=1;
for (int i=0; i<4; i++)
{
    for (int j=0; j<5; j++)
    {
        cout << "R2[" << i << ", " << j <<
        "]" << relacja_R2[i][j] << endl;
    }
}
```

Uczniowie chcą ocenić poziom wiedzy z zakresu funkcji liniowych. Do wyznaczenia relacji

$R_3 = X \times Z$  wykorzystamy  $S - T$ -złożenie relacji  $R_1$  i  $R_2$ , wykorzystując jako  $S$ -konormę funkcję  $\max$ , a jako  $T$ -normę funkcję  $\min$ . Wobec tego, zgodnie z [4], złożenie relacji  $\max - \min$  jest zapisane następująco:

$$R_3(x, z) = \max_{y \in Y} (\min(R_1(x, y), R_2(y, z))).$$

Złożenie relacji  $\max - \min$  jest dosyć trudne do zakodowania, ponieważ wymaga zastosowania trzech pętli, ale oczywiście jest możliwe do opracowania dla uczniów:

```
double R3[3][5];
for (int i=0; i<3; i++)
{
    for (int j=0; j<5; j++)
    {
        R3[i][j]=0;
        for (int k=0; k<4; k++)
        {
            R3[i][j]=max(R3[i][j],
            min(R1[i][k], R2[k][j]));
        }
    }
}
```

Wobec tego, relacja  $R_3$  jest równa:

$$R_3 = \begin{bmatrix} 0,8 & 0,3 & 0,8 & 0,8 & 0,4 \\ 0,7 & 0,7 & 0,8 & 0,8 & 0,8 \\ 0,9 & 0,6 & 0,9 & 0,9 & 0,7 \end{bmatrix}.$$

Rozważmy wartość  $R_3(X_1, U_1) = 0,8$ , która oznacza, że uczeń  $U_1$  posiadał wiedzę na temat miejsc zerowych funkcji liniowych na poziomie **80%**. Podobnie, jak w przypadku zbiorów rozmytych, można stworzyć ze studentami kilka przykładów relacji, zakodować je w C++ i opracować interpretacje uzyskanych wyników algorytmu w języku C++.

#### 4. Systemy wnioskujące

Wraz z uczniami można zastosować i zakodować prosty schemat wnioskowania rozmytego, typu Tagano-Sugeno [4]. Załóżmy, że określamy poziom trudności zadania  $x$  jako stosunek liczby osób, które rozwiązały dane zadanie, to liczby osób w grupie czy klasie. Poziom rozwiązania zadania oznacza liczbę punktów  $y$  jaką otrzymał rozważany uczeń transformowaną do przedziału **[0,1]**. Ponadto, można określić próg zdawalności zadania, czy wymagany poziom znajomości tematu  $z$ . Jako przesłanki zdefiniujemy cztery zbiory rozmyte „łatwe\_zadanie”, „trudne\_zadanie”, „słaba\_odpowiedz” i „dobra\_odpowiedz”, co może być zakodowane w następujący sposób:

```

double latwe_zadanie,
trudne_zadanie, slaba_odpowiedz,
dobra_odpowiedz;
if (x < 0.2)
    trudne_zadanie=1;
else if (x < 0.8)
    trudne_zadanie = -(5/3) * x + 4/3;
else
    trudne_zadanie = 0;
if (x < 0.2)
    latwe_zadanie=0;
else if (x < 0.8)
    latwe_zadanie = (5/3) * x - (1/3);
else
    latwe_zadanie = 1;
if (y < 0.3)
    slaba_odpowiedz=1;
else if (y < 0.7)
    slaba_odpowiedz = -2.5 * y + 1.75;
else
    slaba_odpowiedz = 0;
if (y < 0.3)
    dobra_odpowiedz=0;
else if (y < 0.7)
    dobra_odpowiedz = 2.5 * y - 0.75;
else
    dobra_odpowiedz = 1;

```

Następnie w celu zdefiniowania następnika stworzymy macierz  $Y[3]$  złożoną z 4 wartości określających 4 różne poziomy wiedzy określone regułami:

*Jeśli zadanie jest łatwe i odpowiedź jest słaba, to ocena jest niedostateczna, czyli  $Y[0] = 2$ .*  
*Jeśli zadanie jest łatwe i odpowiedź jest dobra, to ocena jest dostateczna, czyli  $Y[1] = 3$ .*  
*Jeśli zadanie jest trudne i odpowiedź jest słaba, to ocena jest dobra, czyli  $Y[2] = 4$ .*  
*Jeśli zadanie jest trudne i odpowiedź jest dobra, to ocena jest bardzo dobra, czyli  $Y[3] = 5$ .*

W ostatniej części naszego programu, określamy wagi przy pomocy funkcji min, wyznaczamy ostateczną ocenę jako średnią ważoną i porównujemy ją z ustalonym wcześniej poziomem znajomości materiału:

```

double w[3], Y[3];
w[1]=min(latwe_zadanie, slaba_odpowiedz);
Y[0]=2;
w[2]=min(latwe_zadanie, dobra_odpowiedz);
Y[1]=3;
w[3]=min(trudne_zadanie, slaba_odpowiedz);
Y[2]=4;
W[4]=min(trudne_zadanie, dobra_odpowiedz);
Y[3]=5;
double ostateczna_ocena;

```

```

ostateczna_ocena=(w[0]*Y[0] + w[1]*Y[1]+
w[2]*Y[2] + w[3]*Y[3])/(w[0]+w[1]+w[2]+w[3]);
cout << "ostateczna ocena wynosi " <<
ostateczna_ocena << endl;

```

```

if (ostateczna_ocena >= z)
    cout << "Umiesz ten rozdział wystarczająco
dobrze. Przejdź do następnego rozdziału";
else
    cout << "Musisz powtorzyć materiał jeszcze
raz";

```

Po zakodowaniu systemu wnioskującego, można pokazać uczniom, że wnioskowanie rozmyte nie musi być trudne i mogą zaimplementować prosty system wnioskujący, a następnie można stworzyć bardziej skomplikowane systemy wnioskujące.

## 5. Podsumowanie

Uczniowie szkół średnich, a nawet podstawowych, powinni kształcić proste metody sztucznej inteligencji. Obecnie, wykorzystuje się coraz częściej systemy wnioskujące, więc uczniowie powinni rozumieć podstawy zbiorów i relacji rozmytych. W szkole rozpatruje się tylko jeden z rodzajów niepewności, a mianowicie rachunek prawdopodobieństwa, natomiast w życiu codziennym znajdujemy się w sytuacjach innego rodzaju niepewności, który może być opisany metodami logiki rozmytej.

Język programowania C++ doskonale nadaje się do kodowania zbiorów, czy relacji rozmytych oraz wyznaczania ich sum, różnic, czy złożzeń, a także do zbudowania prostego systemu wnioskującego z uczniami podczas zajęć z informatyki. Należy jednakże pamiętać o opracowaniu pojęć logiki rozmytej oraz stosowanych metod na bardzo niskim poziomie abstrakcji.

## Literatura

1. Drab T., „Informatyka oparta na rachunkach”, Materiały XVI konferencji Informatyka w Edukacji, Toruń 2019, <https://iwe.mat.umk.pl/tom-iwe2019/19.pdf>, dostęp 11.10. 2019 r.
2. Podstawa programowa, Informatyka, liceum/technikum, Podstawa programowa.pl, <https://podstawaprogramowa.pl/Liceum-technikum/Informatyka>, dostęp 11.10.2019 r.
3. Sysło M.M., „Inteligencja +”, Materiały XVI konferencji Informatyka w Edukacji, Toruń

- 2019, <https://iwe.mat.umk.pl/tom-iwe2019/03.pdf>, dostęp 11.10.2019 r.
4. Rutkowski L., Metody i techniki sztucznej inteligencji, PWN, Warszawa 2012.
  5. Zadeh L. A., Fuzzy sets, Information and Control, 8 (3), 1965, str. 338-353.

# IMPLEMENTACJA ŚRODOWISKA TESTOWEGO W JĘZYKU R

Jan Baumgart

Student 1 roku MU Informatyka

Uniwersytet Kazimierza Wielkiego

Wydział Matematyki, Fizyki i Techniki

e-mail: [janek@ukw.edu.pl](mailto:janek@ukw.edu.pl)

**Streszczenie:** W pracy przedstawione zostały biblioteki oraz ich zastosowanie do implementacji środowiska testowego algorytmów rojowych. Wykorzystane moduły pozwalają przygotować w pełni sprawne narzędzie w szybki sposób. Przedstawione środowisko testowe w tym artykule zostało przygotowane do wspomagania badań nad algorytmami rojowymi i może być wykorzystane jako zasób sieciowy jak i skrypt dostępny lokalnie. Metody wykorzystane w tym artykule mogą posłużyć do budowy środowisk testowych dla wielu innych nie związanych z algorytmami rojowymi scenariuszy.

**Słowa kluczowe:** R, Shiny R, Shiny Dashboard, DT, Algorytmy, Środowiska testowe.

## Implementation of a test environment in R

**Abstract:** The work presents libraries and their application for implementing the swarm algorithms test environment. The modules used allow you to prepare a fully functional tool quickly. The test environment presented in this article has been prepared to support research on swarm algorithms and can be used as a network resource as well as a locally available script. The methods used in this article can be used to build test environments for many other scenery not related to swarm algorithms.

**Keywords:** R, Shiny R, Shiny Dashboard, DT, Algorithms, Test environments.

### 1. Wprowadzenie

Podczas rozwiązywania wielu problemów naukowych często trudnym zadaniem jest przedstawienie wyników w sposób przystępny. W każdym projekcie niezależnie czy w środowisku korporacyjnym czy naukowym nadchodzi moment prezentacji danych celem omówienia osiągniętych wyników. Dlatego często za kluczowe uważa się przygotowanie środowiska testowego które pozwoli na czytelną reprezentację efektów naszej pracy. Zaprojektowanie i przygotowanie odpowiedniego narzędzia niezależnie od wykorzystywanego środowiska czy omawianego problemu może czasem okazać się zadaniem zajmującym więcej czasu niż samo rozwiązanie

problemu. Głównymi trudnościami jakie można napotkać na swej drodze są problemy z czytelną reprezentacją wyników czy zaprojektowanie prostego w obsłudze interfejsu użytkownika.

Pomysł rozwiązania problemu powstał i został rozwinięty podczas badań nad implementacją skierowanych liczb rozmytych w algorytmach rojowych. Interfejs został przygotowany w środowisku R ze względu na początkową implementację algorytmów rojowych aby zachować spójność badań i wystrzec się konieczności przepisywania aktualnie istniejących implementacji oraz aby zachować główną zaletę języka skryptowego R którą jest prędkość kompilacji oraz łatwość eliminacji błędów. Głównymi bibliotekami wykorzystywanymi podczas

przygotowywania omawianego środowiska są moduły Shiny, Shiny Dashboard oraz DT. Pozostałe omawiane moduły to biblioteki takie jak: ShinyCSSLoaders, ShinyJS oraz wiele innych modułów poprawiających czytelność i odbiór wyników pracy.

Przedstawione zrzuty ekranu narzędzia są przykładowym wykorzystaniem środowiska testowego przygotowanego właśnie z użyciem omówionych bibliotek dostępnych w środowisku R.

## 2. Środowisko R

Środowisko R jest bardzo często wykorzystywane do obliczeń statystycznych i ich wizualizacji. Głównymi powodami szerokiego zainteresowania omawianego języka jest dostępność (środowisko R jest dostępne jako Open Source na warunkach licencji GNU (*General Public License*)), szybkość działania oraz elastyczność dostępnych narzędzi. Te główne zalety spowodowały że wiele firm zdecydowało się na wykorzystanie komercyjnego środowiska R. Najlepszymi przykładami są firmy takie jak Facebook - R używany do analizy zachowań związanych ze zmianami statusu, Google - do mierzenia efektywności reklamy oraz przewidywań ekonomicznych oraz wiele innych korporacji takich jak Twitter, Microsoft, Uber, Airbnb, IBM. Jednak zanim język R znalazł swoje zastosowanie w świecie modeli biznesowych swoje początki język przeżył na uczelniach na całym świecie i tam króluje do dnia dzisiejszego. Został zaadaptowany na wiele różnych platform Unix oraz Windows i MacOS. R to nie tylko język programowania, to całe zintegrowane środowisko, zawierające oprogramowanie do wykonywania operacji na danych oraz reprezentacji wyników. R to optymalnie zaprojektowany i spójny system przygotowany do operacji na zbiorach danych. Zastosowanie R w projektach pozwala na ominięcie problemów związanych z innymi nieelastycznymi programami do operacjach nad danymi, problemów jak limitacje odczytu danych do wąskiego typu danych czy problem z brakiem narzędzi do reprezentacji wyników.

## 3. Biblioteka Shiny

Shiny to darmowa biblioteka przygotowana przez znanych w kręgu twórców oprogramowania RStudio. Wykorzystując omawiany moduł w bardzo prosty i przyjazny sposób można przygotować dynamiczne strony internetowe wykorzystując tylko język R bez konieczności nauki kolejnych języków programowania jak JavaScript czy języka znaczników HTML. Stworzone aplikacje z wykorzystaniem moduły Shiny są generowane dynamicznie co pozwala na zmianę treści strony

podczas pracy skryptu. Omawiana biblioteka nie ogranicza i pozwala jednak na korzystanie z technologii takich jak HTML, CSS czy JavaScript. Moduł Shiny jest w pełni zgodny ze środowiskiem R, co pozwala na jego używanie na każdej maszynie obsługującej środowisko R. Dzięki zastosowaniu przez twórców, biblioteki HTTPUV która dostarcza komunikację dwukierunkową między serwerem a użytkownikiem, interakcja między stworzoną przez nas stroną a naszymi skryptami dokonywana jest natychmiastowo, a całą zaawansowana implementacja komunikacji obsługiwana jest właśnie przez moduł Shiny.

Poniżej przedstawiony został prosty przykład jak przygotować prostą aplikację z użyciem biblioteki Shiny. Dobrą metodyką jest stworzenie dwóch osobnych plików UI.R oraz Server.R, lecz celem ukazania łatwości przygotowania aplikacji przykłady kodu aplikacji przygotowanej będą w jednym pliku App.R.

```
#Komenda ładująca
#Shiny do naszego skryptu.
library(shiny)

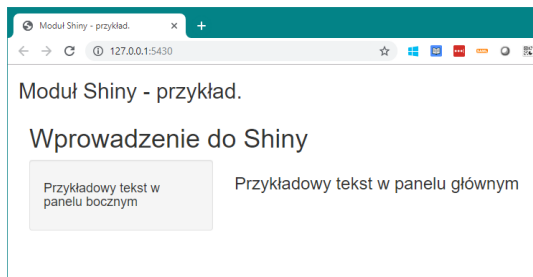
#Zdefiniowanie strony
#UI dla naszego programu.
ui <- fluidPage(
  #Nadanie tytułu naszej stronie.
  titlePanel("Moduł Shiny - przykład."),

  #Nazwa nazwy panelu głównego.
  headerPanel("Wprowadzenie do Shiny."),

  #Zdefiniowanie panelu bocznego.
  sidebarPanel(
    #Przykładowy nagłówek w panelu bocznym.
    h4("Przykładowy tekst w panelu bocznym.")
  ),
  #Zdefiniowanie panelu głównego.
  mainPanel(
    h3("Przykładowy tekst w panelu głównym.")
  )
)
#Logikę naszego programu zawieramy
#w części server naszej aplikacji.
server <- function(input, output){
}

#Zdefiniowanie głównych funkcji
#dla biblioteki Shiny.
shinyApp(ui = ui, server = server)
```

**Rysunek. 1** Przykładowy kod prostej aplikacji z wykorzystaniem modułu Shiny wraz z komentarzem. Jest to cały plik app.R. Źródło: własne.



**Rysunek. 2** Reprezentacja aplikacji z powyżej załączonego kodu. Źródło: własne.

Jak widać na załączonych rysunkach przygotowanie aplikacji internetowej używając biblioteki Shiny jest bardzo szybkie i proste. Dodanie kolejnych elementów to kwestia pojedynczych linijek kodu.

#### 4. Biblioteka Shiny Dashboard

Użycie biblioteki Shiny pozwala przy niskim nakładzie pracy na osiągnięcie bardzo ładnych lecz prostych interfejsów. W tym miejscu na pomoc przychodzi kolejna biblioteka rozbudowująca możliwości pakietu Shiny.

Moduł Shiny Dashboard jest pakietem pozwalającym w bardzo szybki i przy małym skomplikowaniu stworzyć zaawansowane widoki dla naszej aplikacji. Shinydashboard to pakiet R, którego zadaniem jest ułatwienie (jak sama nazwa wskazuje) budowania przejrzystych dashboardów (z ang. kokpitów) z Shiny.

```
library(shiny)
#Załadowanie modułu ShinyDashboard
library(shinydashboard)
#Konfiguracja DashboardPage jako domyślnego widoku.
ui <- dashboardPage(
  #Załadowanie domyślnych modułów.
  dashboardHeader(),
  dashboardSidebar(),
  dashboardBody()
)
server <- function(input, output) { }
shinyApp(ui, server)
```

**Rysunek. 3** Przykładowy kod prostej aplikacji z wykorzystaniem modułu Shiny oraz Shiny Dashboard wraz z komentarzem. Jest to cały plik app.R. Źródło: własne.

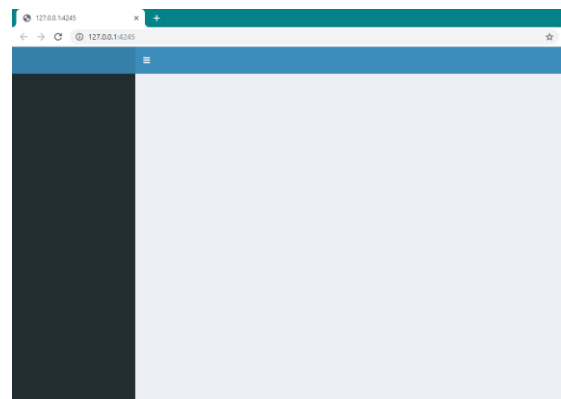
Część interfejsu użytkownika w aplikacji zbudowanej z Shiny Dashboard zawiera 3 podstawowe elementy zamknięte w dashboardPage:

1. dashboardHeader - elementy dostępne w nagłówku strony;

2. dashboardSidebar - elementy dostępne w panelu po lewej (domyślnie) stronie;
3. dashboardBody - elementy dostępne w głównym panelu aplikacji.

Aplikacja przygotowana z pomocą Shiny Dashboard jest bardzo ładna a przy tym zachowuje kompatybilność z elementami modułu Shiny w pełni co pozwala na bardzo efektowne i proste budowanie zaawansowanych aplikacji.

Przygotowanie aplikacji z wykorzystaniem opisanego modułu jest jeszcze prostsze a efekty jeszcze lepsze. Ponieważ korzystamy z wielu gotowych, przygotowanych szablonów. I tym sposobem w 8 liniach kodu możliwe jest przygotowanie ładnie wyglądającego szablonu do naszej aplikacji.

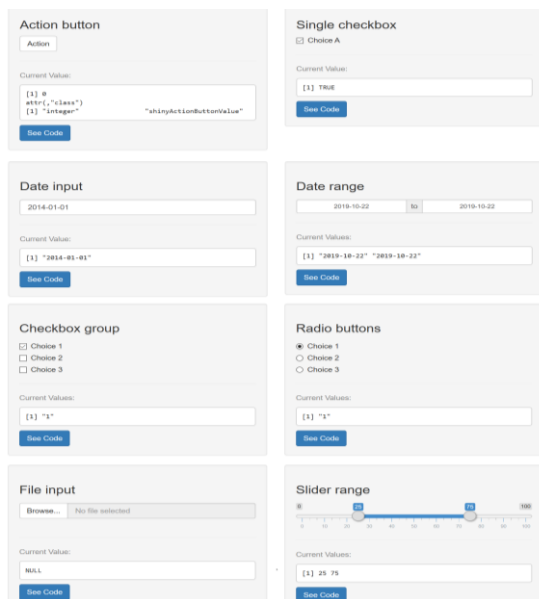


**Rysunek. 4** Reprezentacja aplikacji z powyżej załączonego kodu z wykorzystaniem Shiny Dashboard. Źródło: własne.

Możliwość korzystania z elementów biblioteki Shiny jest w tym przypadku kluczowa, ponieważ dzięki połączeniu możliwości dwóch wymienionych bibliotek otrzymujemy narzędzie, które pozwoli nam zaprezentować wyniki naszej pracy w sposób bardzo profesjonalny w bardzo krótkim czasie. Jest to funkcjonalność, która w innych językach programowania wiązałaby się ze skomplikowaniem projektu, utratą kompatybilności czy innego rodzajami ograniczeń. Pakiet Shiny dla wielu jest głównym powodem wyboru R ponad język Python. Python pomimo łatwiejszej struktury kodu nie oferuje w tak prosty sposób reprezentacji wyników w tak szybki sposób. To w wielu przypadkach dyskwalifikuje Python podczas wyboru z zastosowań przy projektach bazujących na przetwarzaniu danych.

## 5. Widżety a możliwości

Shiny oferuje szereg dostępnych elementów graficznych do reprezentacji danych nazywanych 'widżetami'. To właśnie widżety grają pierwsze skrzypce podczas budowania interfejsu. Widżety umożliwiają użytkownikom wysyłanie informacji do aplikacji. Widżety jak cała nasza aplikacja są dynamiczne oznacza to, że gdy użytkownik zmieni wartość na widżecie, wartość również się zmieni po stronie serwerowej naszej aplikacji. Pozwala to na dynamiczne zmiany konfiguracji naszych aplikacji oraz na bieżącą interakcję z naszą aplikacją, bez potrzeby przeładowywania strony.



**Rysunek. 5** Przykłady niektórych widżetów dostępnych w Shiny. Źródło: <https://shiny.rstudio.com/gallery/widget-gallery.html>

Jednak nie musimy się ograniczać do ogromnej biblioteki widżetów Shiny, każda biblioteka R bazująca na HTML i JavaScript może zostać zaimplementowana w naszą aplikację. A takich modułów dodatkowych jest bardzo wiele.

Na chwilę uwagi zasługuje tu wykorzystywana wielokrotnie przeze mnie biblioteka DT (skrót od *DataTables*). Moduł poza skromną nazwą nie reprezentują sobą nic skromnego. To przeogromne narzędzie pozwalające na wyświetlanie i pracę na tabelach. Przykładowe wykorzystanie tej tabeli znajdziemy w dokumentacji Shiny pod adresem: <https://gallery.shinyapps.io/012-datatables/>. Biblioteka DT poza podstawową obsługą wyświetlania danych pozwala również na ich segregację, paginację itp. co jest niezbędne przy pracy z wynikami w formie tabeli.

Showing 10 entries

Search:

|    | carat | cut       | color | clarity | depth | table | price | x    | y    | z    |
|----|-------|-----------|-------|---------|-------|-------|-------|------|------|------|
| 1  | 0.52  | Ideal     | D     | VS2     | 61.4  | 56    | 1664  | 5.16 | 5.19 | 3.18 |
| 2  | 0.5   | Very Good | F     | SI1     | 62.3  | 60    | 1250  | 5.07 | 5.11 | 3.17 |
| 3  | 0.61  | Ideal     | G     | VVS2    | 61.6  | 54    | 2242  | 5.45 | 5.49 | 3.37 |
| 4  | 0.36  | Premium   | G     | VS2     | 62.5  | 58    | 756   | 4.55 | 4.51 | 2.83 |
| 5  | 0.7   | Very Good | E     | VS2     | 63.5  | 54    | 2889  | 5.62 | 5.66 | 3.58 |
| 6  | 0.56  | Ideal     | F     | VS1     | 61.7  | 56    | 2016  | 5.32 | 5.28 | 3.27 |
| 7  | 1.19  | Premium   | E     | I1      | 60.2  | 61    | 3572  | 6.91 | 6.87 | 4.15 |
| 8  | 0.52  | Ideal     | F     | IF      | 60.6  | 57    | 2575  | 5.21 | 5.22 | 3.16 |
| 9  | 0.38  | Ideal     | E     | IF      | 62.7  | 55    | 1433  | 4.61 | 4.67 | 2.91 |
| 10 | 0.51  | Ideal     | E     | VS2     | 62.1  | 54    | 1608  | 5.13 | 5.15 | 3.19 |

Showing 1 to 10 of 1,000 entries

Previous 1 2 3 4 5 ... 100 Next

**Rysunek. 6** Przykłady wykorzystania biblioteki DT. Źródło: własne.

O modułach dostępnych w R oraz współpracujących z biblioteką Shiny można pisać bez końca, a lista dostępnych ciągle rośnie z dnia na dzień. Z wartych wspomnienia warto zapoznać się z modułami takimi jak:

- ShinySense - Jest to to pakiet modułów, które pomagają Shiny obsłużyć moduły sprzętowe (m.in. nagrywać obraz z kamery internetowej, nagrywać dźwięk, przechwytywać dane akcelerometru).
- ShinyFiles - Przeglądarka systemu plików po stronie serwera dla Shiny.
- RegexSelect - Pozwala na wyszukiwanie wyrażeń regularnych w naszym zaznaczeniu. Wielokrotnie używana biblioteka do wyszukania szczegółowych danych.
- Waiter - Ekrany ładowania, niezbędne podczas pracy nad większymi zbiorami danych.
- Shinytoastr - Powiadomienia z aplikacji w formie pop up.
- Rintrojs - Moduł do tworzenia prezentacji krok po kroku i klikalnych wskazówek.
- plotly - Interaktywny generator wykresów 2D/3D. Wspiera specjalne wsparcie dla łączenia / podświetlania / filtrowania widoków.

oraz wiele innych bardzo ciekawych, wartych odkrycia bibliotek. Możliwości R są tak szerokie jak wybór dostępnych bibliotek, co może wydawać się nieskończone.

## 6. Budowa środowiska testowego

Celem projektu było zaprojektowanie i przygotowanie narzędzia, które pozwoliłoby testować nowo zaimplementowane algorytmy rojowe również z wykorzystaniem skierowanych liczb rozmytych pod kątem optymalizacji. Głównymi wyznacznikami podczas implementacji

rozwiązania była otwartość na rozwój, duża baza dostępnych algorytmów rojowych, brak ograniczeń implementacji nowych ustawień oraz możliwość przyjaznej reprezentacji danych. Największym wyzwaniem było przygotowanie narzędzia w sposób prosty do użytkowania, tak aby reprezentacja wyników była w przyjaznej użytkownikowi formie. Co miało na celu znacznie zachęcić do wykorzystywania narzędzia a tym samym zwiększyć zainteresowanie algorytmami rojowymi.

Środowisko testowe zostało przygotowane w sposób modularny tak aby można było je w przyszłości rozwijać w prosty sposób. Składa się z komponentów uruchamiających aplikację, komponentów interfejsu użytkownika, komponentów odpowiedzialnych za logikę środowiska testowego oraz plików serwowanych statycznie.

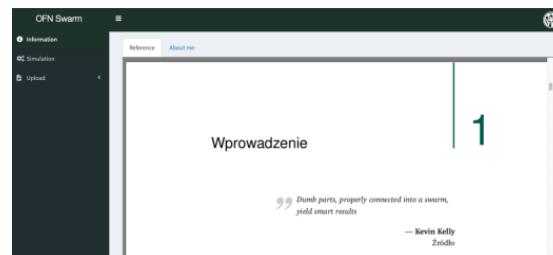
Komponent uruchamiający aplikację zawiera spis wszystkich ustawień globalnych. W tym komponencie również załadowane są wszystkie wykorzystywane przez nasz program biblioteki.

```
library(shiny)
library(shinydashboard)
library(shinycssloaders)
library(shinyjs)
library(dt)
library(waiter)
library(plotly)
```

**Rysunek. 7** Zrzut ekranu bibliotek ładowanych z głównym komponencie środowiska testowego. Źródło: własne.

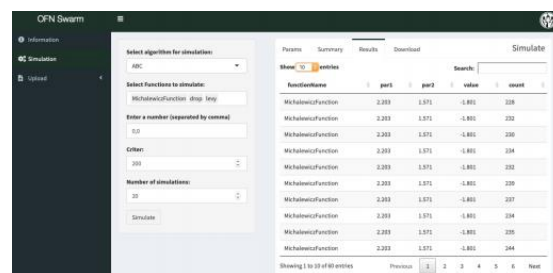
Moduły interfejsu użytkownika oraz logiki naszego środowiska testowego również zostały przygotowane tak, aby ich rozwój był prosty i dostępne nawet dla początkującego użytkownika. Każda dodana funkcja to dodanie nowego pliku w odpowiedniej lokalizacji oraz załadowanie w komponencie uruchamiającym.

W pierwszej zakładce, na stronie powitalnej umieszczona została dokumentacja projektu oraz krótkie wprowadzenie do algorytmów rojowych oraz problemów optymalizacji. Zostało to zrobione dzięki możliwości implementacji w języku R funkcji Markdown i połączeniu dokumentacji przygotowanej w języku LaTeX za jej pomocą z naszym środowiskiem.



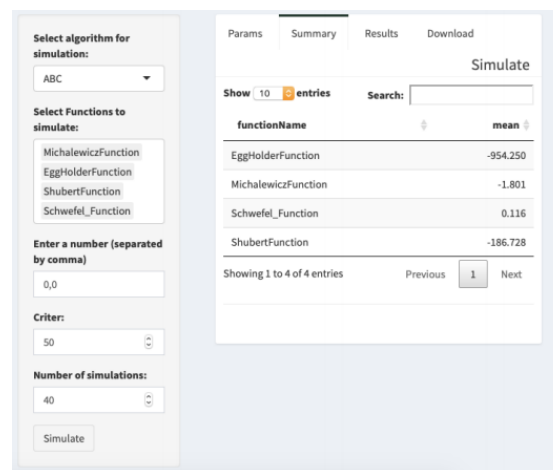
**Rysunek. 8** Zrzut ekranu pierwszej strony oprogramowania.

Drugi odnośnik w menu z prawej strony przenosi nas do zakładki testowania algorytmów na wybranych funkcjach matematycznych z wykorzystaniem wybranych algorytmów. Interfejs został przygotowany w taki sposób aby umożliwić testowanie algorytmów rojowych zarówno z wykorzystaniem skierowanych liczb rozmytych jak i w implementacji klasycznej, pozwala również na testowanie wielu algorytmów jednocześnie.



**Rysunek. 9** Zrzut ekranu strony testowania w środowisku testowym. Źródło: własne.

Przygotowanie narzędzia w taki sposób pozwala na znaczne skrócenie czasu testowania algorytmów rojowych pod kątem optymalizacji oraz na szybkie i dokładne porównywanie wyników. Wynik jest reprezentowany za pomocą biblioteki DT oraz widgetu zakładek.



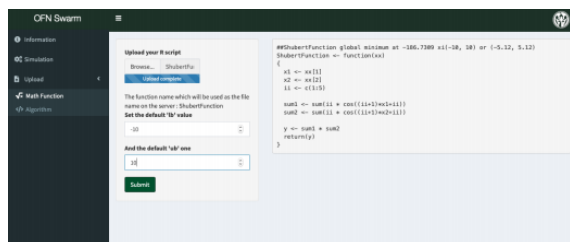
**Rysunek 1.** Wykorzystane widgety.

Zakładki pozwalają na wybór pomiędzy podstronami z wynikiem szczegółowym, podsumowaniem, wygenerowanym grafem oraz możliwością pobrania wyników w formacie CSV lub PDF. Co pozwala na późniejszą dalszą obróbkę danych lub przygotowanie obszernego raportu.

| functionName        | Z OFN        | Bez OFN      | U źródła  |
|---------------------|--------------|--------------|-----------|
| EggHolderFunction   | -958.1018285 | -954.2496102 | -959.6507 |
| MichalewiczFunction | -1.801303408 | -1.80130341  | -1.8013   |
| Schwefel_Function   | 0.109273939  | 0.11634984   | 0         |
| ShubertFunction     | -186.7301957 | -186.7284601 | -186.7309 |

**Tabela. 1** Przykład tabeli wyników wyeksportowanych do CSV. Źródło: własne.

Ostatnie dwie zakładki zostały dodane aby ułatwić rozwój narzędzia użytkownikowi końcowemu. Pozwalają na dodanie nowych algorytmów rojowych czy to z wykorzystaniem skierowanych liczb rozmytych czy też nie, pozwalają również na dodanie kolejnych nowych testowych funkcji matematycznych.



**Rysunek. 12** Zrzut ekranu strony dodawania funkcji matematycznych. Źródło: własne.

Dzięki zastosowaniu takiej metody dodawania nowych algorytmów czy funkcji matematycznych nie ma konieczności ingerowania w pliki środowiska testowego. Cała praca operacyjna podczas testów może zostać wykonana z punktu widzenia interfejsu użytkownika. Pozwala to na wykorzystanie chmury czy zasobów sieciowych do przeprowadzania badań, przez co zasoby do testowania wcale nie muszą się ograniczać do zasobów naszego komputera. Twórcy Shiny R, poza przygotowaniem narzędzia idealnego do szybkiego tworzenia łatwo dostępnych interfejsów dla naszego oprogramowania w R przygotowali również prosty i skuteczny sposób publikacji naszego oprogramowania w formie strony internetowej dostarczając nam kolejny moduł

biblioteki o nazwie Shiny Server. Właśnie dzięki połączeniu dobrych praktyk podczas projektowania narzędzia oraz dostępnych udogodnień przygotowanych przez społeczność języka R. Nasze środowisko testowe może być wykorzystane w nieskończenie wiele sposobów, dla różnych scenariuszy. Bez zmartwień o brak zasobów czy ograniczenia z punktu widzenia modyfikacji interfejsu.

## 7. Podsumowanie i wnioski

Dzięki zastosowaniu dostępnych, omówionych bibliotek języka R oraz narzędzi dostarczonych przez twórców R Studio takich jak Shiny R Server. Środowisko testowe przygotowane w środowisku R spełnia w pełni swoje założenia jako oprogramowanie, które można wielokrotnie wykorzystać i dalej rozwijać. Z przebiegu pracy jednoznacznie wynika że przygotowanie narzędzia bazując na języku R i dostępnych bibliotekach znacznie ułatwia przygotowanie środowisk testowych oraz ogólne przeprowadzenie badań przy problemach optymalizacyjnych a implementacja dodatkowych algorytmów czy funkcji matematycznych jest bardzo prosta.

## Literatura

1. Matthew Campbell. „Shiny R Dashboards“. W: kw. 2019.
2. Leila Etaati. „Introduction to R“. W: czer. 2019
3. Beeley, Chris. Web application development with R using Shiny. Packt Publishing Ltd, 2013.
4. Allaire, J. "RStudio: integrated development environment for R." Boston, MA 770 2012.
5. Beeley, Chris, and Shitalkumar R. Sukhdeve. Web Application Development with R Using Shiny: Build stunning graphics and interactive data visualizations to deliver cutting-edge analytics. Packt Publishing Ltd, 2018.
6. Chang, Winston, and Barbara Borges Ribeiro. "Shinydashboard: create dashboards with 'Shiny'." R package version 0.7. 2018.