Studia i Materiały Informatyki Stosowanej

Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników naukowych, doktorantów i studentów

patronat: Polskie Towarzystwo Informatyczne

Studies and Materials in Applied Computer Science

Studies and Materials in Applied Computer Science

Journal of young researchers, PhD students and students

Endorsed by Polish Information Processing Society

Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników naukowych, doktorantów i studentów

Tom 2,Nr 2,2010

Bydgoszcz 2010

Studia i Materiały Informatyki Stosowanej
czasopismo młodych pracowników naukowych, doktorantów i studentów

ISSN 1689-6300
ISBN 978-83-925180-8-2

Projekt okładki: Łukasz Zawadzki
DTP: Sebastian Szczepański

Wydawca:

Wydział Matematyki, Fizyki i Techniki
Uniwersytet Kazimierza Wielkiego
ul. Chodkiewicza 30
85-064 Bydgoszcz
tel. (052) 34-19-331
fax. (052) 34-01-978
e-mail: simis@ukw.edu.pl

Kontakt:

dr inż. Jacek Czerniak
dr inż. Marek Macko
Uniwersytet Kazimierza Wielkiego
ul. Chodkiewicza 30
85-064 Bydgoszcz
e-mail: jczerniak@ukw.edu.pl
mackomar@ukw.edu.pl

Druk (ze środków sponsora):
Oficyna Wydawnicza MW

Nakład 510 egz.

Bydgoszcz 2010

SPIS TREŚCI

CONTENTS

Bardzo Was tu brakuje...

10 kwietnia 2010 roku, w drodze na uroczystości 70. rocznicy Zbrodni Katyńskiej zginął Prezydent Rzeczypospolitej Polskiej Lech Kaczyński, Małżonka Prezydenta Maria Kaczyńska oraz członkowie polskiej delegacji i załoga samolotu. Wieczny odpoczynek racz Im dać Panie!

Lista pasażerów rządowego samolotu

1. Lech Kaczyński
2. Maria Kaczyńska
3. Ryszard Kaczorowski
4. Joanna Agacka-Indecka
5. Ewa Bąkowska
6. gen. Andrzej Błasik
7. Krystyna Bochenek
8. Anna Maria Borowska
9. Bartosz Borowski
10. gen. Tadeusz Buk
11. abp gen. Miron Chodakowski
12. Czesław Cywiński
13. Leszek Deptuła
14. ppłk Zbigniew Dębski
15. Grzegorz Dolniak
16. Katarzyna Doraczyńska
17. Edward Duchnowski
18. Aleksander Fedorowicz
19. Janina Fetlińska
20. Jarosław Florczak
21. Artur Francuz
22. gen. Franciszek Gągor
23. Grażyna Gęsicka
24. gen. Kazimierz Gilarski
25. Przemysław Gosiewski
26. ks. Bronisław Gostomski
27. Mariusz Handzlik
28. ks. Roman Indrzejczyk
29. Paweł Janeczek
30. Dariusz Jankowski
31. Izabela Jaruga-Nowacka
32. ks. Józef Joniec
33. Sebastian Karpiniuk
34. wiceadm. Andrzej Karweta
35. Mariusz Kazana
36. Janusz Kochanowski
37. gen. Stanisław Komornicki
38. Stanisław Jerzy Komorowski
39. Paweł Krajewski
40. Andrzej Kremer
41. ks. Zdzisław Król
42. Janusz Krupski
43. Janusz Kurtyka
44. ks. Andrzej Kwaśnik
45. gen. Bronisław Kwiatkowski
46. Wojciech Lubiński
47. Tadeusz Lutoborski
48. Barbara Mamińska
49. Zenona Mamontowicz-Łojek
50. Stefan Melak
51. Tomasz Merta
52. Stanisław Mikke
53. Aleksandra Natalli-Świat
54. Janina Natusiewicz-Mirer
55. Piotr Nosek
56. Piotr Nurowski
57. Bronisława Orawiec-Löffler
58. ks. ppłk Jan Osiński
59. ks. płk Adam Pilch
60. Katarzyna Piskorska
61. Maciej Płażyński
62. bp gen. Tadeusz Płoski
63. gen. Włodzimierz Potasiński
64. Andrzej Przewoźnik
65. Krzysztof Putra
66. ks. Ryszard Rumianek
67. Arkadiusz Rybicki
68. Andrzej Sariusz-Skąpski
69. Wojciech Seweryn
70. Sławomir Skrzypek
71. Leszek Solski
72. Władysław Stasiak
73. Jacek Surówka
74. Aleksander Szczygło
75. Jerzy Szmajdziński
76. Jolanta Szymanek-Deresz
77. Izabela Tomaszewska
78. Marek Uleryk
79. Anna Walentynowicz
80. Teresa Walewska-Przyjałkowska
81. Zbigniew Wasserman
82. Wiesław Woda
83. Edward Wojtas
84. Paweł Wypych
85. Stanisław Zając
86. Janusz Zakrzeński
87. Gabriela Zych
88. Dariusz Michałowski

"Niech tuli Was do snu po brzask podmorskich pieśń połonin.
Niech dobry Bóg sto morskich dróg do Aylo Wam prostuje.
Niech nasze myśli do Was mkną, bardzo Was tu brakuje,
Bardzo Was tu brakuje,
Bardzo Was tu ..."

Requiem dla nieznajomych przyjaciół z s/y Bieszczady
Wykonawcy:Banana Boat
Autor tekstu: Pawła Jędrzejko

Lista członków załogi

1. Arkadiusz Protasiuk
2. Robert Grzywna
3. Andrzej Michalak
4. Artur Ziętek
5. Barbara Maciejczyk
6. Natalia Januszko
7. Justyna Moniuszko
8. Agnieszka Pogródka-Więcławek

SŁOWO WSTĘPNE

Szanowni Czytelnicy,

skład obecnego numeru czasopisma Studia i Materiały Informatyki Stosowanej zbiegł się z katastrofą w Smoleńsku. Postanowiliśmy więc zrezygnować z typowej dla SiMIS kolorystyki aby chociaż w takim drobny sposób przyłączyć się do poczucia żałoby jakie nam wszystkim towarzyszyło. Następujące po katastrofie niezwykłe zjawiska atmosferyczne i te natury wulkanicznej i te hydrologiczne dopełniają nastrój tego półrocza zmuszając nas do refleksji bardziej metafizycznej i egzystencjalnej.
Ten nastrój będzie zatem w aktualnym numerze SiMIS musiał zastąpić kolejny artykuł z cyklu „Spotkania z Mistrzami", kolejne przemyślenia profesorskie znajdą Państwo w następnych numerach w tak nadsyłania ich przez zaproszonych autorów.
Dziękujemy gorąco wszystkim, którzy włączyli się samorzutnie do promowania SiMIS wśród swoich środowisk, szczególne podziękowania kierujemy do Członków Rady Naukowej, dostajemy z różnych stron kraju wiadomości, że informacje o SiMIS pojawiają się w gablotach i podczas spotkań z PT zainteresowanymi autorami. Szczególne podziękowania za aktywność w tym numerze składamy dla środowiska doktorantów UT AGH. Bardzo nam miło gościć Państwa na naszych łamach. Licznie reprezentowani są też doktoranci z WI ZUT (dawniej WI PS) oraz pojedyncze osoby z innych miejsc Polski. Na Państewa ręce składamy też podziękowania Waszym Dziekanom, Kierownikom Studiów Doktoranckich i wszelkim przełożonym za wsparcie i motywację.

Redaktorzy Naczelni SiMIS,
Dr inż. Jacek Czerniak,
Dr inż. Marek Macko

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str.15-22

NADESŁANE ARTYKUŁY

PRZEGLĄD PODSTAWOWYCH METOD REPREZENTACJI KSZTAŁTÓW 3D

Dariusz Frejlichowski, Patrycja Nuszkiewicz

Zachodniopomorski Uniwersytet Technologiczny
Wydział Informatyki
ul. Żołnierska 49, 71-210 Szczecin
e-mail: dfrejlichowski@wi.zut.edu.pl, pnuszkiewicz@wi.zut.edu.pl

Streszczenie: *W artykule przedstawiono przegląd istniejących rozwiązań w dziedzinie reprezentowania kształtów trójwymiarowych, stosunkowo nowym podejściu do opisywania obiektów w przetwarzaniu, rozpoznawaniu i indeksowaniu obrazów. Deskryptory kształtu 3D stają się coraz potrzebniejsze i coraz powszechniej stosowane. Wynika to z rozwoju sprzętu komputerowego, a co za tym idzie możliwości szybkiego przetwarzania skomplikowanych scen trójwymiarowych. Znajduje się przy tym kolejne obszary zastosowań trójwymiarowego opisu obiektów, m.in. w biometrii, systemach CAD i indeksowaniu.*

Słowa kluczowe: *Kształt 3D, indeksowanie obrazów, deskryptory kształtu 3D*

A discussion on selected problems of lossless binary digital images compression

Abstarct: *In the paper a brief survey on existing approaches to the representation of three-dimensional shapes was presented. It is a new way of describing objects in image processing, recognition and indexing. The 3D shape descriptors become more and more useful and widely used. Rapid development of computer hardware and the possibilities of fast processing of three-dimensional scenes cause it. Many new applications of 3D shape description can be easily found, e.g. in biometrics, CAD systems and image retrieval.*

Keywords: *3D shape, image retrieval, 3D shape descriptors*

1. WPROWADZENIE

W ostatnich latach odnotować można wzrost zainteresowania aplikacjami wykorzystującymi obiekty trójwymiarowe. Sprzyjały temu znaczne ulepszenia narzędzi służących do tworzenia modeli 3D, skanerów trójwymiarowych oraz wzrost przepływu informacji, głównie w Internecie. Ponadto sprzęt komputerowy jest obecnie wystarczająco szybki i tani, aby dane trójwymiarowe mogły zostać przetworzone i wyświetlone w czasie rzeczywistym. Rozwój grafiki trójwymiarowej widoczny jest już nie tylko w ramach informatyki, ale także i w innych dziedzinach, takich jak: medycyna, biologia molekularna, czy inżynieria mechaniczna. Ze względu na dużą złożoność modeli 3D, jak również ogromną ich ilość w różnego rodzaju bazach, pojawił się problem ich szybkiego i skutecznego wyszukiwania, rozpoznawania i klasyfikowania. Wybrany do tego celu algorytm powinien efektywnie odróżniać cechy charakterystyczne dla różnych klas kształtów trójwymiarowych oraz znajdować modele najbardziej podobne do zadanego na wejściu systemu. Wymagany jest również optymalny czas działania rozwiązania, tak aby wyniki wyszukiwania lub rozpoznawania przedstawiane były w rozsądnie krótkim czasie.

W niniejszym artykule opisane zostały najważniejsze używane dotychczas podejścia do problemu reprezentacji kształtów 3D. Stosowanie deskryptorów tego typu

pozwala na wydobycie najistotniejszych informacji o obiektach, redukując ilość potrzebnych danych (nie musimy używać dzięki temu całych modeli trójwymiarowych), ale także pozwala na uniezależnienie się od różnego rodzaju deformacji porównywanych obiektów.

Kształty trójwymiarowe posiadają cechy, które znacząco odróżniają je od obrazów dwuwymiarowych, w kontekście rozwiązania określonego problemu, np. efektywnego rozpoznawania ([12], [13]). Zazwyczaj, w przeciwieństwie do modeli 2D, obiekty 3D nie są zależne od położenia kamer, źródeł światła lub otaczających obiektów na scenie. W efekcie, obiekty nie zawierają odbić, cieni lub części innych obiektów. Ułatwia to proces wyznaczania miary podobieństwa pomiędzy różnymi modelami. Z drugiej strony, zauważalny jest dużo większy rozmiar opisu obiektów trójwymiarowych, co utrudnia wyszukiwanie odpowiadających sobie własności oraz parametrów modeli. Ponadto, większość obiektów, znajdujących się w dużych bazach danych, zawiera źle zorientowane, niepołączone lub brakujące wielokąty. Poprawienie jakości takich modeli jest trudne w realizacji, gdyż bardzo często wymaga ingerencji człowieka. Dlatego jednym z ważniejszych cech stawianych efektywnemu deskryptorowi kształtu trójwymiarowego jest odporność na błędy, wynikające z uszkodzenia lub braku wielokątów.

Ze względu na wspólne, charakterystyczne cechy prezentowane w artykule podejścia zostały podzielone na cztery grupy: deskryptory geometryczne, strukturalne, symetryczne oraz lokalne. W publikacji pominięto problem porównywania otrzymanych reprezentacji, koncentrując się na samym procesie ich tworzenia.

2. CHARAKTERYSTYKA WYBRANYCH METOD OPISU KSZTAŁTU OBIEKTÓW TRÓJWYMIAROWYCH

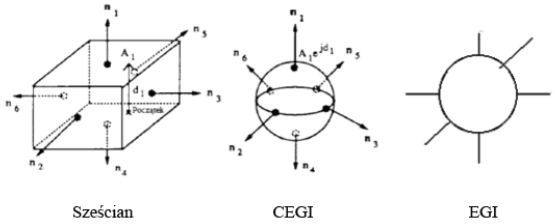
2.1. Deskryptory geometryczne

EGI, CEGI

Pierwszym deskryptorem kształtu, który w sposób znaczący wpłynął na późniejsze badania, był opisany w [5], pochodzący z roku 1983, deskryptor EGI (*Extended Gaussian Image*). Istotną wadą tej reprezentacji jest brak odporności na skalowanie i obrót – deskryptor zostaje tak samo przekształcony, jak zmodyfikowany obiekt. Natomiast zaletą tego podejścia jest niezależność od położenia obiektu na scenie, czyli jego przesunięcia (translacji) w przestrzeni. Dzięki tym właściwościom deskryptor EGI znalazł liczne zastosowania, m.in. w systemach rozpoznawania, wyznaczania orientacji obiektów 3D w przestrzeni, podziale map głębokości na mniejsze obiekty. Warto przy tym wspomnieć, że istnieją algorytmy pozwalające na otrzymanie źródłowego obiektu na bazie jego deskryptora EGI. Cecha odwracalności deskryptora jest rzadko spotykaną właściwością w istniejących rozwiązaniach, dlatego jej spełnienie przez omawiane podejście na pewno należy uznać za istotną zaletę. Budowa deskryptora EGI oparta jest na obrazie gaussowskim, który otrzymujemy poprzez mapowanie powierzchni wektorów normalnych obiektu na sferę jednostkową (sferę gaussowską). Wektory normalne zostają umieszczone na sferze w taki sposób, aby punkt początkowy wektora znajdował się w punkcie środkowym sfery, a punkt końcowy – w punkcie na sferze, odpowiadającym orientacji danej powierzchni (ściany obiektu) w przestrzeni. Deskryptor EGI wzbogaca obraz gaussowski o informacje o rozmiarze powierzchni wszystkich ścian obiektu. Oznacza to, że dla każdego punktu na sferze gaussowskiej ustalona zostaje waga, równa powierzchni ściany obiektu. Wyznaczone wagi przedstawione są jako wektory równoległe do wektorów normalnych (danej powierzchni), o długości równej wartości obliczonej wagi.

Najważniejszą wadą deskryptora EGI jest brak informacji o położeniu ścian obiektu (np. nie można odczytać, które ściany obiektu są ze sobą połączone). Cecha ta powoduje niejednoznaczność – tylko w przypadku wielościanów wypukłych deskryptor jest unikalny. Powstało wiele propozycji rozwiązania powyższego problemu, np. algorytm opisany w [7] i [14], zachowujący informację o położeniu powierzchni poprzez sformułowanie równania ściany obiektu w podwójnej przestrzeni (jednoczesnej reprezentacji orientacji i położenia ściany obiektu).

Kolejny deskryptor, CEGI *(Complex Extended Gaussian Image*), jest rozszerzeniem poprzednio opisanego, przechowującym informacje o orientacji, powierzchni ścian oraz pozycji obiektu w przestrzeni. W przypadku deskryptora EGI wielkość wagi, związanej z wektorem normalnym ściany modelu 3D, równa jest powierzchni określonej ściany. Aby powstał deskryptor CEGI należy dodać do każdej wagi część urojoną – odległość pomiędzy powierzchnią ściany, a określonym początkiem (zwróconą w kierunku wektora normalnego danej ściany). Oznacza to, że waga (związana z określonym wektorem normalnym) jest liczbą zespoloną, której część rzeczywista równa jest powierzchni ściany, a urojona jest odległością mierzoną od określonego początku do powierzchni ściany obiektu. Na rys.1. przedstawiono graficzne interpretacje deskryptorów CEGI i EGI dla prostego obiektu 3D.



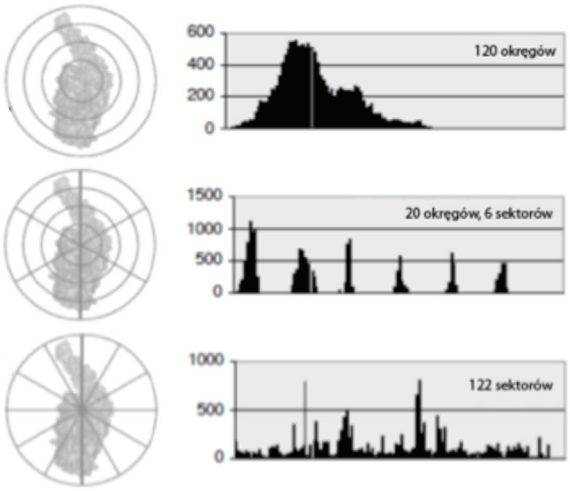
Rysunek. 1 Przykładowe interpretascje deskryptorów EGI i CEGI dla sześcianu, źródło: [18], [19].

W przypadku, gdy obiekt jest wypukły, deskryptor CEGI równy jest deskryptorowi EGI. Wartość wag deskryptora CEGI jest niezależna od położenia obiektu w przestrzeni. Dlatego porównując wagi, możliwe jest rozpoznanie oraz wyznaczenie orientacji obiektów – tak jak w przypadku deskryptora EGI. Dodatkowo, porównując różnice części urojonych wag zespolonych, można wyznaczyć zmianę odległości wzdłuż wektorów normalnych. Omówione właściwości deskryptora CEGI skłaniają ku stwierdzeniu, że jest on lepiej dostosowany do indeksowania większej bazy obiektów niż EGI.

Histogram kształtu

Histogram kształtu trójwymiarowego po raz pierwszy został zaproponowany w [1]. Wyznaczenie deskryptora opiera się tu na odpowiednim podziale przestrzeni, w której znajdują się obiekty. We wspomnianej publikacji opisano trzy techniki:

- Podział wykorzystujący okręgi o wspólnym punkcie środkowym. Promień zewnętrznego okręgu jest zależny od wielkości największego obiektu, znajdującego się w bazie danych. Główną cechą takiego deskryptora jest niezależność od obrotu obiektu 3D względem środka układu współrzędnych.
- Podział wykorzystujący sektory, rozpoczynające się w punkcie centralnym modelu. Budowa deskryptora, opartego na tej metodzie, jest bardziej złożona niż w poprzednim przypadku. W pierwszej fazie realizacji podziału, wykorzystany zostaje wielościan foremny, aby w sposób regularny wyznaczyć punkty na powierzchni sfery otaczającej obiekt. Zdefiniowane punkty, w połączeniu z diagramem Voronoi, automatycznie dzielą przestrzeń na sektory o wspólnym punkcie centralnym. Otrzymany opis jest inwariantem obrotu i skalowania.
- Podział będący połączeniem dwóch poprzednich technik, bardziej złożony, ale zawierający więcej informacji o obiekcie 3D.



Rysunek. 2 Histogramy kształtu, zbudowane techniką wykorzystującą kolejno: okręgi współśrodkowe, sektory oraz złożenie obu, źródło: [1].

Ilustracja graficzna opisanych różnych wersji histogramu kształtu 3D, z uwzględnieniem różnych technik podziału przestrzeni, została przedstawiona na rys.2. Z lewej strony pokazano sposób podziału przestrzeniu, z prawej – histogram kształtu.

Rozkład kształtu

Kolejna metoda budowy deskryptorów kształtów trójwymiarowych koncentruje się na obliczeniu rozkładów kształtu na podstawie modelu. Pierwszym etapem jest tu wybór funkcji kształtu. Najważniejszą właściwością funkcji powinna być prostota oraz odporność na przekształcenia, którym mogą zostać poddane obiekty trójwymiarowe. W [12] i [13] przedstawione zostały następujące funkcje kształtu:

- **A3**: Miara kąta pomiędzy trzema losowymi punktami, znajdującymi się na powierzchni obiektu;
- **D1**: Miara odległości pomiędzy wybranym a losowym punktem, znajdującym się na powierzchni obiektu. Wybrany punkt to centroid powierzchni;
- **D2**: Miara odległości pomiędzy dwoma losowymi punktami, znajdującymi się na powierzchni obiektu;
- **D3**: Miara pierwiastka kwadratowego powierzchni trójkąta, którego wierzchołki są losowymi punktami, znajdującymi się na powierzchni obiektu;
- **D4**: Miara pierwiastka trzeciego stopnia objętości czworościanu, zbudowanego z czterech losowych punktów, znajdujących się na powierzchni obiektu.

Po wyborze funkcji kształtu wyznaczany jest rozkład kształtu. Dla danego obiektu trójwymiarowego obliczanych zostaje wiele przykładowych rozkładów kształtu. Na ich podstawie budowany zostaje histogram, przedstawiający ilość wcześniej obliczonych wartości mieszczących się w określonym przedziale. Rozkład

kształtu zostaje zrekonstruowany z wyznaczonego i zapisany w postaci wektora.

2.2. Deskryptory strukturalne

Wielo-rozdzielczościowy graf Reeba (MRG)
Pierwszy deskryptor strukturalny, oparty na teorii grafu Reeba, opisany został w [3]. Przedstawiono tam wielo-rozdzielczościowy graf Reeba (*Multiresolutional Reeb Graph – MRG*).

Graf Reeba jest jedną z podstawowych struktur danych, reprezentujących (kodujących) kształt i właściwości obiektu 3D. Pierwszym etapem jego budowy jest definicja funkcji μ (najczęściej równej funkcji wysokości), według której następuje podział modelu 3D na obszary. Każdy węzeł grafu Reeba reprezentuje składnik danego obszaru połączony krawędziami ze składnikami obszarów sąsiednich.

Graf MRG jest złożony z wielu grafów Reeba, wyznaczonych dla różnej liczby obszarów. Jako pierwszy wyznaczony zostaje graf dla jednego obszaru, składający się z jednego węzła, następnie liczba obszarów zostaje dwukrotnie zwiększona, itd. Wskutek podziału obiektu na różną liczbę obszarów, grafy MRG zawierają więcej informacji i lepiej aproksymują model 3D niż podstawowy graf Reeba. Węzły grafu Reeba o określonym poziomie rozdzielczości (określonej liczbie obszarów) mogą zostać wyznaczone poprzez połączenie węzłów grafu o większym poziomie rozdzielczości. Wykorzystując te właściwości, podobieństwo pomiędzy obiektami może zostać wyznaczone z użyciem strategii „od ogółu do szczegółu" (*coarse-to-fine*) dla różnych poziomów rozdzielczości.

Jednym z najważniejszych elementów budowy grafu Reeba i jednocześnie deskryptora MRG, jest wybór funkcji dzielącej obiekt 3D na obszary. W [3] wykorzystano funkcję, zdefiniowaną jako suma odległości geodezyjnej z punktu v do wszystkich punktów na płaszczyźnie S. Zapewnia ona niezmienność przy translacji obiektu oraz przy deformacji jego siatki, na przykład po pojawieniu się niewielkiego szumu i zakłóceń. Aby funkcja $\mu(v)$ zachowała niezmienność względem skalowania obiektu, dodano normalizację. Dodatkowo, z powodu braku punktu odniesienia (względem którego dokonywane są obliczenia), funkcja $\mu(v)$ jest stabilna, co oznacza, że zmiany w modelu nie wpływają na zmianę położenia punktu odniesienia, jak w przypadku standardowego obliczania odległości geodezyjnej.

Deskryptor szkieletowy
Deskryptor szkieletowy ([17]) jest reprezentacją „grafu szkieletowego" obiektu 3D. Graf zawiera we wszystkich węzłach informacje topologiczne, odnoszące się do lokalnych właściwości modelu oraz całości grafu. W przeciwieństwie do opisanego poprzednio grafu MRG deskryptor szkieletowy jest bardzo intuicyjnym deskryptorem kształtu, ponieważ w przejrzysty i zrozumiały dla użytkownika sposób wizualizuje strukturę obiektu 3D. Dzięki tej właściwości możliwe jest wydajniejsze wyszukiwanie obiektów podobnych, także w przypadku, gdy obiekt z zapytania stanowi część większego obiektu w bazie. Dodatkowo mamy możliwość wskazania części modelu 3D, która ma zostać odnaleziona lub nadania wyższej wagi określonym częściom podczas wyszukiwania. Niemniej istotną zaletą deskryptora szkieletowego jest możliwość zrozumiałej wizualizacji wyników wyszukiwania, co pomaga zrozumieć i ocenić stopień podobieństwa porównywanych modeli. Kolejną zaletą deskryptora szkieletowego jest stała topologia grafu dla poruszającego się obiektu, dzięki czemu może on być stosowany dla obiektów w ruchu.

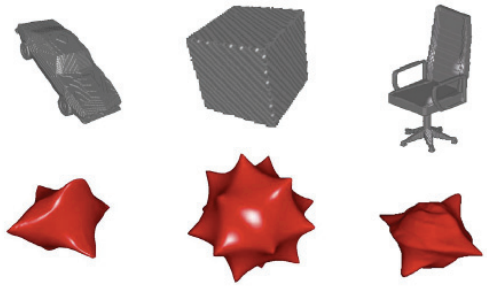
Proces wyszukiwania i porównywania obiektów za pomocą deskryptora szkieletowego rozpoczyna się od wyznaczenia szkieletu. W pierwszym etapie następuje 'wokselizacja' wszystkich modeli 3D oraz wyznaczenie płaszczyzny środkowej (*medial surface*). Następnie pomniejszany jest rozmiar uzyskanej płaszczyzny, tak aby mogła ona zostać przedstawiona w postaci grafu.

Deskryptor szkieletowy nie jest idealny dla wszystkich obiektów 3D. Trudność stanowić mogą proste modele, które odnajdywane zostają jako mniejsze części bardziej złożonych obiektów. Z drugiej strony, bardzo dużą zaletą grafów szkieletowych jest możliwość wpływania na wyniki rozpoznawania poprzez odpowiednie zdefiniowanie parametrów lokalnego dopasowania oraz złożoności szkieletu. Przy tym, czas obliczeń deskryptora jest proporcjonalny do liczby wokseli w danym obiekcie.

2.3. Deskryptory symetryczne

Deskryptor symetrii osiowej
Pierwsze badania nad symetrią kształtów skupione były na klasyfikacji obiektów ze względu na grupy symetrii. Jest to dobry sposób na zmniejszenie ilości informacji opisującej dany kształt, jeżeli występują w nim osie symetrii. Jednakże o wiele wydajniejsze jest stosowanie tzw. miar osiowych ([8]). Deskryptor symetrii osiowej (*Reflective Symmetry Descriptor* – RSD) jest dwuwymiarową funkcją opisującą symetrie względem każdej płaszczyzny poprowadzonej przez środek ciężkości obiektu (centroid). Ideą deskryptora jest wyznaczenie miary symetrii dla wszystkich płaszczyzn, nawet w przypadku, kiedy nie odpowiadają one idealnej symetrii osiowej modelu. Na rys. 3 znajdują się przykładowe obiekty oraz reprezentujące je RSD. Deskryptory zostały zwizualizowane poprzez skalowanie wektorów jednostkowych na kuli, proporcjonalnie do

miary symetrii płaszczyzny przechodzącej przez środek ciężkości obiektu i normalnej do wektora. Zauważyć można, że RSD zapewnia ciągłą miarę symetrii dla wszystkich płaszczyzn oraz posiada wierzchołki zgodne z płaszczyznami symetrii lub bliskie symetrii modelu. Deskryptor przechowuje bardzo dużo informacji o kształcie obiektu – poza opisem różniących się części obiektu, odzwierciedla ich przestrzenne zależności.



Rysunek. 3 Wizualizacja deskryptorów symetrii osiowej dla trzech obiektów, źródło: [8].

Deskryptor RSD ma szereg zalet. Po pierwsze daje w rezultacie charakterystyczny opis o cechach globalnych. Po drugie, z definicji wykorzystywana jest postać kanoniczna funkcji dwuwymiarowej, co ułatwia wspólną parametryzację w modelach trójwymiarowych. Dodatkowo, w określonych warunkach, reprezentacja jest odporna na szum i inne niewielkie zakłócenia, ale tylko dopóki wszystkie punkty symetrii tworzą wspólną całość

Dzięki zastosowaniu sferycznej funkcji harmonicznej uzyskujemy opis obiektu niezależny od obrotu w trzech wymiarach ([9]). Kolejną zaletą tego rozwiązania jest możliwość jego umieszczenia w innych istniejących deskryptorach, jako dodatkowy etap w ich konstrukcji. W porównaniu z innymi podobnymi rozwiązaniami normalizującymi lub przekształcającymi zapis obiektu (np. normalizacja względem obrotu) możemy uzyskać lepsze wyniki przy rozpoznawaniu obiektów 3D, przy jednoczesnym zmniejszeniu rozmiaru uzyskiwanego opisu oraz czasu porównywania. Przykładowymi reprezentacjami kształtu, które mogą zostać opisane z użyciem sferycznego zapisu harmonicznego są: EGI ([5]), histogram kształtu ([1]), EXT ([15]), opisujący maksymalny „zasięg" kształtu wzdłuż promieni wychodzących z jednego punktu środkowego, RSD ([8]) oraz EDT.

Rozszerzony deskryptor kształtu

Symetria jest szczególnie przydatną cechą, ponieważ opisuje globalne informacje o kształcie obiektu. Wydobycie niewielkiej ilości informacji o symetrii pozwala na uzyskanie opisu obiektu wystarczającego do skutecznego wyznaczania podobieństwa pomiędzy modelami. Ponadto, jeżeli dwa obiekty trójwymiarowe różnią się przynajmniej w jednym punkcie, to w odpowiedzi uzyskamy informację, że są one inne. Wcześniejsze, prostsze techniki, klasyfikujące modele ze względu na rodzaj symetrii, zwracają informację binarną – symetria jest lub nie. W przypadku deskryptorów symetrycznych, uzyskujemy niejako strukturę ciągłą ([10]), gdzie przechowywane są miary symetrii, nawet w przypadku, kiedy obiekt nie jest symetryczny. Dzięki takiej ciągłej klasyfikacji symetrii możemy porównywać i rozpoznawać obiekty bez wykonywania procesu normalizacji, koniecznego przy większości innych deskryptorów. Dodatkowo deskryptor jest niezależny od obrotu obiektu.

Bazując na powyższych cechach symetrii kształtu w [10] zaproponowano rozbudowanie istniejących deskryptorów kształtu o informację o symetrii. Rozszerzono w tym celu sferyczną reprezentację harmoniczną, opisaną w poprzednim rozdziale. Wadą tej reprezentacji jest rozpatrywanie każdego składnika częstotliwości niezależnie i brak informacji charakteryzujących ich położenie względem siebie. Dlatego też dodano do sferycznej reprezentacji harmonicznej o charakterze lokalnym globalną informację o symetrii modeli 3D.

Pierwszy etap procesu budowy rozszerzonego deskryptora symetrycznego polega na przekształceniu sferycznego deskryptora kształtu w sferyczną reprezentację harmoniczną. W tym celu funkcja sferyczna wyrażona zostaje przez składniki częstotliwości oraz przechowana zostaje norma każdego z nich. Następnie obliczone zostają deskryptory symetryczne, dla k-krotnych osi symetrii funkcji sferycznej. Ostatnim etapem jest rozszerzenie sferycznej reprezentacji harmonicznej o informacje o symetrii – skalowanie k-krotnych osi symetrii, aby otrzymać najlepszy rozmiar informacji o nieregularnej częstotliwości. Oznacza to, że dla każdego typu symetrii przechowywana jest kopia sferycznej reprezentacji harmonicznej. Dodatkowo, oddzielenie informacji o symetrii od informacji o częstotliwości, umożliwia efektywne porównanie dwóch modeli trójwymiarowych. Procesy porównywania symetrii i częstotliwości mogą być wykonywane niezależnie, a następnie wyniki mogą zostać połączone w jedną miarę podobieństwa.

2.4. Deskryptory lokalne

Opisywane dotychczas deskryptory kształtu trójwymiarowego miały charakter globalny. Powstały w celu ułatwienia procesu rozpoznawania obiektów 3D, jednak nie są wystarczająco skuteczne w sytuacji, gdy charakterystycznymi cechami określonej klasy obiektów są lokalne właściwości kształtu. Założenie podobieństwa obiektów w klasie na poziomie ogólnym (całego obiektu)

może się niekiedy okazać błędne. Dlatego też prowadzone są także prace z lokalnymi deskryptorami kształtu. W [16] uwaga skupiona została na analizie bazy danych obiektów i wyborze lokalnych właściwości, zapewniających najlepsze wyniki przy rozpoznawaniu modeli 3D. W tym celu obiekt reprezentowany był przez wiele lokalnych deskryptorów kształtu (każdy opisywał pewien podobszar kształtu). Podobieństwo pomiędzy dwoma modelami obliczone było na podstawie podobieństwa pomiędzy dwoma lokalnymi deskryptorami.

Inne przykładowe metody wyznaczania lokalnej reprezentacji kształtu bazują na wyborze podzbioru deskryptorów na podstawie istotności ([2]) lub prawdopodobieństwa ([6]). Tak zwane obrazy obrotowe ([6]) analizują obiekty poprzez utworzenie cylindrycznego odwzorowania lokalnych zbiorów punktów na powierzchni. Wadą w tym przypadku jest czas rozpoznawania, który znacznie wzrasta przy zwiększającej się liczbie porównywanych deskryptorów. Wyklucza to zastosowanie tego podejścia w dużych bazach danych. Aby przyspieszyć proces rozpoznawania modeli trójwymiarowych, można zastosować selekcję zbioru lokalnych deskryptorów. Najprostszą metodą jest wybór losowy, jednak w takim przypadku nie jest pewne, że wybrany deskryptor będzie wystarczająco charakterystyczny dla danego obiektu. Jest też bardzo prawdopodobne, że do uzyskania zadowalających wyników rozpoznawania będzie potrzebna większa liczba deskryptorów losowych.

Odrębny problem stanowi określenie, w jaki sposób wybierać podobszary obiektu, by można je było uznać za charakterystyczne dla niego. Pierwsze techniki bazowały na eksperymentach psychologicznych ([4]), według których system widzenia człowieka dzieli złożony kształt na części oraz wybiera określone właściwości, przed rozpoczęciem procesu rozpoznawania. W kolejnej metodzie ([2]) wskazywano części obiektu na podstawie właściwości jego krzywizny. Podobne rozwiązanie zastosowane zostało w [11].

System wyznaczania lokalnych deskryptorów kształtu opisany w [16] został podzielony na dwa etapy. Pierwszym jest trening deskryptorów dla obiektów znajdujących się w bazie, którego wynikiem jest funkcja dystynkcji (funkcja cech charakterystycznych dla obiektu). Na początku, kształty zostają znormalizowane względem rozmiaru, a na płaszczyźnie każdego obiektu wybierane są losowo punkty. W miejscu, w którym znajduje się wylosowany punkt, zbudowany zostaje deskryptor kształtu. Następnie, na podstawie wyników procesu rozpoznawania treningowej bazy modeli, wyznaczone zostaje prawdopodobieństwo dla każdego deskryptora, które określa najbardziej istotne (niepowtarzalne) cechy obiektu, wystarczająco dyskryminujące podczas procesu rozpoznawania.

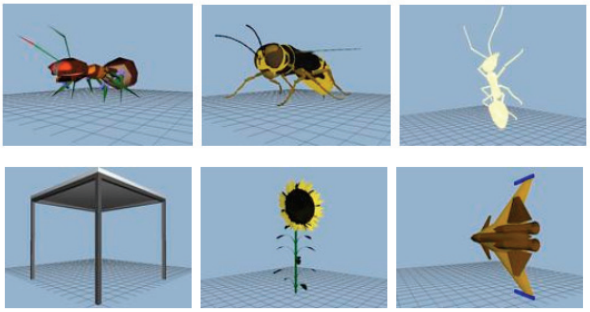
Następnie, zdefiniowana zostaje funkcja dystynkcji – mapująca prawdopodobieństwo deskryptora na przewidywany wynik procesu rozpoznawania. W tym celu dla każdego lokalnego deskryptora kształtu przeprowadzony zostaje proces rozpoznawania. Wyznaczona zostaje również jedna ze standardowych metryk rozpoznawania – *DCG (Discounted Cumulative Gain*). Jej wartość zawiera się w przedziale <0,1>, gdzie lepsze wyniki procesu rozpoznawania znajdują się bliżej wartości 1.

Dla wszystkich deskryptorów obiektów znajdujących się w bazie obliczone zostaje prawdopodobieństwo oraz DCG procesu rozpoznawania. Następnie deskryptory zostają uporządkowane według rosnącego prawdopodobieństwa, a w przypadku jednakowego prawdopodobieństwa w drugiej kolejności pod uwagę są brane średnie wartości DCG. W rezultacie otrzymujemy histogram średniego wyniku procesu rozpoznawania (DCG) w funkcji prawdopodobieństwa (indeksami są osiągnięte wartości prawdopodobieństwa – funkcja dystynkcji).

3. PODSUMOWANIE

W artykule przedstawiony został wstępny przegląd istniejących rozwiązań w dziedzinie reprezentowania kształtu 3D. Opisane metody podzielone zostały na cztery grupy – geometryczne, strukturalne, symetryczne i lokalne. W przypadku każdej z nich przedstawiono podstawowe właściwości najpopularniejszych obecnie rozwiązań.

W wielu badaniach nad deskryptorami kształtu 3D wykorzystywana jest baza danych „Princeton Shape Benchmark" ([20]). Zawiera ona 1814 modeli obiektów trójwymiarowych, znalezionych w Internecie. Dla każdego obiektu dostępny jest plik, przechowujący informację o geometrii obiektu, obraz dwuwymiarowy obiektu w formacie JPEG oraz plik tekstowy, zawierający podstawowe informacje, np. adres strony internetowej, na której się znajduje, ilość wielokątów, maksymalne i minimalne wartości zmiennych na poszczególnych osiach. Baza danych ułatwia w znacznym stopniu rozwój badań nad deskryptorami kształtu 3D, ponieważ zawiera obiekty o bardzo zróżnicowanej liczbie poligonów (od kilkudziesięciu do kilkudziesięciu tysięcy), różnym rozmiarze, położeniu oraz obrocie w przestrzeni trójwymiarowej.



Rysunek. 4 Przykładowe obrazy obiektów trójwymiarowych, znajdujących się w bazie *„Princeton Shape Benchmark”*, źródło: [20].

Ponieważ wzrasta liczba zastosowań deskryptorów kształtu 3D (w rozpoznawaniu, wyszukiwaniu, systemach CAD i wielu innych) można się spodziewać w najbliższych latach pojawienia się wielu nowych podejść, tak jak to miało miejsce w przypadku kształtów 2D, gdzie w przybliżeniu można obecnie określać liczbę stosowanych na świecie metod na kilka setek.

Literatura

1. Ankerst M., Kastenmüller G., Kriegel H., Seidl T., “3D Shape Histograms for Similarity Search and Classification in Spatial Databases”, 6th International Symposium on Spatial Databases, pp. 207-226, 1999
2. Gal R., Cohen-Or D., "Salient Geometric Features for Partial Shape Matching and Similarity", ACM Transactions on Graphics, vol. 25, pp. 130-150, 2006
3. Hilaga M., Shinagawa Y., Kohmura T., Kunii T. L., „Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes”, 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 203-212, 2001
4. Hoffman D. D., Singh M., “Salience of Visual Parts”, Cognition, vol. 63, pp. 29-78, 1997
5. Horn B., “Extended Gaussian Images”, Proc. of the IEEE A.I. Memo No. 740, Vol. 72 (12), pp. 1671–1686, 1984
6. Johsnon A., “Surface Landmark Selection and Matching in Natural Terrain”, IEEE Conf. on Computer Vision and Pattern Recognition, pp. 413-420, 2000
7. Kang S., Ikeuchi K., “Determining 3-D Object Pose Using the Complex Extended Guassian Image”, CVPR, pp. 580–585, 1991
8. Kazhdan M., Chazelle B., Dobkin D., Funkhouser T., Rusinkiewicz S., “A Reflective Symmetry Descriptor for 3D Models”, Algorithmica, vol 38, pp. 201-225, 2003
9. Kazhdan M., Funkhouser T., Rusinkiewicz S., “Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors”, 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 156-164, 2003
10. Kazhdan M., Funkhouser T., Rusinkiewicz S., “Symmetry Descriptors and 3D Shape Matching”, 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 115-123, 2004
11 Novotni M., Degener P., Klein R., “Correspondence Generation and Matching of 3D Shape Subparts”, Technical Report CG-2005-2, Universität Bonn, 2005
12. Osada R., Funkhouser T., Chazelle B., Dobkin D., “Matching 3D Models with Shape Distributions”, International Conference on Shape Modeling and Applications, pp. 154-166, 2001
13. Osada R., Funkhouser T., Chazelle B., Dobkin D., “Shape Distributions”, ACM Transactions on Graphics, vol. 21, pp. 807-832, 2002
14. Roach J., Wright J., Ramesh V., “Spherical Dual Images: A 3D Representation Method for Solid Objects that Combines Dual Space and Gaussian Spheres”, CVPR, pp. 236-241, 1986
15. Saupe D., Vranić D. V., “3D Model Retrieval with Spherical Harmonics and Moments”, 23rd DAGM-Symposium on Pattern Recognition, pp. 392-397, 2001
16. Shilane P., Funkhouser T., “Selecting Distinctive 3D Shape Descriptors for Similarity Retrieval”, IEEE International Conference on Shape Modeling and Applications, 2006
17. Sundar H., Silver D., Gagvani N., Dickinson S., “Skeleton Based Shape Matching and Retrieval”, Shape Modeling International, pp. 130-139, 2003
18. “Extended Gaussian Images”, http://www.cs.cf.ac.uk/Dave/AI2/node197.html
19. Kazhdan M., „Presentation of Extended Gaussian Images”, http://www.cs.jhu.edu/~misha/Fall04/EGI1.ppt
20. „Princeton Shape Benchmark”, http://shape.cs.princeton.edu/benchmark/

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str.15-22

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str. 23-30

NIEINWAZYJNE METODY OCENY I POPRAWY BEZPIECZEŃSTWA SYSTEMÓW KOMPUTEROWYCH BAZUJĄCE NA STANDARDACH DISA

Marcin Kołodziejczyk

Akademia-Górniczo-Hutnicza
wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Al. Mickiewicza 30
30-059 Kraków
e-mail: marcink@agh.edu.pl

Artykuł jest próbą analizy dostępnych, nieinwazyjnych metod pozwalających na całościową analizę i ocenę bezpieczeństwa systemów komputerowych. Opisane metody nie ingerują bezpośrednio w działanie gotowych systemów, skupiając się w dużej mierze na statycznej analizie systemu. Analiza taka opiera się na stwierdzeniu zgodności systemu z poszczególnymi wymaganiami bezpieczeństwa, tworzonymi przez organizację DISA. Organizacja ta swoje wymagania przedstawia w trojaki sposób: tworząc statyczne wymagania tzw. STIG'i, manualne procedury, mające pomóc w analizie bezpieczeństwa oraz skrypty automatyczne dla części systemów (tzw. SRR). Przekrój systemów objętych wymaganiami jest ogromny, zaczynając od systemów operacyjnych, idąc przez systemy bazodanowe, sieci a skończywszy na tworzonych przez programistów aplikacjach. Artykuł zawiera również krótką analizę pewnych braków, niedogodności i wad wyżej wymienionych metod oraz jest próbą odpowiedzi w jakim kierunku powinien iść ich rozwój.

Słowa kluczowe: *Autentykacja (uwierzytelnianie), autoryzacja, aplikacja, baza danych, bezpieczeństwo, cheklista (procedura manualna), GoldDisk, hasła, interfejs, konta użytkowników, Linux, logi, logowanie zdarzeń systemowych, Microsoft Windows, PKI (infrastruktura klucza publicznego), SRR, STIG, system operacyjny, Unix, usługi sieciowe, wymagania*

Non-intrusive assessment approach and improving security of computer systems based on DISA standards

Abstarct: *This article is an attempt to analyze the available, non-intrusive methods of analyzing and assessing if the overall security of computer systems. These methods do not interfere directly in the working systems, focusing largely on static analysis of the system. The analysis base on the finding of compliance with the various safety requirements, formed by the DISA organization. This organization presents its requirements in three ways: by creating the static requirements, so-called STIGs, manual procedures (checklists), helped in the analysis of the security and automated scripts for system components (SRR). The scope of the systems covered by the requirements is very big, including operating systems, database systems, the network and applications created by software developers. This article also contains a brief analysis of some shortcomings, disadvantages and drawbacks of the above-mentioned methods and is an attempt to answer for the question: what should be done in the nearest future for static security analysis.*

Keywords: *authentication, authorization, application, checklist (manual procedure), database, GoldDisk, passwords, interface, Linux, logs, Microsoft Windows, network services, operating system, PKI (Public Key Infrastructure), requirements, security, SRR, STIG, system event logging, Unix, user accounts*

1. WSTĘP

Celem niniejszej publikacji jest prezentacja metod weryfikacji bezpieczeństwa istniejących, bądź tworzonych systemów informatycznych i telekomunikacyjnych w oparciu o wytyczne organizacji DISA (ang. Defence Information Systems Agency) dostarczającej rozwiązania informatyczne m.in. dla Departamentu Obrony Stanów Zjednoczonych (ang. Departament of Defense). Istnieje wiele innych metod audytu jak COBIT, CRAMM, które nie są tematem tego

artykułu. DISA zdecydowanie się wyróżnia najbardziej precyzyjnymi wymaganiami dotyczącymi technicznych aspektów bezpieczeństwa systemu komputerowego. Przez to, że większość wymagań tworzonych przez organizację DISA jest przeznaczona dla Departamentu Obrony USA, można zakładać, że poziom bezpieczeństwa opisany w takich wymaganiach jest wysoki. Wymagania te pozwalają nie tylko ocenić aktualny stan bezpieczeństwa systemu, ale również identyfikują słabe punkty systemu, a także pozwalają identyfikować zagrożenia już na etapie tworzenia danej wersji systemu. Metody oparte na wytycznych Departamentu Obrony są w dużej mierze bezinwazyjne w związku z czym istnieje znikome ryzyko uszkodzenia działającego systemu. Metody te zakładają również bardzo dobrą znajomość badanego systemu przez osobę, która przeprowadza audyt bezpieczeństwa. Wymagania stawiane przez DISA są podzielone na różne kategorie, względem rodzaju systemu jak również zagrożenia stwarzanego przez potencjalne dziury w bezpieczeństwie poszczególnych składowych systemu.

2. PODZIAŁ WYMAGAŃ [1]

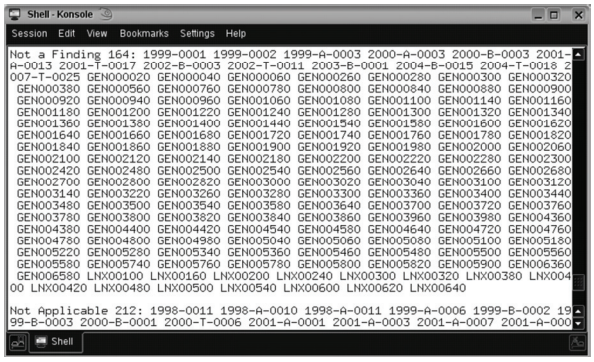
Duże, złożone systemy często są budowane w oparciu o środowiska przynajmniej częściowo heterogeniczne (np. systemy Unix i Windows). Z kolei ilość małych systemów, korzystających nawet tylko z jednego konkretnego systemu operacyjnego i pisanych w jednym konkretnym środowisku programistycznym jest ogromna. Nie da się, więc stworzyć jednego dokumentu, określającego niezbędne wymagania bezpieczeństwa dla wszystkich systemów. Należy, podzielić ogólne wymagania systemowe na mniejsze i dokładniejsze, wydzielając przy tym konkretne wymagania stawiane przed poszczególnymi składowymi systemu. Tak, więc wymagania dla systemu Linux będą nieco inne niż wymagania dla systemu Microsoft Windows. Podział taki ma jeszcze jedną zaletę, mianowicie wraz z wprowadzeniem nowej wersji produktu na rynek należy zmienić tylko jedną linię wymagań. Wytyczne tworzone przez organizację DISA obejmują systemy operacyjne, serwery baz danych, tworzone aplikacje użytkownika, konfigurację sieci, urządzeń USB, Web-serverów, maszyn wirtualnych itp. Zabierając się za analizę bezpieczeństwa konkretnego systemu należy najpierw wyodrębnić jego składowe, dla których DISA dostarcza konkretne wymagania.

Kolejnym podziałem jest podział wytycznych na trzy kategorie: wymagania, zwane dalej STIG'ami (ang. Security Technical Implementation Guides), procedury manualne (tzw. checklisty) i skrypty (ang. SRR - Security Readiness Review evaluation scripts). Należy tutaj zwrócić uwagę, że niestety te 3 kategorie bardzo często różnią się wieloma szczegółami i nie są sobie równoważne. Wszystkie trzy kategorie wymagań się częściowo uzupełniają (np. czasami procedura manualna jest bardziej restrykcyjna niż skrypt lub odwrotnie), chociaż niestety w wielu przypadkach są sprzeczne w szczegółach. To samo wymaganie w kilku miejscach może mieć inne znaczenie. Przykładem może być chociażby wymaganie GEN000800 dla systemów UNIX. Wymaganie (STIG) mówi, że ostatnie 10 haseł nie może być ponownie użyte, natomiast procedura manualna mówi o 5 hasłach. Dla przeciętnego programisty i administratora nie są to, więc duże różnice. Chcąc jednak spełnić jak najwięcej wytycznych organizacji DISA, wymagania takie jak przytoczone wyżej mogą sprawiać kłopot. Na szczególną uwagę zasługują skrypty, które można błyskawicznie uruchomić na badanym systemie, nie zmieniając przy tym żadnych z jego ustawień. Skrypty takie niestety dostępne są tylko na ograniczoną liczbę systemów. Przykładem mogą być skrypty SRR dla systemów klasy UNIX oraz tzw. GoldDisk dla Microsoft Windows. Brakuje np. skryptów dla wielu baz danych, przez co cała analiza bezpieczeństwa musi być przeprowadzana ręcznie.

Każde wymaganie przynależy do jednej z kilku kategorii. Kategoria I jest najbardziej krytyczną kategorią z punktu widzenia bezpieczeństwa. Luki w implementacji wymagań kategorii I mogą prowadzić do największych zniszczeń w systemie i wiążą się ze stosunkowo dużym ryzykiem ataku bądź przypadkowego uszkodzenia systemu. Najmniej ważne wymagania znajdują się w kategorii III lub IV (zależnie od systemu). Niespełnienie części z tych wymagań nie pociąga za sobą dużego ryzyka dla bezpieczeństwa systemu.

3. SYSTEMY UNIX

Wymagania dla systemów Unix obejmują wiele odmian tego systemu. Przykładami mogą być systemy takie jak: Solaris, HP-UX, AIX, IRIX oraz Linux. Problem sprawdzenia pewnych ustawień systemowych nie jest trywialny, gdyż niektóre, specyficzne pliki konfiguracyjne na Solarisie mogą być umiejscowione w innym katalogu niż np. w systemie HP-UX czy Linux. Warto tutaj wspomnieć, że skrypty automatyczne, tzw. SRR'y, dostępne dla Unix'a obsługują wszystkie wymienione systemy, uwzględniając różnice między nimi. Niestety nie są one jednak wolne od błędów.



Rysunek 1: Fragment wyniku skryptów SRR uruchomionych na systemie Linux. Widoczna tutaj jest lista spełnionych wymagań (Not a Finding) oraz fragment specyficznych wymagań, które się nie aplikują do danej wersji systemu (Not Applicable).

Pierwszą ważną rzeczą, na jaką zwraca się uwagę w wymaganiach jest integralność systemu. Chodzi tutaj zarówno o integralność sprzętową (np. nie wpinanie dodatkowych urządzeń do sieci) jak również integralność programową, która może być zapewniona przez okresowe sprawdzanie sum kontrolnych plików (tzw. hash'y), które są niezbędne do prawidłowej pracy systemu. W ten sposób można weryfikować czy pliki systemowe nie zostały zmienione (zmianę taką może powodować np. wirus, koń trojański czy inne dowolne, szkodliwe oprogramowanie). Watro zwrócić uwagę, że większość wymagań dot. integralności jest kategorii II. Tylko dwa z nich są kategorii I.

Kolejnym ważnym aspektem bezpieczeństwa systemów Unix, jest kontrola dostępu do wszelkiego rodzaju plików i zasobów. DISA w swoich wymaganiach precyzuje tutaj konfiguracje kont użytkowników, włącznie z określeniem warunków integralności między plikami takimi jak: /etc/passwd, /etc/shadow oraz /etc/group. Wymagania te specyfikują również zakres dostępnych identyfikatorów dla użytkownika i grupy, precyzując, które z nich są systemowe lub zarezerwowane i których nie powinno się używać. Ważną rzeczą, z prawnego punktu widzenia wielu państw (m.in. USA) jest wypisywanie odpowiedniego komunikatu w trakcie interakcyjnego procesu logowania użytkownika. Aby można było wyciągnąć prawne konsekwencje od intruza, należy go po prostu przedtem o takiej możliwości poinformować i wyświetlić stosowny komunikat (np. że ma do czynienia z systemem federalnym i nieautoryzowany dostęp jest karalny). Konta użytkowników powinny być obłożone restrykcjami czasowymi, zarówno pod względem długości ważności konta, jak również mechanizmu wylogowywania lub zamrażania sesji użytkownika po 15 minutach nieaktywności. Konta powinny być również zamrażane, jeśli użytkownik nie skorzysta z niego dłużej niż np. przez 35 dni. Wymagania dostarczane przez STIGi są bardzo restrykcyjne względem haseł użytkowników. Należy tutaj zauważyć jednak, że obecna wersja STIG'ów powstała w 2006 roku i wymagania dot. haseł zostały zmodyfikowane w checklistach oraz SRR'ach na bardziej restrykcyjne. Wymagania te precyzują długość haseł a także to, że każde hasło powinno zawierać małe i duże litery, cyfry oraz znaki specjalne. Szczególną uwagę przywiązuje się do konta administratora zwanego „root". Konto to powinno być szczególnie chronione chociażby przez sprawdzanie zmiennej środowiskowej $PATH (tak, aby administrator przypadkiem nie uruchomił innego programu niż zamierzony) czy w przypadku zdalnego dostępu do tego konta, zapewnienie tylko i wyłącznie szyfrowanego połączenia SSH. Użytkownik „root" posiada specjalne wymagania dotyczące praw dostępu do swoich plików, które różnią się od wymagań stawianych przed resztą użytkowników.

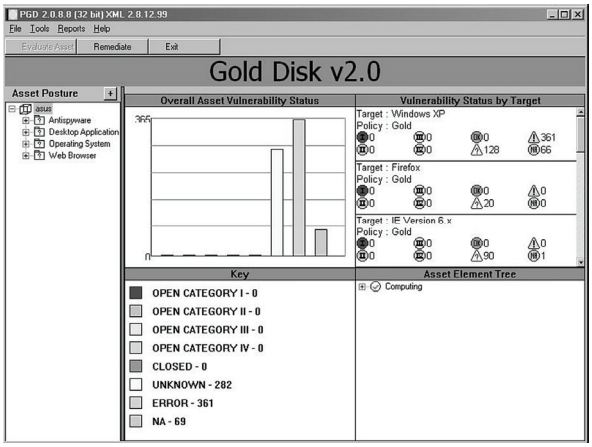
Ważną rzeczą w przypadku awarii lub włamania do systemu jest posiadanie stosownych logów. Wymagania Unix'owe kładą szczególny nacisk na procesy syslog'a oraz audit'a. Pierwszy z nich powinien działać na osobnej maszynie tak, aby w przypadku zapchania się logów nie odcinać dostępu do części funkcjonalnej systemu. Wyjątkiem, który pozwala na użycie syslog'a lokalnie jest sytuacja, w której syslog loguje tylko lokalne zdarzenia i nie akceptuje żadnych komunikatów z zewnątrz. O ile syslog nie jest zbyt uciążliwy dla systemu o tyle proces audyt może generować ogromne ilości informacji (włącznie z logowaniem udanych/nieudanych prób dostępu do pliku, czytania/pisania itd.). Może to powodować błyskawiczne zapychanie dysku, jednak DISA mimo to wymaga, aby pewne zdarzenia systemowe niezależnie od tego były logowane. Sposobem na bezpieczną implementację tych wymagań może być wprowadzenie rotacji logów, tak aby najstarsze logi były pakowane a następnie przy braku miejsca usuwane z dysku lub archiwizowane na zewnętrznych nośnikach danych. Dziwnym wydaje się tutaj fakt, że o ile istnieją wymagania dotyczące rotacji logów audytowych o tyle brak jest takich wymagań dla sysloga.

W przeciwieństwie do systemów Windows, wiele dystrybucji Linux'a zawiera dużą liczbę różnych serwerów i usług sieciowych. Można tutaj wymienić serwery www, poczty elektronicznej, ftp, tftp, telnet, DNS, samba, SSH, SNMP i inne. Wiele z nich jest opisanych i posiadają swoje, precyzyjnie zdefiniowane wymagania w STIG'ach.

4. SYSTEMY MICROSOFT WINDOWS

Obecna wersja wymagań dla systemów Microsoft Windows powstała w maju 2007 roku i obejmuje systemy Windows 2000, 2003, XP, 2008 oraz Vista. W chwili pisania artykułu brak jest wymagań dla specyficznych dla Windowsa 7. Podobnie jak w przypadku systemów UNIX, pierwszą rzeczą na jaką zwraca się uwagę w wymaganiach jest spójność systemu i cotygodniowe sprawdzanie czy pliki systemowe nie zostały zmienione. W tym przypadku podany jest nawet przykładowy skrypt, który może być uruchamiany co tydzień.

STIG'i dla Microsoft Windows przywiązują dużą wagę do bootowania systemu operacyjnego. Uruchomienie innego systemu z płyty bootowalnej lub pen drive'a znacząco ułatwia wiele rodzajów ataków. W przypadku systemów Windows znika częściowo problem niespójności wymagań z ręczną „checklistą", gdyż w przypadku systemu plików, wymagania w większości powołują się na checklistę, same nie precyzując żadnych konkretnych praw dostępu. Podobnie rzecz ma się z rejestrem systemowym. STIG'i dla systemu Windows również odnoszą się do logowania zdarzeń systemowych, konta administratora, automatycznego wylogowywania użytkownika po określonym czasie bezczynności oraz do haseł i usług sieciowych.



Rysunek 2: Program GoldDisk dla systemu MS Windows, służący do analizy aktualnego stanu bezpieczeństwa systemu operacyjnego.

Warto zwrócić uwagę, że w przypadku systemu MS Windows pojawiają się nowe wymagania dotyczące na przykład kosza. Ze względów bezpieczeństwa, wszystkie pliki powinny być usuwane natychmiast, nie powinny w ogóle się znajdować w koszu. Jest to poniekąd zablokowanie funkcjonalności dostarczanej przez kosz. Wymagania dotyczące kosza, nie istnieją jak na razie dla systemów Linux mimo, że wiele środowych graficznych posiada zaimplementowaną taką funkcjonalność. Wymagania dla Windows'a obejmują dodatkowo szeroki zestaw oprogramowania dostarczanych z systemem operacyjnym jak chociażby: .NET Framework, MSN Explorer, Windows Messenger, Media Player czy Internet Explorer. Microsoft dostarcza dla nowych systemów narzędzie zwane „Security Compliance Management Toolkit". Jest to owoc współpracy Microsoftu i National Security Agency, który pozwala w łatwiejszy sposób zabezpieczyć system w sposób wymagany przez STIGi.

5. BEZPIECZEŃSTWO BAZ DANYCH

Bazy danych są integralną częścią wielu systemów informatycznych. Niestety wymagania tworzone przez DISA dla baz danych mają postać płaską, czyli nie posiadającą żadnej hierarchii jak to ma miejsce w przypadku systemów operacyjnych. Wymagania bazodanowe nie są przez to w żaden sposób pogrupowane. Pierwszą ważną rzeczą poruszaną przez STIG'i jest to, aby używana baza danych była w wersji wspieranej przez producenta i aby były na niej zainstalowane wszystkie aktualne poprawki (ang. patch), zwłaszcza te dotyczące bezpieczeństwa (tj. security patche) [1,8]. Podobnie jak w przypadku systemów operacyjnych, należy monitorować wykonywalne pliki baz danych oraz wszystkie biblioteki używane przez bazę. Zmiana pliku wykonywalnego z silnikiem bazy danych może umożliwić intruzowi nieautoryzowany dostęp do danych. Również prawa dostępu do wszystkich plików skojarzonych z bazą powinny być jak najbardziej restrykcyjne. Ciężko jest tutaj podać konkretne wartości, gdyż liczba dostępnych na rynku silników baz danych jest ogromna. Ponadto bazy te można instalować na różnych platformach systemowych i sprzętowych.

Aby zapobiec utracie dużej części informacji zaleca się tworzenie kopii zapasowej wszystkich baz danych dostępnych w systemie. Wymagania nie precyzują jednak rodzaju kopii, czy ma to być tzw. backup pełny, przyrostowy czy różnicowy, ani jak często ma być taki backup wykonywany. Tworzenie kopii w przypadku baz danych różni się nieco od zwykłego kopiowania plików. Każda baza do tego celu udostępnia swoje narzędzia, które w poprawny sposób zrzucają zawartość bazy do pliku. Zwykłe kopiowanie może np. naruszyć więzy integralności. Kopiując plik nie mamy pewności czy akurat nie ma rozpoczętych transakcji w bazie lub czy baza nie aktualizuje jakichś danych. Specjalne narzędzia dostarczane przez producentów baz danych rozwiązują te

problemy. Warto zwrócić uwagę, że również kopia zapasowa powinna być ściśle chroniona odpowiednimi prawami dostępu, gdyż wydostanie się takiego backupu na zewnątrz może być równoznaczne z wyciekiem wszystkich danych z bazy.

Każdy dodatkowy, opcjonalny komponent, który nie jest należycie utrzymywany jest potencjalną luką w bezpieczeństwie systemu. Dlatego w systemach akredytowanych przez DISA wymaga się, aby wszystkie nieużywane komponenty bazy danych były usunięte. Przykłady, dostępne często w katalogach „examples" również pochodzą pod te wymagania a co za tym idzie, na systemie produkcyjnym nie powinny być dostępne.

O ile to możliwe zaleca się szyfrowanie wszystkich połączeń do bazy danych oraz samych danych w niej zawartych. Wymagania obejmują również tzw. infrastrukturę klucz publicznego (ang. Public Key Infrastructure – PKI) [6,7]. Niestety nie wszystkie dostępne na rynku silniki baz danych implementują wspomnianą funkcjonalność. Podobnie rzecz ma się z audytami bazodanowymi oraz logami. Mimo wielu wymagań, w większości systemów tylko część z nich może być spełniona, gdyż wiele z wymaganej funkcjonalności jest nie zaimplementowane.

Każdy użytkownik czy aplikacja korzystająca z bazy danych powinna mieć swoje osobne konto. Do konta tego powinny być przydzielone role, które uprawniają danego użytkownika do wykonania tylko niezbędnych dla niego operacji. Wszelkie inne operacje powinny być zabronione. W przypadku aplikacji mającej dostęp do bazy, hasła nie mogą być przechowywane tekstem jawnym, w żadnym pliku, skrypcie czy kodzie źródłowym. W przypadku narzędzi interaktywnych, w których użytkownik proszony jest o podanie swojej nazwy i hasła, hasło to nie powinno pojawiać się na ekranie jako tekst jawny podczas jego wpisywania. Wymagania dotyczące blokowania kont czy komplikacji haseł są podobne jak dla systemów operacyjnych Unix i Windows.

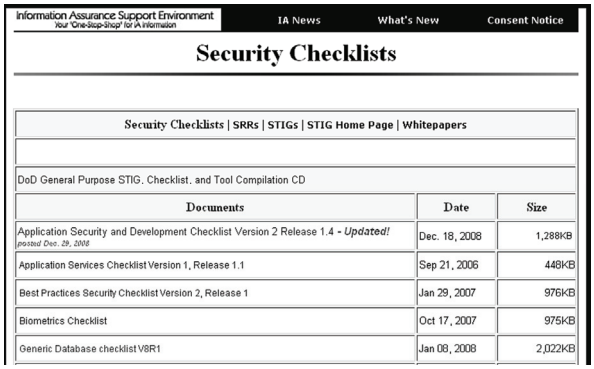
STIGi bazodanowe zalecają również blokowanie dostępu z bazy do zewnętrznych obiektów. Jeśli taki dostęp istnieje, należy odpowiednio nim zarządzać i powinien być stosownie udokumentowany. O ile to możliwe warto również zainstalować serwer bazy danych na osobnej maszynie, przeznaczonej tylko dla konkretnej bazy. Zmniejszy to ryzyko zablokowania dostępu do bazy przez inną usługę działającą niewłaściwie. Liczba równoległych połączeń do bazy powinna być również limitowana. Na szczególną uwagę zasługuje liczba równolegle wykonujących się transakcji. Może ona mieć duży wpływ na wydajność bazy i w skrajnych przypadkach może być niebezpieczna i służyć do przeprowadzenia tzw. ataku DoS (ang. Denial of Service).

Ostatnią rzeczą wartą uwagi w przypadku baz danych są skrypty automatyczne. Niestety dostępne są one tylko dla dwóch baz: MS SQL oraz Oracle (w tym drugim przypadku dostępne są skrypty na platformy UNIX'owe i Windowsowe). Autor testował tylko skrypty dla bazy Oracle. W przypadku wymagań specyficznych dla Oracle'a skrypty spisują się bardzo dobrze. Sytuacja ma się gorzej w przypadku wymagań ogólnych, gdyż tylko ok. 10% wymagań jest sprawdzanych przez skrypt. Powodem braku skryptów dla reszty silników bazodanowych jest ich różnorodność i specyficzność. Należałoby bowiem stworzyć osobne skrypty dla każdej bazy a ponadto w wielu przypadkach skrypty te byłyby zależne od konkretnej platformy.

6. INNE STIG'I

Oprócz opisanych wyżej STIG'ów dla systemów Unix, Windows oraz baz danych istnieje całe mnóstwo innych. Czasami zdarza się, że mimo, iż sam dokument ze STIG'ami nie istnieje, to istnieją procedury manulane (tzw. checklisty) lub skrypty automatyczne. Niestety w przypadku STIG'ów jedynymi systemami operacyjnymi dla których istnieją precyzyjne wytyczne są systemy Windows i UNIX/Linux. Nie zostały tutaj opisane pozostałe wymagania takie jak np. wymagania dot. sieci bezprzewodowych [4], maszyn wirtualnych czy urządzeń biometrycznych. Oprócz wyżej wymienionych istnieją jeszcze wymagania aplikacyjne, które skupiają się na bezpiecznym ich tworzeniu przez programistów. Wymaga się np. aby aplikacja sprawdzała na swoim interfejsie dane wejściowe jeśli np. obsługuje formatujące ciągi znaków, aby była odporna np. na taki typu „buffer overflow". Duży nacisk położony jest również na uwierzytelnianie użytkowników aplikacji oraz na autoryzację zadań przez nią wykonywanych. Ostatnie wersje STIG'ów aplikacyjnych zawierają kilkadziesiąt nowych wymagań w większości dotyczących samego procesu wytwarzania oprogramowania, np. używania kontroli wersji czy monitorowania wszelkiego rodzaju błędów.

Wszystkie, opisane i nieopisane tutaj wymagania, skrypty i procedury można znaleźć na stronach DISA [1]. Warto tylko zwrócić uwagę na ilość dokumentów. W chwili pisania artykułu było ich ok. 30. Procedur manualnych było natomiast ok. 70. Skryptów automatycznych było ok. 20.



Information Assurance Support Environment
Your "One-Stop-Shop" for IA Information

IA News | What's New | Consent Notice

Security Checklists

Security Checklists | SRRs | STIGs | STIG Home Page | Whitepapers

DoD General Purpose STIG, Checklist, and Tool Compilation CD

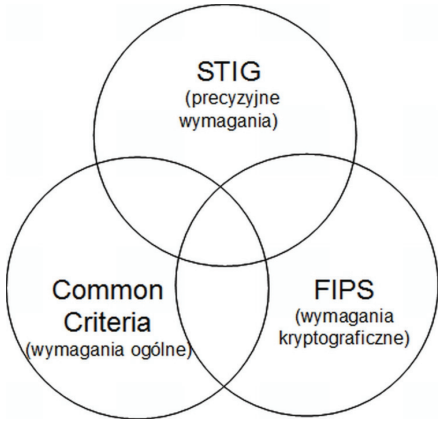
Documents	Date	Size
Application Security and Development Checklist Version 2 Release 1.4 - *Updated!* posted Dec. 29, 2008	Dec. 18, 2008	1,288KB
Application Services Checklist Version 1, Release 1.1	Sep 21, 2006	448KB
Best Practices Security Checklist Version 2, Release 1	Jan 29, 2007	976KB
Biometrics Checklist	Oct 17, 2007	975KB
Generic Database checklist V8R1	Jan 08, 2008	2,022KB

Rysunek 3: Fragment strony DISA [1] prezentującej dokumenty z wymaganiami bezpieczeństwa

Na rysunku 3 widać fragment strony z procedurami manualnymi (checklistami). W celu zdobycia skryptów automatycznych lub wymagań, należy wybrać z górnego menu odpowiednio „SRRs" lub „STIGs". W przypadku zastosowań w systemach komercyjnych, trzeba pamiętać, że mimo iż wszystkie wymagania są dostępne publicznie to jednak dostępne są tylko w najnowszej wersji. Dlatego tworząc konkretny projekt watro wcześniej zrobić kopię bazową wymagań na której system ma bazować.

7. PROCES COMMON CRITERIA [2]

Poruszając temat audytowania bezpieczeństwa systemów komputerowych należy wspomnieć o procesie zwanym Common Criteria. Jest to bardzo znany i rozpowszechniony proces oceny systemu komputerowego. Jest to proces zdecydowanie bardziej ogólny niż ocena zgodności ze STIG'ami. Oznacza to, że proces ten jest mniej techniczny. Jest to powód dla którego autor pominął go w dokładnych rozważaniach w artykule.



Rysunek 4: Zależności między STIG'ami, Common Criteria oraz FIPS

Common Criteria definiuje kilka podstawowych pojęć. Pierwszym z nich jest tzw. profil ochrony (ang. Protection Profile). Jest to zbiór wymagań jakie powinien spełniać konkretny produkt, np. System operacyjny, antywirus czy jakieś urządzenie sieciowe np. router. Jednym z etapów certyfikacji wg. Common Criteria jest wybranie celu, tzw. Security Target. Jest to dokładny opis właściwości jakie spełnia certyfikowany obiekt. Weryfikacja dokonywana jest na jednym z siedmiu poziomów pewności – EAL (ang. Evaluation Assurance Level). Wyższy poziom nie oznacza większego bezpieczeństwa systemu. EAL określa sposób oceny zgodności produktu z deklarowaną specyfikacją (np. testy funkcjonalne, formalna weryfikacja itd.)

8. STANDARDY FIPS [3]

FIPS (ang. Federal Information Processing Standard) jest organizacją zajmującą się głownie kryptograficznymi aspektami bezpieczeństwa. Na FIPS standardowo składają się dwie podgrupy. Pierwszą z nich jest Cryptographic Algorithm Validation Program (CAVP). Część ta odpowiedzialna jest za testowanie i ocenę algorytmicznych rozwiązań kryptograficznych. Nie ingeruje ona w konkretne implementacje a jedynie zaleca używanie wyspecyfikowanych algorytmów. Drugą gałęzią jest Cryptographic Module Validation Program. Ta część organizacji zajmuje się testowaniem konkretnych implementacji algorytmów. FIPS wypuszcza co jakiś czas listę certyfikowanych produktów. Obecna wersja tej listy to FIPS 140-2. Na liście znajdują się konkretne produkty wraz z podanie dokładnej wersji sprzętu lub oprogramowania (w zależności od rodzaju

produktu). Warto zwrócić uwagę, że nie każda wersja oprogramowania jest certyfikowana i obecność produktu z przed roku nie oznacza, że najnowszy produkt też jest bezpieczny i znajduje się na liście FIPS 140-2. W systemach o wysokich wymaganiach dot. bezpieczeństwa warto więc sprawdzać obecność używanych bibliotek i modułów na liście FIPS 140-2.

9. DALSZE KIERUNKI ROZWOJU

Niestety ciężko jest przewidzieć zmiany w wymaganiach tworzonych przez DISA oraz inne organizacje. Po analizie aktualnego stanu oraz rozwoju jaki nastąpił przez ostatnie dwa lata, można jednak spróbować wyciągnąć pewne wnioski. Wydaje się, że na pewno wraz z pojawieniem się nowego systemu operacyjnego Microsoftu, Windowsa 7, muszą pojawić się dla niego nowe, specyficzne wymagania dotyczące jego bezpieczeństwa. W przypadku systemów UNIX widać niewielki zastój w wymaganiach, które nie zmieniły się od marca 2006 roku. Jednak mimo to co kilka miesięcy powstają nowe wersje checklist, czyli manualnych procedur, które czasami są bardziej restrykcyjne niż same STIG'i. Co jakiś czas pojawiają się również skrypty automatyczne, które zawierają mniej błędów i testują coraz więcej wymagań. Widać wyraźnie, że sporym problemem są bazy danych, do których – w wielu przypadkach – brak jest oprogramowania testującego oraz wymagania nie są uporządkowane i nierzadko można je interpretować na kilka sposobów. Niektóre wymagania osiągnęły już swój docelowy poziom dojrzałości, niektóre zaś nadal ewoluują i co kilka miesięcy pojawiają się w nowych wersjach. Wydaje się, że DISA w dalszym ciągu będzie utrzymywała i aktualizowała swoje wymagania z czasem uzupełniając je skryptami automatycznymi lub coraz ściślejszymi procedurami manualnymi. Aplikacje co prawda mają swoje wymagania, jednak problemem wydaje się stwierdzenie czy dana aplikacja je spełnia. Jak zagwarantować, że tworzona aplikacja jest odporna np. na atak buffer overflow albo nie ma w ogóle wycieków pamięci? [5] Można użyć wielu komercyjnych narzędzi do wychwytywania tego typu błędów programistycznych, jednak o wiele łatwiej jest znaleźć błąd niż udowodnić, że go nie ma.

10. PODSUMOWANIE I WNIOSKI

Niniejsza publikacja miała na celu przegląd i krótkie wprowadzenie do metod nieinwazyjnego audytu bezpieczeństwa systemów opartych na wytycznych organizacji DISA oraz krótkie porównanie ich z innymi najbardziej znanymi metodami. Autor zastanawia się nad dalszymi kierunkami badań i rozwoju metod oceny bezpieczeństwa systemów. Istniejące metody pozwalają w znacznym stopniu zweryfikować poziom zabezpieczeń, jak również w wielu przypadkach pokazują pewne spojrzenie na kwestie bezpieczeństwa widziane od strony organizacji mających wysokie wymagania dotyczące tej kwestii. Zaletą opisanych wymagań i narzędzi jest fakt, że w trakcie testowania nie powinny uszkodzić systemu, w przeciwieństwie do innych, często inwazyjnych narzędzi takich jak np. skanery sieciowe takie jak Nessus, Nmap czy Foundstone. DISA jednak zaleca mimo to tworzenie kopii zapasowych przed uruchomieniem swoich skryptów. Opisane metody nie mogą zagwarantować, że system jest bezpieczny. W celu dokładnej weryfikacji należy zastosować przynajmniej kilka różnych narzędzi. Stosując jednak tylko checklisty lub SRR'y można przynajmniej dostać pogląd na to, co konkretnie w systemie stanowi zagrożenie, co z kolei pozwala stosunkowo niskim kosztem zwiększyć bezpieczeństwo poszczególnych jego komponentów.

Otwartą sprawą wydają się protokoły sieciowe. W jaki sposób je weryfikować, jeśli nie ma do nich wymagań? Jak zweryfikować wyższe warstwy korzystające ze znanych protokołów? Te pytania pozostają nadal otwarte, gdyż nie da się na nie udzielić prostej odpowiedzi. Wydaje się, że mogą one wyznaczyć pewne trendy i kierunki badań w przyszłości. Zagadnienie bezpieczeństwa protokołów sieciowych jest szczególnie bliskie autorowi. Wydaje się, że właśnie w tym kierunku należy prowadzić pracę i badania mające na celu wprowadzenie pewnych standardów wymaganych przez nowoczesne protokoły sieciowe.

Literatura

1. Dokumenty i skrypty ze strony organizacji IASE oraz DISA:
   http://iase.disa.mil/stigs/stig/index.html
   http://iase.disa.mil/stigs/checklist/index.html
   http://iase.disa.mil/stigs/SRR/index.html
2. Common Criteria portal:
   http://www.commoncriteriaportal.org/
3. Informacje o FIPS 140:
   http://csrc.nist.gov/publications/fips/
4. Marcin Kołodziejczyk - Applying of security mechanisms to low layers of OSI/ISO network model - Automatyka, wyd. AGH, Kraków 2010 (w druku, w języku angielskim)

5. Michael Howard, David LeBlanc, John Viega - The 19 Deadly Sins of Software Security – McGraw-Hill/Osborne, California 2005
6. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone - Handbook of Applied Cryptography - CRC Press 1996 (wersja online)
7. Marek R.Ogiela - Security of computer systems - Wydawnictwa AGH, Kraków 2002
8. The Top 10 Most Critical Internet Security Threats - (2000-2001 Archive) - http://www.sans.org/top20/2000/
9. Marcin Kołodziejczyk - Tablice tęczowe jako skuteczna optymalizacja algorytmu brute-force - Elektrotechnika i Elektronika, wyd. AGH, Kraków 2009/2010 (w druku)

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str.31-34

WYZNACZANIE STREF ODDZIAŁYWANIA POLA MAGNETYCZNEGO W GABINETACH STOSUJĄCYCH APARATURĘ DO MAGNETOTERAPII

Wojciech Kraszewski
Mikołaj Skowron
Przemysław Syrek

Akademia Górniczo-Hutnicza
Wydział Elektrotechniki Automatyki Informatyki i Elektroniki
al.Mickiewicza 30, 30-059 Kraków
e-mail: (wkraszew, mskowron, syrekp)@agh.edu.pl

Streszczenie: *Pola magnetyczne niskich częstotliwości stosowane są w leczeniu wielu schorzeń. Szczególną rolę odgrywają przy stymulacji zrostu kostnego. W przypadku każdego ze schorzeń, pole może osiągać wartości, które według polskich przepisów wymagają skracania czasu ekspozycji, co dotyczy personelu medycznego przebywającego w pobliżu aplikatora. Środowisko MatLab, ze względu na bardzo bogaty zestaw funkcji umożliwiających prezentację wyników, jest narzędziem bardzo użytecznym, umożliwiającym obliczenia rozkładu pola, jak i jego przestrzenną prezentację. Pozwala to ocenić rozkład pola w obrębie leczonych narządów, jak również podział otoczenia na odpowiednie strefy, ze względu na zagrożenie z nim związane.*

Słowa kluczowe: *Strefy ochronne, pole magnetyczne, magnetoterapia, MatLab*

Determination of zones due to influence of magnetic field in surgeries using magnetotherapy's instruments

Abstarct: *Low-frequency magnetic fields are used in therapy of many diseases. They have a special role in stimulation of adhesion of bone fractures. Healing of any illness involves magnetic field, that may reach values of a danger level, and its application, due to polish regulations, urges medical staff to shorten the time of exposition to field in surrounding of applicators. MatLab environment, for the sake of its very rich set of graphical functions, is very useful tool, and provides programmers the possibility to solve field problems, and to introduce its three-dimensional presentation. It allows user to evaluate distribution of field within treated organs, and to split surrounding into adequate zones, with respect to danger connected with field value.*

Keywords: *Protective zones, magnetic field, magnetoteraphy, MatLab*

1. WSTĘP

Przypadki pozytywnego oddziaływania prądu elektrycznego na zrastanie się złamań odkryto już w XIX wieku. Dynamiczny rozwój terapii oraz odkrywanie i opis zjawisk zachodzących w kościach, rozpoczął się w latach pięćdziesiątych ubiegłego stulecia. Urządzenia wspomagające leczenie kości, podzielić można na trzy grupy według metody działania: wykorzystujące prąd stały, przy użyciu wszczepianych elektrod; umieszczanie elektrod naskórnych i stosowanie prądu o częstotliwości rzędu kilku Hz (metoda pojemnościowa); zastosowanie prądu zmiennego, generowanego poprzez zmienne w czasie pole magnetyczne (metoda nieinwazyjna).

Pola magnetyczne stosowane są w ramach specjalizacji medycznej o nazwie medycyna fizykalna. Jedną z metod leczenia jest magnetoterapia, która stosowana jest w leczeniu szerokiej gamy schorzeń. Jednym z najczęstszych zastosowań jest wspomaganie leczenia złamań kończyn. Stosowane są pola magnetyczne o częstotliwości nieprzekraczającej 100 Hz i wartościach od 0,1 do 20 mT [3]; dla porównania,

ziemskie pole magnetyczne osiąga wartości od 30 do 60 μT (w zależności od szerokości geograficznej).

2. STREFY OCHRONNE

Oddziaływanie pola magnetycznego na pacjenta, w przypadku leczenia jest niezbędne, na tym polega istota leczenia, a negatywne oddziaływanie jest krótkotrwałe – od kilkunastu do kilkudziesięciu minut na dobę. Personel medyczny narażony jest na wspomniane oddziaływanie przez cały czas pracy. Należy zatem w każdym z gabinetów stosujących aparaturę do magnetoterapii odpowiednio określić usytuowanie przestrzenne miejsc przebywania personelu. W Polsce, przepisy i normy dotyczące bezpieczeństwa i higieny pracy, wyszczególniają możliwość oraz dopuszczalny czas przebywania w polu magnetycznym. Czas zależy od wartości natężenia pola (w publikacji przeliczono wartości natężenia i podawana jest wartość indukcji). Czas przebywania nie podlega ograniczeniom jedynie w strefie bezpiecznej (o indukcji poniżej 83 μT). W strefie ochronnej, przebywanie dozwolone jest pod warunkiem skracania czasu ekspozycji, strefa ta podzielona jest na trzy obszary [2]: pośredni (83-251 μT), zagrożenia (251 μT –2,5 mT) oraz niebezpieczny (powyżej 2,5 mT). Gdy stanowisko pracy znajduje się w jednej ze stref ochronnych, wówczas warunki ekspozycji powinny być okresowo kontrolowane, wraz z jednoczesnym wymogiem okresowych, specjalistycznych badań lekarskich dla pracowników.

3. WYZNACZANIE ROZKŁADU POLA MAGNETYCZNEGO

Ze względu na skomplikowany kształt cewek stosowanych w aplikatorach, oraz różne wymiary i rodzaje samych aplikatorów, pole magnetycznego może być wyznaczane jedynie numerycznie. Ponieważ przyjmuje się pewne uproszczenia dotyczące obszaru otaczającego aplikator (stała przenikalność magnetyczna w całym otoczeniu), a częstotliwości stosowane nie przekraczają 1 KHz, do wyznaczania pola stosuje się prawo Biota-Savarta [1]:

$$dB_{Pi} = \frac{\mu I}{4\pi} \frac{dl_i \times r_{Pi}}{|r_{Pi}|^3}$$

gdzie odpowiednio dB_{Pi} –wektor pola magnetycznego od elementarnego fragmentu cewki, d_{li} –elementarny odcinek cewki, r_{Pi} –odległość punktu P_i od środka odcinka l_i. Pole magnetyczne w punkcie P_i jest sumą (całką) z powyższego wyrażenia, po wszystkich elementach l_i, na które podzielono cewkę.

Ze względu na to, iż większość cewek zbudowana jest z wielu (kilkudziesięciu zwojów), krok dyskretyzacji (dl) należy dobierać w rozsądnych granicach. Uwzględniając realne wymiary przekrojów przewodów, a także fakt, iż w tym zagadnieniu, nie jest konieczne obliczanie pola w odległościach mniejszych niż 10 mm od samych przewodów (obudowa, zabezpieczenia), cewka może zostać opisana jako nieskończenie cienki przewód wiodący prąd. Odnosząc rozwiązanie numeryczne do analitycznego w przypadku prostych kształtów (np. cewka kołowa), błąd względny przy podziale okręgu na 100 równych elementów ma wartość poniżej 0,1%.

4. ŚRODOWISKO OBLICZENIOWE

Obliczenia wykonano w środowisku MatLab. Dzięki bardzo bogatym możliwościom prezentacji wyników(szczególnie w przypadkach 3D), możliwe jest zaprezentowanie wyników i przedstawienie rozkładu pola magnetycznego zarówno na leczonych częściach ciała, jak i na powierzchniach, na których oddziaływanie pola jest niepożądane.
Generowanie dowolnej powierzchni odbywa się za pomocą instrukcji:

```
Powierzchnia =surface(x,y,z);
```

gdzie macierze x, y, z –zawierają przygotowane odpowiednio punkty tworzące wybrana powierzchnię (kość, tułów itp.).

Dowolna cewka (lub kilka cewek) opisana jest jako macierz o wymiarach Nx3, gdzie N-liczba odcinków którymi została przybliżona, a każdy wiersz macierzy zawiera początek danego odcinka, będący zarazem końcem poprzedniego.

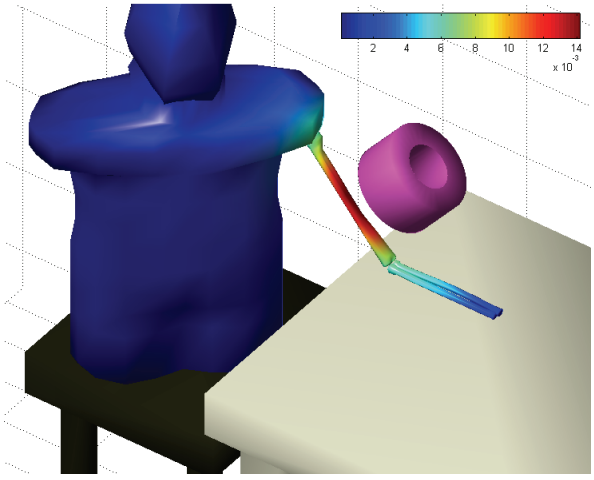
```
Cewka(i,:) =[rx ry rz];
```

Dodatkowo program zawiera kilka funkcji umożliwiających obracanie, przesuwanie powierzchni i cewek, a także ich skalowanie.
Poniżej przedstawiony został fragment, w którym obliczana jest wartość modułu pola magnetycznego w każdym z elementów, za pomocą których tworzona jest wybrana powierzchnia.

```
x=get(powierzchnia,'XData');
y=get(powierzchnia,'YData');
z=get(powierzchnia,'ZData');
for k=1:size(x,1)
for m=1:size(x,2)
   Point=[ x(k,m) y(k,m) z(k,m) ];
   B=[0 0 0];
   for i=1:size(cewka,1)-1
      dli = cewka(i+1,:)-cewka(i,:);
      rPi = Point - cewka(i,:);
      dB =cross(dli, rPi) /(rPi.^(3/2));
       B = B+dB;
    end
    B= mi*I /(4*pi) *B;
    Bmodul=sqrt(B(1)^2+B(2)^2+B(3)^2);
set(powierzchnia(k,m),'CData',Bmodul)
end
end
caxis([0 max(max(...
       get(powierzchnia('Cdata'))))])
```

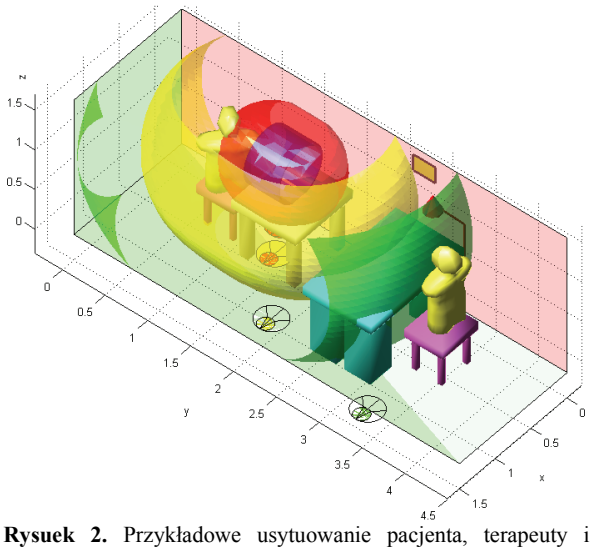
5. WYNIKI

Program umożliwia przedstawienie wyników obliczeń. Na rys.1, pokazano rozkład pola magnetycznego na powierzchni kości (ramienna, łokciowa, promieniowa) oraz na powierzchni tułowia i głowy. Maksymalna wartość osiągnęła 14 mT (na powierzchni kości ramiennej).



Rysuek 1 Rozkład pola magnetycznego na powierzchni kości i tułowia

Kolejna możliwość programu -wykreślanie powierzchni ekwipotencjalnych (isosurface), użyta została do przedstawiania kolejnych stref (wg norm); dzięki temu łatwo przedstawić kolejne powierzchnie, oddzielające poszczególne strefy, wyróżnione w normach dotyczących bezpieczeństwa (rys.2). W tej części oprócz pacjenta, przedstawiono również możliwe umiejscowienie stanowiska terapeuty.



Rysuek 2. Przykładowe usytuowanie pacjenta, terapeuty i aplikatora oraz powierzchnie oddzielające kolejno: strefę bezpieczną, pośrednią, zagrożenia i niebezpieczną

6. PODSUMOWANIE I WNIOSKI

Program umożliwia prezentację wyników, która jest łatwa w interpretacji. Na podstawie uzyskanych wyników, można ocenić wpływ zastosowanego aplikatora na rozkład pola również poza leczonym obszarem. Z rys.2 widać, że stanowisko pracy terapeuty znajduje się na granicy strefy bezpieczeństwa i strefy pośredniej (w odległości około 2 metrów od aparatury). Powierzchnie mają w przybliżeniu kształt elipsoidy, których dłuższy promień pokrywa się z osią aplikatora. W przypadku braku możliwości przeniesienia stanowiska terapeuty, konieczne jest obrócenie o 90 stopni (w płaszczyźnie 0xy) samego aplikatora, co zmiejszy wartość pola przy biurku terapeuty.

Literatura

1. Cieśla A.: Elektrotechnika : elektryczność i magnetyzm w przykładach i zadaniach, AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków 2006.
2. Rozporządzenie Ministra Pracy i Polityki Społecznej, W sprawie najwyższych dopuszczalnych stężeń i natężeń czynników szkodliwych dla zdrowia w środowisku pracy, Zał. 2/E: Pola i promieniowanie elektromagnetyczne z zakresu częstotliwości 0 Hz-300 GHz, Dziennik Ustaw nr 217, Warszawa, 29.11.2002
3. Sieroń A., i Inn..: Zastosowanie pól magnetycznych w medycynie. II wyd., α-medica press, Bielsko-Biała 2002

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str. 35-46

PRZEGLĄD I KLASYFIKACJA ZASTOSOWAŃ, METOD ORAZ TECHNIK EKSPLORACJI DANYCH

Marcin Mirończuk

Politechnika Białostocka
Wydział Elektryczny
ul.Wiejska 45E, 15-351 Białystok
e-mail: m.marcinmichal@gmail.com

Streszczenie: *Wzrost ilości danych jak i informacji w aktualnych systemach informacyjnych wymusił powstanie nowych procesów oraz technik i metod do ich składowania, przetwarzania oraz analizowania. Do analizy dużych zbiorów danych aktualnie wykorzystuje się osiągnięcia z obszaru analizy statystycznej oraz sztucznej inteligencji (ang. artificial intelligence). Dziedziny te wykorzystane w ramach procesu analizy dużych ilości danych stanowią rdzeń eksploracji danych. Aktualnie eksploracja danych pretenduje do stania się samodzielną metodą naukową wykorzystywaną do rozwiązywania problemów analizy informacji pochodzących m.in. z systemów ich zarządzania. W niniejszym artykule dokonano przeglądu i klasyfikacji zastosowań oraz metod i technik wykorzystywanych podczas procesu eksploracji danych. Dokonano w nim także omówienia aktualnych kierunków rozwoju i elementów składających się na tą młodą stosowaną dziedzinę nauki.*

Słowa kluczowe: *Eksploracja danych, metody eksploracji danych, techniki eksploracji danych, zastosowania eksploracji danych, ED*

Data mining review and use's classification, methods and techniques

Abstarct: *The large quantity of the data and information accumulated into actual information systems and their successive extension extorted the development of new processes, techniques and methods to their storing, processing and analysing. Currently the achievement from the statistical analyses and artificial intelligence area are use to the analysis process of the large data sets. These fields make up the core of data exploration – data mining. Currently the data mining aspires to independent scientific method which one uses to solving problems from range of information analysis comes from the data bases menagments systems. In this article was described review and use's classification, methods and techniques which they are using in the process of the data exploration. In this article also was described actual development direction and described elements which require this young applied discipline of the science.*

Keywords: *Data mining, data mining methods, data mining techniques, data mining classification, data minig review*

1. WSTĘP

Rozwój technologii umożliwia składowanie coraz to większych ilości danych. Zasoby gromadzonej informacji w istniejących systemach informatycznych (różnego rodzajów bazach danych) zwiększyły się na tyle, iż przestają się sprawdzać uprzednio wystarczające rozwiązania przeszukiwania, wydobywania i transformowania informacji w wiedzę lub inny rodzaj informacji. Z tego też względu w procesie tworzenia systemów informatycznych bardzo istotne staje się konstruowanie stosownych narzędzi informatycznych i metod umożliwiających odpowiednie przekształcanie danych w informację. Nowe metody wraz z odpowiednimi narzędziami powinny umożliwiać też wydobywanie informacji z baz danych, z możliwością łatwego transformowania ich do wiedzy.

Odpowiedzią na rosnące zapotrzebowanie dotyczące gromadzenia, przeszukiwania czy analizy ciągle powiększających się zbiorów informacji w platformach informatycznych stała się eksploracja danych (*ang. data mining – DM*) tzw. wydobywanie wiedzy z danych. Ze względu na odmienne podejścia środowisk naukowych i biznesowych do eksploracji danych (ED) powstały różne

definicje tego zagadnienia. Najczęściej wykorzystywanymi w publikacjach definicjami, które najlepiej ukazują aktualny pogląd na temat ED, są:

Definicja 1 Interdyscyplinarna stosowana metoda naukowa która wywodzi się z dziedziny nauki i techniki, jaką jest informatyka zajmująca się wizualizacją i wydobywaniem ukrytej „implicite" informacji z dużych zasobów informacyjnych (baz danych) [1]. Wykorzystuje w tym celu technologie przetwarzania informacji oparte o statystykę i sztuczną inteligencję: uczenie maszynowe, metody ewolucyjne, logikę rozmytą oraz zbiory przybliżone.

Definicja 2 Jeden z etapów procesu odkrywania wiedzy z baz danych, określany w skrócie jako KDD (*ang.* k*nowledge discovery in databases*). Termin ten został użyty w pierwszej pracowni KDD w 1989 roku w celu uwydatnienia tego, iż wiedza jest końcowym produktem odkrywania sterowanego danymi (*ang. data-driven discovery*). Odkrywanie wiedzy z baz danych jest wykorzystywane w takich naukach jak, sztuczna inteligencja (*ang. artificial inteligence – AI*) czy też uczenie maszynowe (*ang. machine learning*) [2, 3].

Definicja 3 Kompletna metodologia CRISP-DM (*ang.* c*ross-industry standard process for data mining*) opracowana przez trzy przedsiębiorstwa przemysłowe: SPSS (*ang.* s*tatistical package for the social science*), NCR (*ang. national cash register corporation*) oraz Daimler Chrysler. Metodologia ta dostarcza ujednolicony, elastyczny oraz kompletny model przeprowadzania procesu eksploracji danych w przedsiębiorstwach, niezależnie od ich specyfikacji [4-6].

Do definicji pierwszej należy odnosić się z dystansem ze względu na to, iż eksploracja danych jest młodą, dynamicznie rozwijającą się dyscypliną naukową. Natomiast definiowanie dyscypliny naukowej jest zawsze zadaniem kontrowersyjnym, ponieważ badacze często nie zgadzają się co do dokładnego zakresu i granic swojego obszaru badań [7]. Niemniej skala zastosowań eksploracji danych, stosowany aparat matematyczny i ilości publikacji odnoszących się do niej w opisach badań i analizy danych, pozwala na wstępne postrzeganie i definiowanie jej w ramach samodzielnej dyscypliny naukowej.

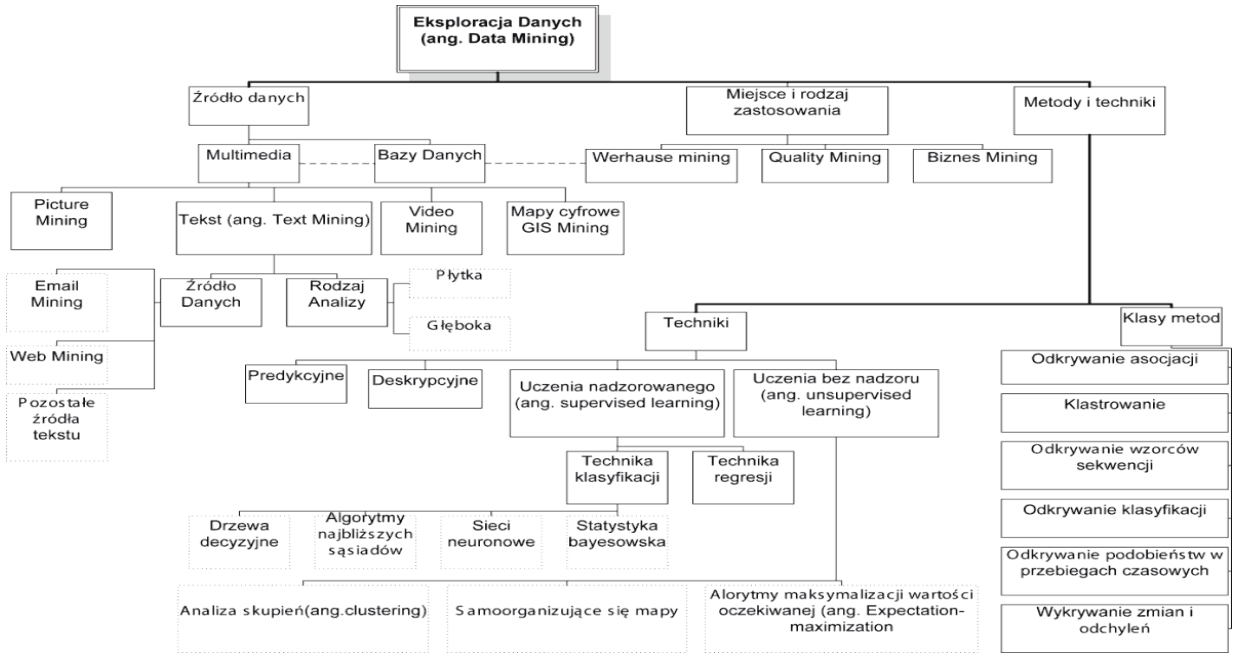
Przytoczone wyżej trzy definicje (*Def.1, Def.2, Def.3*) różnią się pod względem przyjmowanego punktu widzenia na temat eksploracji danych. Wspólną ich podstawę stanowi analiza zbiorów danych obserwowanych w celu znalezienia nieoczekiwanych związków i podsumowania danych w oryginalny sposób tak, aby były zarówno zrozumiałe, jak i przydatne w odpowiednich zastosowaniach [7]. Analiza ta zachodzi najczęściej w istniejących systemach informatycznych, w których zgromadzone informacje przekształcane są w inne rodzaje informacji przystosowanych do przeprowadzenia eksploracji danych (systemy preeksploracyjne). Analiza danych może także zachodzić w specjalnie skonstruowanych platformach informatycznych, przy konstruowaniu których odgórnie zakłada się potrzebę przeprowadzenia eksploracji danych (systemy posteksploracyjne).

W artykule skupiono się na opisie eksploracji danych jako całościowej dziedzinie nauki służącej do wydobywania wiedzy z systemów informacyjnych. Opis ED jako procesu można znaleźć w pracach [3, 8, 9]. W punkcie drugim niniejszego artykułu przedstawiono autorską klasyfikację eksploracji danych. Podział ten obejmuje trzy główne grupy: źródła danych, miejsce i rodzaj zastosowania oraz metody i techniki. W dalszych punktach rozwinięto i omówiono ww. trzy grupy. W punkcie trzecim przedstawiono odmiany eksploracji danych występujące w zależności od źródła danych jakie są do niej wykorzystywane. Punkt czwarty zawiera opis rodzaju badań jakie są wykonywane za pomocą ED oraz w jakich dziedzinach jest ona wykorzystywana. W punkcie piątym omówiono matematyczne podstawy całej dziedziny. Przedstawiono w nim podstawowe techniki i metody wykorzystywane do przeprowadzenia eksploracji danych.

2. PRZEGLĄD I KLASYFIKACJA ZASTOSOWAŃ, METOD ORAZ TECHNIK EKSPLORACJI DANYCH

Eksploracja danych jest dziedziną multidyscyplinarną, która skupia wokół siebie wiele dziedzin związanych z przechowywaniem, przetwarzaniem i analizowaniem informacji jak i wdrażaniem pozyskanej wiedzy w wyniku jej przeprowadzenia w zadany proces. Aktualnie termin eksploracja danych stosowany jest w różnych kontekstach, zależnych najczęściej od zastosowań czy tematu przeprowadzanych badań. Zastosowanie eksploracji danych w wielu różnych dziedzinach nauki jak i biznesu spowodowało powstanie wokół niej dużej liczby rozmaitej i nie do końca usystematyzowanej terminologii.

W punkcie tym przedstawiono autorską usystematyzowaną klasyfikację eksploracji danych i związane z nią pojęcia. Wynika ona z analizy dostępnej autorowi literatury, w której przedstawiono różne nie usystematyzowane pojęcia i zastosowania ED. Opracowaną klasyfikację prezentuje Rysunek. 1.



Rysunek. 1 Klasyfikacja eksploracji danych w zależności od rodzaju źródła danych, zastosowania oraz metod i technik [opracowanie własne]

Rysunek. 1 nie przedstawia kompletnej klasyfikacji ED ze względu na to, iż jest to w dalszym ciągu młoda interdyscyplinarna dziedzina w trakcie dynamicznego kształtowania [7, 8]. Niemniej da się już wyodrębnić podstawowe nazewnictwo związane z kontekstem jej użycia oraz metod i technik używanych do jej przeprowadzenia. Autorski podział obejmuje trzy podstawowe grupy według których można rozpatrywać ED. Są to: źródła danych (rozdział 3), miejsce i rodzaj zastosowań (rozdział 4) oraz metody i techniki (rozdział 5).

3. ŹRÓDŁA DANYCH DO EKSPLORACJI DANYCH

Źródła danych do eksploracji danych stanowią różnego rodzaje pliki płaskie oraz bazy danych i systemy ich zarządzania (systemy zarządzania bazą danych – SZBD) [10]. Ogólnie, eksploracja danych przeprowadzana jest na pewnego rodzaju informacjach przechowywanych w SZBD, na który składa się baza danych. W szczególności baza danych jak i system zarządzający nią może być dostosowany do składowania i operowania na danych multimedialnych. Z tego też względu wyróżnione zostały dwa główne źródła ED: bazy danych i szczególny ich przypadek, który stanowią multimedialne bazy danych.

3.1. Bazy danych

Bazy danych z systemami ich zarządzania a także bazy multimedialne z systemami ich zarządzania można podzielić na dwie kategorie, wyodrębnione pod względem zastosowania czasowego i przeznaczenia ED. Pierwszą kategorię stanowią systemy *preeksploracyjne*, w których najczęściej analiza danych zachodzi w istniejących systemach informatycznych, w których zgromadzone informacje przekształcane są w inne rodzaje informacji przystosowanych do przeprowadzenia eksploracji danych [8]. Drugą kategorię stanowią natomiast platformy *poseksploracyjne,* w których analiza danych może zachodzić w specjalnie skonstruowanych platformach. Przy ich budowie odgórnie zakłada się potrzebę przeprowadzenia eksploracji danych [8].

3.2. Multimedia i multimedialne bazy danych

Multimedia stanowią szczególny przypadek bazy danych, która wykorzystuje różne formy składowania

i przeszukiwania informacji w celu dostarczania odbiorcom wiedzy na ich temat. Multimedialne bazy danych i systemy ich zarządzania przystosowane zostały do pracy z takimi źródłami danych jak np.: tekst, dźwięk, grafika, wideo. Ze względu na to na jakim typie mediów dokonywana jest ED wydzielono osobne podgałęzie klasyfikacji obejmujące: eksplorację obrazów (podrozdział 3.3), eksplorację nagrań audio (podrozdział 3.4), eksplorację danych wideo (podrozdział 3.5), eksplorację danych w geograficznym systemie informacji (podrozdział 3.6) oraz eksplorację danych tekstowych (podrozdział 3.7).

3.3. Eksploracja obrazów

Eksploracja obrazów (*ang. image mining, ang. picture mining*) [11, 12] dotyczy wydobywania wiedzy poprzez odkrywanie relacji między obrazami, czy też wzorów ukrytych (niejawnie) występujących w obrazach oraz pomiędzy nimi. Dziedzina ta wykorzystuje metody pochodzące z: widzenia komputerowego (*ang. computer vision*), przetwarzania obrazu, odzyskiwania obrazu, eksploracji danych, uczenia maszynowego, baz danych i sztucznej inteligencji. W eksploracyjnej analizie obrazów wyróżnia się dwa podejścia. Pierwsze polega na odkrywaniu z dużych zbiorów obrazów ich pojedynczych egzemplarzy. Drugie natomiast polega na odkrywaniu połączeń między zbiorami obrazów oraz występujących między nimi asocjacji.

3.4. Eksploracja nagrań audio

Eksploracja nagrań audio (*ang. audio mining*) [12, 13] polega na przetwarzaniu i analizowaniu danych dźwiękowych. Zajmuje się ekstrakcją, przetwarzaniem oraz wydobywaniem wiedzy z modeli muzycznych. Podstawowym jej celem jest wyszukiwanie informacji muzycznej (*ang. music information retrieval - MIR*). Wiedza ta pozwala użytkownikom poszukiwać i odnajdywać muzykę przy pomocy zawartości bazującej na tekście (*ang. content-based text*) i pytaniach audio takich jak: zapytanie-przez-ogłuszający/śpiewający/grający/wyjątki lub poprzez specyfikację z wykazem muzycznych terminów takich jak "szczęśliwy", "energiczny" itd. oraz poprzez połączenie i kombinację obydwu rodzajów wyszukiwań. Efektem takiej eksploracji może być ranking odpowiedzi oparty na oszacowaniu podobieństw odnoszących się do powiązanych plików audio.

3.5. Eksploracja danych wideo (nagrań filmowych)

Eksplorację danych wideo (*ang. video mining*) [13-15] definiujemy jako nienadzorowane odkrywanie wzorów w zawartościach baz multimedialnych przechowujących audio-wizualne dane. Za pomocą eksploracyjnej analizy danych wideo istnieje możliwość odkrycia interesujących zarejestrowanych zdarzeń, które dostępne są *a priori*. Wyróżniamy trzy typy nagrań audio-wizualnych, które są poddawane analizie:
- wyprodukowane np. filmy, reportaże, dramaty,
- nieopracowane dane filmowe np. monitoring ruchu ulicznego, wideo z nadzoru,
- nagrania medyczne np. ultra dźwiękowe wideo zawierające echokardiografie.

Na wszystkich trzech grupach dokonywana jest analiza dotycząca:
- wykrywania przyczyn wywołanych zdarzeń np. pojazdów wjeżdżających na teren chroniony, ludzi wchodzących i wychodzących z chronionych budynków,
- określania typowych i nieprawidłowych wzorów działalności,
- klasyfikacji obserwowanej działalności do wybranej kategorii np. chodzenie, jeżdżenie rowerem,
- grupowania i określania interakcji pomiędzy jednostkami (obiektami).

Eksploracyjna analiza wideo nie jest tylko procesem, który automatycznie ekstrahuje (wydobywa, przetwarza) zawartość i strukturę nagrań wideo, cech przesuwającego się obiektu, przestrzenne lub temporalne (czasowe) korelacje tych cech. Nastawiona jest ona również na odkrywanie wzorców struktury wideo, aktywności obiektów, zdarzeń etc. z olbrzymich zbiorów danych wideo bez niewielkich założeń co do ich zawartości. Przy użyciu eksploracyjnych technik analizy wideo takich jak, podsumowania, klasyfikacja, raportowanie o zdarzeniach (*ang. event alarm*) implementowane są tzw. sprytne aplikacje video. Zasadniczą różnicą pomiędzy konwencjonalną eksploracją danych a eksploracją nagrań filmowych jest fakt, iż głęboka analiza wideo operuje na mocno nieustrukturyzowanych danych. Surowe (nieopracowane) dane wideo zawierają tylko piksele, nawet przetworzone dane wideo są również złożonymi typami posiadającymi rozłączne wymiary. Dlatego też konwencjonalne algorytmy eksploracji danych nie mogą zostać bezpośrednio zastosowane w tej grupie danych.

3.6. Eksploracja danych w geograficznym systemie informacji

Eksploracja danych w geograficznym systemie informacji (*ang. geographic information system mining – GIS Mining*) stanowi analizę danych przeprowadzaną w tzw. geograficznym systemie informacyjnym GIS, który

reprezentuje dane opisujące aspekty powierzchni ziemi takie jak np. drogi, domy etc. [16]. W sensie funkcjonalnym systemy GIS służą do analizy przestrzennej realizowanej poprzez [17-19]:
- predefiniowane w systemie raporty, zestawienia, wykresy wraz z wizualizacją przestrzenną,
- języki zapytań (np. *ang. standard query language – SQL*) do zintegrowanej graficznej i opisowej bazy,
- wyzwalacze (*ang. triggers*) używane w aktywnych systemach do ciągłego przetwarzania danych w wyniku ich uaktualnień.

Bazę graficzną tworzą cyfrowe mapy tematyczne (*ang. digital maps*), ortofotomapy, numeryczne modele terenu. Poszczególne obiekty bazy graficznej są połączone z bazą opisową, której zawartość wynika ze struktury fizycznej - wynikającej głównie z rodzaju i zakresu informacji zawartych w bazie danych [20-22]. System zarządzania taką bazą określa się jako system zarządzania bazą danych przestrzennych SZBDP (a*ng. spatial database systems – SDBS*). Systemy zarządzania bazami danych przestrzennych są to systemy do zarządzania ww. danymi przestrzennymi poprzez np. wyszukiwanie, składowanie, uaktualnianie [23, 24]. Ilość jak i wielkość dostępnych przestrzennych baz danych szybko rośnie. Z tego też względu zostają ograniczone ludzkie możliwości analizy danych w nich zebranych dotyczących takich zagadnień, jak: odkrywanie ukrytych regularności, reguł lub skupień ukrytych w danych [16]. W celu poszerzenia i umożliwienia analizy tak dużej ilości danych zebranych w multimedialnych przestrzennych bazach danych stosuje się podejście określane jako: eksploracyjne odkrywanie danych przestrzennych (*ang. spatial data mining*) lub odkrywanie wiedzy w przestrzennych bazach danych (*ang. knowladge discovery in spatial databases*) [25]. Podejścia te reprezentują szczególny przypadek odkrywania, gdyż pozwalają wydobyć relacje, które istnieją między przestrzennymi i nie przestrzennymi danymi i innymi charakterystycznymi danymi, które jawnie nie są zgromadzone w przestrzennych bazach danych [26, 27]. Podstawową różnicą pomiędzy odkrywaniem wiedzy z relacyjnych a przestrzennych baz danych jest to, iż atrybuty sąsiadów pewnego obiektu, którym jesteśmy zainteresowani, mogą wpływać na sam obiekt zainteresowania [24].

Typowymi zadaniami odkrywania wiedzy w przestrzennych bazach danych są np. klasteryzacja czy charakteryzacja przez detekcje trendów. Używa się ich w celu odnalezienia ukrytych *implicte* regularności, czy reguł bądź wzorców w danych przestrzennych. Do podstawowych klas metod używanych w przestrzennej eksploracji danych można zaliczyć [16, 24, 28, 29]:
- grupowanie przestrzenne (*ang. spatial clustering*) – polega na grupowaniu obiektów bazy danych do znaczących podklas (klastrów) w taki sposób, aby poszczególne obiekty klastra były jak najbardziej podobne do siebie i jak najbardziej różne od elementów pozostałych klastrów. Zastosowanie klastrowania (grupowania) w przestrzennych bazach danych używa się np. w tworzeniu katalogu tematycznych map w geograficznych systemach informacji poprzez grupowanie wektorów cech,
- detekcja trendów przestrzennych (*ang. spatial trend detection*) – trend może zostać zdefiniowany jako czasowy wzór występujący w kilku seriach danych np. alarmy w sieci lub występowanie nawrotów chorób. W przestrzennym systemie bazy danych przestrzenny trend definiowany jest jako wzór zmiany nieprzestrzennych atrybutów w sąsiedztwie kilku obiektów bazy danych np. „kiedy następuje przeniesienie się ludzi z miasta X, siła nabywcza spada”,
- klasyfikacja przestrzenna (*ang. spatial classification*) – zadaniem klasyfikacji jest przydzielenie obiektu do klasy ze zbioru dostępnych wyselekcjonowanych klas na podstawie wartości atrybutów obiektu. Przestrzenna klasyfikacja może zostać użyta do wyjaśnienia odchyleń pomiędzy teoretycznymi a odkrytymi trendami przestrzennymi,
- charakterystyka przestrzeni (*ang. spatial characterization*) – jej zadaniem jest odnalezienie zwięzłego opisu (uogólnienia na pewien temat) dla wybranego podzbioru bazy danych.

3.7. Eksploracja danych tekstowych

Eksploracja danych tekstowych (*ang. text mining lub text data mining*) jest to analiza tekstu polegająca na wykorzystaniu inteligentnych reguł z zakresu uczenia maszynowego, lingwistyki komputerowej zajmującej się analizą języka naturalnego (*ang. natural language processing – NLP*), metod statystycznych oraz technik m.in. z zakresu przeszukiwania i grupowania danych [30]. Wykorzystywana jest do pozyskiwania informacji (wiedzy) z dużych nieustrukturyzowanych zbiorów danych tekstowych [31, 32]. Grupę tą można podzielić na dwie dodatkowe podgrupy. Pierwsza uwzględnia rodzaj przeprowadzanej analizy na tekście stanowiącym źródło danych (podrozdział 3.8). Druga natomiast została wydzielona ze względu na typ danych, rodzaj dokumentów (podrozdział 3.9).

3.8. Eksploracja danych tekstowych ze względu na typ analizy

W eksploracyjnej analizie tekstu, przeprowadzanej na tekście stanowiącym źródło danych, dostępne są dwie metody przetwarzania tekstu: płytkie i głębokie. Pierwsza metoda dotycząca płytkiej analizy tekstu (*ang. shallow text processing – STP*), określa grupę działań na tekście, których efekt jest niepełny w stosunku do głębokiej analizy tekstu. Polegają one na rozpoznawaniu struktur tekstów nierekurencyjnych lub o ograniczonym poziomie rekurencji, które mogą być rozpoznane z dużym stopniem pewności. Struktury wymagające złożonej analizy wielu możliwych rozwiązań są pomijane lub analizowane częściowo. Analiza skierowana jest głównie na rozpoznawanie nazw własnych, wyrażeń rzeczownikowych, grup czasownikowych bez rozpoznawania ich wewnętrznej struktury i funkcji w zdaniu. Analiza dotyczy też głównie dużych zbiorów dokumentów tekstowych a nie pojedynczych dokumentów a także takich zagadnień jak m.in. klasyfikacja (kategoryzacja) dokumentów (*ang. document classification lub document categorization*) ich grupowania (*ang. dokument clustering*) i wyszukiwania z nich informacji (*ang. information retrieval – IR*) [33-35].

Druga metoda opiera się na tzw. głębokiej analizie tekstu (*ang. deep text processing – DTP*) i jest procesem komputerowej analizy lingwistycznej wszystkich możliwych interpretacji i relacji gramatycznych występujących w tekście naturalnym. Zazwyczaj jest bardzo złożona i z reguły dotyczy pojedynczego dokumentu. Pomija się wszelkie zależności statystyczne i stosuje się rozwiązania polegające na przetwarzaniu danych w oparciu o predefiniowane wzorce lub gramatyki [33, 36].

3.9. Eksploracja danych tekstowych ze względu na rodzaj dokumentów

Istnieje wiele źródeł danych nadających się do przeprowadzenia na nich eksploracyjnej analizy tekstu. Jedynym ich wymogiem jest to, aby informacja w nich była zakodowana w postaci znaków ASCI. Do źródeł danych w postaci dokumentów tekstowych, na których przeprowadzana jest tekstowa eksploracja danych, należą:
- wiadomości email (*ang. email mining*) – eksploracja tych wiadomości może być rozpatrywana jako specyfikacja badań z zakresu ogólno pojętej eksploracji tekstu nad internetowymi wiadomościami email [37-39]. Zasadniczymi cechami wyróżniającymi tę grupę od innych grup tekstowych są m.in.: wiadomości email są częściowo uporządkowane i posiadają zorganizowaną narzuconą przez standardy formę [40, 41], wiadomości tekstowe są znacznie krótsze od dokumentów, które podaje się zwykle analizie tekstu, emaile mogą zawierać treści na temat rozmaitych dyskusji na dowolne tematy. Fakt ten prowadzi do tego, że np. klasyfikacja poczty staje się bardziej trudna [38],
- dokumenty ogólnoświatowej multimedialnej sieci oprogramowania w internecie (*ang. world wide web wining – WEB Mining*) – technika wykorzystywana przy eksploracji tych danych ma na celu odkrywanie i uzyskiwanie przydatnych informacji, wiedzy i wzorców z dokumentów i usług Internetowych powszechnie określanych jako World Wide Web (WWW) [42, 43]. W obrębie tej techniki możemy wyróżnić trzy jej specjalizacje [44, 45]: eksploracja struktury dokumentów WWW (*ang. web structure mining – WSM*), eksploracja zawartości dokumentów WWW (*ang. web content mining*) i eksploracja użyteczności dokumentów WWW (*ang. web usage mining*). Eksploracja struktury dokumentów jest procesem, którego zadaniem jest wydobycie struktury informacji z sieci Web poprzez analizę hiperlinków tzn. linków wchodzących i wychodzących z dokumentu (strony, serwisu). Metoda ta wykorzystuje strukturę dokumentu, w którym strony jako węzły są połączone z innymi stronami za pomocą odnośników [46]. Algorytmami wykorzystywanymi do przeprowadzania WSM są m.in.: Hits (*ang. hyperlink-induces topic search*) i Page Rank (Google) [42]. Web Mining koncentruje się na dostarczaniu rozwiązań z zakresu [44, 47-51]: odnajdywania powiązanych informacji na podstawie np. analizy linków [52] i zawartości stron [53], tworzenia nowej wiedzy na podstawie informacji dostępnych na stronach Web, personalizowania i adaptowania stron Web, uczenia stron o zachowaniach klientów lub indywidualnych użytkowników np. na podstawie poruszania się użytkowników po portalu internetowym czy też segmentacji użytkowników danego serwisu i uzyskiwaniu informacji o ich położeniu geograficznym (przestrzenna natura Web Mining) [54]. Ponadto badania za pomocą Web Mining takich struktur WWW jak: blogi, fora, aukcje internetowe, obwieszczenia sklepowe mogą służyć do badania zmienności rynku [55] oraz wzmacniać tzw. analizę wokół klienta [56] poprzez wykrywanie odpowiednich grup społecznych [57], do których można zaadresować wybraną ofertę bądź ochrony go przed oszustwem poprzez zastosowanie np. odpowiednich algorytmów z zakresu badania reputacji uczestników aukcji *on-line* [58],
- pozostałe niesklasyfikowane dokumenty – jest to grupa uwzględniająca zbiór dokumentów niewymienionych i aktualnie nieobjętych klasyfikacją, które wraz z pojawianiem się nowych źródeł i form danych tekstowych czekają na sklasyfikowanie.

4. MIEJSCE I RODZAJ ZASTOSOWAŃ EKSPLORACJI DANYCH

Grupa ta obejmuje miejsca i sektory zastosowania technik eksploracji danych w przedsiębiorstwach zaprojektowanych według procesu planowania zasobów (*ang. enterprise resource planning – ERP*) i ze względu na ich zapotrzebowania. Nazwy poszczególnych podkategorii wywodzą się głownie od typu zastosowania i zapotrzebowania na ED. Można wyróżnić tutaj takie podgrupy, jak: eksploracja danych w biznesie (*ang. business mining*), która opisana została w podrozdziale 4.1; eksploracja danych na rzecz jakości (*ang. quality mining*), która opisana została w podrozdziale 4.2; eksploracja danych w hurtowniach danych (*ang. warehouse mining*), która opisana została w podrozdziale 4.3; eksploracja danych w przemyśle (*ang. industry mining*), która opisana została w podrozdziale 4.4 oraz pozostałe sektory i dziedziny zastosowań, które opisane zostały w podrozdziale 4.5.

4.1. Eksploracja danych w biznesie

Eksploracja danych w biznesie jest przeprowadzana w środowisku biznesowym, które w przeciwieństwie do organizacji *no-profit*, nastawione jest na przynoszenie zysków. W organizacjach przynoszących, jak i nieprzynoszących zysku, eksploracja może być realizowana na wszystkich możliwych poziomach organizacji wyrażonej według np. modelu planowania zasobów przedsiębiorstwa (*ang. enterprise resource planning – ERP*) stanowiącego rozwinięcie systemu planowania zasobów produkcyjnych drugiego rzędu (*ang. manufacturing resource planning - MRP II*) [59]. Platforma ERP ma charakter komponentowy i składa się z takich modułów, jak [60-63]: system zarządzania klientami (*ang. customer relationship management – CRM*), system realizacji produkcji (*ang. manufacturing execution system – MES*), system zarządzania zasobami ludzkimi (*ang. human resource management – HRM*), system zarządzania finansami (*ang. financial resource management – FRM*). Dodatkowymi modułami mogą być: system planowania zapotrzebowania materiałowego (*ang. material requirements planning – MRP*) i zapotrzebowania na sprzęt. Przedstawiona lista nie wyczerpuje wszystkich możliwych modułów systemu ERP, daje natomiast, poprzez wykorzystywane jednoznaczne, opisowe pojęcia, wyraźny podział na wewnętrzne jednostki w przedsiębiorstwie profilowane do realizacji jego konkretnych funkcji, procesów i reakcji na jego zapotrzebowania.

W obrębie platformy ERP można wyróżnić dodatkowy abstrakcyjny komponent inteligencji biznesowej nazywany także analizą biznesową (*ang. business intelligence – BI*) [56]. Termin analiza biznesowa ma szerokie znaczenie. Najbardziej ogólnie można przedstawić ją jako proces przekształcania danych w informacje, a informacji w wiedzę, która może być wykorzystana do zwiększenia konkurencyjności przedsiębiorstwa a w przypadku przedsiębiorstwa *no-profit* może przynieść np. poprawę i ulepszenie pewnych aspektów prowadzonych przez nią działań. W obrębie jednostki analizy biznesowej można wyróżnić takie moduły, jak: system powiadamiania kierownictwa (*ang. executive information systems – EIS*), systemy wspomagania decyzji SWD (*ang. decision support systems – DSS*), system wspomagania zarządzania (*ang. management information systems – MIS*), system informacji geograficznej (*ang. geographic information systems – GIS*). Jednostka analizy biznesowej jest platformą pośredniczącą w wymianie i dostarczaniu informacji/wiedzy między poszczególnymi elementami struktury ERP. Jego najniższa warstwa (warstwa danych) reprezentowana jest przez różnego rodzaju systemy zarządzania danymi. W szczególności warstwę danych w tego rodzaju systemach stanowią hurtownie danych. Dostęp do systemów zarządzania danymi i wydobywanie z nich informacji/wiedzy realizowane jest przy wykorzystaniu interfejsu dostępu, na który składają się: aplikacje lub serwer aplikacji przeznaczony do przetwarzania analitycznego, transakcyjnego lub eksploracyjnego.

4.2. Eksploracja danych na rzecz jakości

Eksploracja danych na rzecz jakości (*ang. quality mining, ang. quality control data mining – QC DM*) jest to eksploracja danych wykonywana na rzecz kontroli jakości i polega na zgłębianiu danych w sterowaniu jakością [64]. Od zwykłej eksploracji odróżnia się m.in. tym, iż występuje tutaj konieczność reagowania na zmiany w danych na bieżąco (*on-line*). Innym wyróżnikiem eksploracji danych na rzecz jakości jest konieczność stosowania metod typowych dla sterowania jakością takich jak, karty kontrolne, analiza zdolności procesu, planowanie doświadczenia itp. Dane mają tutaj także swoją specyfikę – pochodzą z procesów technologicznych automatyki przemysłowej. Przez to zazwyczaj uzyskuje się mnóstwo parametrów, które często nie mają żadnego wpływu na wytwarzany w danej chwili produkt, ale mogą być decydujące dla innego produktu. Najczęstszym modelem jaki uzyskuje się w QC DM jest model „czarna skrzynka" [65].

4.3. Eksploracja danych w hurtowniach danych

Hurtownie danych, nazywane także magazynami danych (*ang. data warehouse*), są bardzo dużymi bazami danych, w których gromadzone są dane pochodzące z wielu heterogonicznych źródeł np. innych scentralizowanych lub rozproszonych baz relacyjnych, relacyjno-obiektowych, obiektowych, przestrzennych oraz ze źródeł innych niż bazy danych takich, jak arkusze kalkulacyjne, pliki XML, zasoby WWW [66-68]. Dane zebrane w hurtowni opisują (reprezentują) pewien określony tematycznie wycinek modelowanej rzeczywistości. Po za tym są zintegrowane, zmienne w czasie i stanowią nieulotną kolekcję. Dlatego też hurtownie wykorzystuje się najczęściej do wspierania procesu wspomagania decyzji [68] i stanowią główną bazę SWD. Hurtownia danych jest zazwyczaj wydzieloną centralną (na dany obszar) bazą danych odizolowaną od baz operacyjnych o typie przetwarzania danych transakcyjnych na bieżąco (*ang. on-line transaction processing – OLTP*) a jej struktura i użyte do jej budowy narzędzia są zoptymalizowane pod kątem przetwarzania analitycznego (*ang. on-line analytical processing – OLAP*) [69] lub hybrydowego (OLAP w połączeniu z OLTP) [68].

Hurtownie danych wyróżnione zostały z tego względu iż agregują dane historyczne, które przed załadowaniem do nich są poddawane procesowi oczyszczania, transformacji i ładowania (*ang. extraction, transformation, loading – ETL*). Proces ten przyspiesza znacznie etap wstępnego przetwarzania danych w procesie KDD i CRISP-DM [2-5], który pochłania większość czasu przy przeprowadzaniu ED [70, 71]. Dlatego pod tym względem platformy te stanowią doskonałe środowisko do przeprowadzania zadań eksploracyjnych. Ponadto rozwija się szereg technik oraz metod (np. eksploracyjne zapytania ad-hoc rozszerzone o asocjacyjne reguły) jak i architektur systemów informatycznych (np. równoległa skalowalna infrastruktura OLAP i ED) łączących tradycyjne przetwarzanie analityczne z zawansowaną analityką w postaci eksploracji danych [72-75].

4.4. Eksploracja danych w przemyśle

Eksploracja danych w przemyśle (*ang. industry mining*) [76, 77] polega na wykorzystaniu dogłębnej analizy do rozwiązywania problemów pojawiających się w trakcie: produkcji przemysłowej jak i w etapach projektowych produkcji. Najczęściej firmy produkcyjne posiadają już systemy informatyczne dwojakiego rodzaju:

- pierwsze z nich związane są z ogólną obsługą, a więc z administracją, zarządzaniem, księgowością, zaopatrzeniem, zbytem itp. – system ERP,
- drugie związane są z obsługą procesów technologicznych, które rejestrowane są przez czujniki. Pochodząca z nich informacja gromadzona jest następnie w systemach bazodanowych np. relacyjnych, hurtowniach danych etc. wynikających z przyjętych założeń projektowych związanych z agregacją i przetwarzaniem danych.

Eksploracja danych w przemyśle skupiona jest zwłaszcza na drugim rodzaju systemów, w którym używa się jej do: projektowania i doskonalenia produktu (kontrola, poprawa jakości produktu poprzez kontrolę i poprawę procesu technologicznego), sterowania i optymalizacji procesu produkcyjnego, analiz reklamacji i niezawodności oraz bezpieczeństwa. Niemniej w pierwszym rodzaju systemu eksploracja może zostać zastosowana na etapie projektowania produktu oraz do polepszenia związków z klientami np. poprzez identyfikację ich potrzeb czy też prognozowania popytu na produkt. Cechami rozróżniającymi i wyróżniającymi eksplorację danych w przemyśle od pozostałych grup jest fakt, iż ma ona bliski związek ze statystycznym sterowaniem jakością procesów (SPC) i z eksploracją na rzecz jakości (podpunkt 4.2). Kolejnym wyróżnikiem jest to, iż procesy na linii technologicznej zachodzą w sposób ciągły (*on-line*) więc analiza eksploracyjna musi też zachodzić w sposób ciągły (*ang. on-line data mining*) [78] tak, aby można było reagować na zmiany w czasie rzeczywistym. Ponadto w przemyśle podczas analizy eksploracyjnej wykorzystuje się modele typu „czarna skrzynka" ze względu na bardzo szybkie zmiany produkcji – czas życia produktów i okres stosowania konkretnej technologii ciągle się zmienia.

4.5. Pozostałe

Spektrum aktualnie przeprowadzanych badań w różnych dziedzinach z zastosowaniem ogólnie pojętej eksploracji danych jest tak bogate i różnorodne, iż nie da się w jednym artykule przedstawić i opisać wszystkich możliwych kombinacji i zastosowań tej dziedziny. Lepszym rozwiązaniem zdaje się być skupienie na podstawach ED a mianowicie aparacie matematycznym opisanym w punkcie 5.

Do interesujących, bliżej niescharakteryzowanych w artykule badań, w których ogólno pojęta ED jest wykorzystywana należą takie dziedziny, jak: zabezpieczenia systemów informatycznych i wykrywanie w nich ataków sieciowych [79], medycyna [80, 81], badania naukowe np. dotyczące cząstek elementarnych w

akceleratorze CERN [82], matematyczne dowodzenie twierdzeń [83], ochrona publiczna [84, 85], systemy ekspertowe i wspomagania decyzji [86, 87], energetyka i prognozowanie na zapotrzebowanie energii elektrycznej [88] będące zagadnieniem predykcyjnym [89], telekomunikacja i badanie np. fałszywych alarmów [90, 91] czy też powszechnie opisywane oraz badane zastosowania ED w business mining (przedsiębiorstwach produkcyjnych) [92] a dokładniej w handlu elektronicznym (*ang. electronic commerce, e-commerce*) wykorzystującym web mining) [70, 71, 93, 94]. Szacowany udział procentowy wykorzystania eksploracji danych w różnych sektorach działalności przedsiębiorstw można znaleźć w opracowaniu [95].

5. TECHNIKI I METODY EKSPLORACJI DANYCH

Grupa technik i metod eksploracji danych jest najbardziej priorytatywna ze względu na to, iż zawiera matematyczne podstawy całej dziedziny, które umożliwiają fizyczną realizację algorytmów eksploracji [7] na rzecz badań w wybranej dziedzinie poprzez implementację aplikacyjną.

5.1. Techniki eksploracji danych

Techniki eksploracji można podzielić ogólnie na cztery równoległe kategorie, w skład których wchodzą: techniki predykcyjne (podrozdział 5.2), techniki deskrypcyjne (podrozdział 5.3), techniki uczenia nadzorowanego (podrozdział 5.4) i techniki uczenia bez nadzoru (podrozdział 5.5). Przedstawione kategorie nie są ścisłe tj. technika predykcyjna może posługiwać się technikami z zakresu uczenia nadzorowanego i na odwrót. A zatem mogą istnieć pewnego rodzaju permutacje technik w celu osiągnięcia wyznaczonego celu badań.

5.2. Techniki predykcyjne

Techniki predykcyjne, inaczej nazywane technikami lub modelami przewidywania (*ang. predictive techniques*), starają się na podstawie odkrytych wzorców dokonać uogólnienia i przewidywania wartości danej zmiennej. Pozwalają na przewidywanie wartości zmiennej wynikowej na podstawie wartości pozostałych zmiennych (badawczych lub przewidujących) [7, 96, 97]. Techniki te w SWD wykorzystywane są do przewidywania i szacowania np. zasobów (sprzętu/ludzi) do rozwiązywania postawionego problemu.

5.3. Techniki deskrypcyjne

Techniki deskrypcyjne, nazywane także technikami bądź modelami opisowymi (*ang. description techniques*), służą do formułowania uogólnień na temat badanych danych w celu uchwycenia ogólnych cech opisywanych obiektów oraz ich najważniejszych aspektów [7, 97]. Techniki te w SWD stosuje się do odkrywania grup i podgrup podobnych zdarzeń lub identyfikacji zdarzeń.

5.4. Techniki uczenia nadzorowanego

Techniki uczenia nadzorowanego (*ang. supervised learning*) wykorzystują zbiory danych w których każdy obiekt posiada etykietę przypisującą go do jednej z predefiniowanych klas. Na podstawie zbioru uczącego budowany jest model, za pomocą którego można odróżnić obiekty należące do różnych klas [7, 97]. Technikami z zakresu uczenia nadzorowanego są techniki klasyfikacji stosowane od 1984 roku, do których należą drzewa decyzyjne (1984 rok) [98], algorytmy najbliższych sąsiadów (1992 rok) [99], sieci neuronowe (1991 rok) [100], statystyka bayseyowska (klasyfikacja bayseyowska 1992 rok i sieć bayseyowska 1995 rok) [101], algorytmy maszyny wektorów wspierających SVM (*ang. support vector machine*, 1995 rok) [102] oraz techniki regresji [7].

5.5. Techniki uczenia bez nadzoru

W przypadku technik uczenia bez nadzoru (*ang. unsupervised learning*) brak jest etykiet obiektów, nie ma także zbioru uczącego. Techniki te starają się sformułować model (modele) wiedzy najlepiej pasujące do obserwowanych danych [96, 97]. Technikami z zakresu uczenia bez nadzoru są: techniki analizy skupień, klastrowania (*ang. clustering*) [103], samoorganizujące się mapy (*ang. self-organization map*) [104], algorytmy aproksymacji wartości oczekiwanej (*ang. expectation-maximization*) [105] czy też zbiory przybliżone [106].

5.6. Metody eksploracji danych

Metody eksploracji danych bazują na technikach i stanowią ich uogólnienie. Realizowane są za pomocą wybranej techniki przy użyciu odpowiedniego dla niej algorytmu eksploracji danych [7]. Do metod ED zaliczamy m.in.: odkrywanie asocjacji (podrozdział 5.7), klastrowanie (podrozdział 5.8), odkrywanie wzorców sekwencji reguł (podrozdział 5.9), odkrywanie klasyfikacji (podrozdział 5.10), odkrywanie podobieństw

w przebiegach czasowych (podrozdział 5.11) i wykrywanie zmian i odchyleń (podrozdział 5.12).

5.7. Metody odkrywania asocjacji

Pojęcie reguł asocjacyjnych (*ang. association rulet*) zostało po raz pierwszy wprowadzone w 1993 roku przez R. Agrawala, T. Imielinskiego, A. Swami [107]. Odkrywanie asocjacji (powiązań) polega na wykrywaniu różnego rodzaju zależności występujących między danymi w bazie danych. Precyzyjniej mówiąc zależności te określone są za pomocą korelacji reguł asocjacyjnych wiążących współwystępowanie podzbiorów elementów w dużej kolekcji zbiorów. Znalezione korelacje prezentowane są jako reguły postaci $X \Rightarrow Y$ (wsparcie, ufność), gdzie X i Y są rozłącznymi zbiorami elementów. Termin *wsparcie* oznacza częstotliwość występowania zbioru $X \cup Y$ w kolekcji zbiorów, zaś termin *ufność* określa prawdopodobieństwo warunkowe P(X|Y) [108, 109].

5.8. Metody klastrowania

Klastrowanie, nazywane także grupowaniem lub analizą skupień (*ang. clustering*), polega na znajdowaniu skończonych zbiorów klas obiektów (klastrów) w bazie danych posiadających podobne cechy. Podczas tego procesu zbiór obiektów dzielony jest na takie podzbiory aby jednocześnie maksymalizować podobieństwo między obiektami przypisanymi do tego samego podzbioru i minimalizować podobieństwo miedzy obiektami przypisanymi do różnych podzbiorów zgodnie z zadaną miarą podobieństwa między obiektami [97]. Podczas dokonywania klastrowania nie są znane docelowe podzbiory (grupy) obiektów oraz zazwyczaj nie jest znana ich liczba [108]. Z tego względu klastrowanie należy do tzw. klasyfikacji bez nadzoru i jest rozwiązywana za pomocą przeznaczonych do tego technik wymienionych w podpunkcie 5.5. Ponadto algorytmy przeznaczone do analizy skupień można podzielić na kilka podstawowych kategorii na które składają się [97, 108, 110, 111]: metody hierarchiczne (procedury aglomeracyjne i deglomeracyjne), grupy metod *k*-średnich (*ang. k-means*), metody rozmytej analizy skupień (*ang. fuzzy clustering*) oraz metody niechierarchiczne.

5.9. Metody odkrywania wzorców sekwencji reguł

Problem odkrywania wzorców sekwencji został po raz pierwszy sformułowany w 1995 roku przez niektórych twórców metody asocjacyjnej m.in. Rakesh Agrawal oraz Ramakrishnan Srikant. Sekwencję stanowi uporządkowany ciąg zbiorów elementów, w którym każdy zbiór posiada znacznik czasowy [108]. Wzorce sekwencji stanowią rozwinięcie modelu reguł asocjacyjnych o takie elementy, jak [97, 111]: następstwa zdarzeń, ograniczenia dotyczące maksymalnych interwałów czasowych między kolejnymi wystąpieniami elementów sekwencji. Wprowadzenie interwałów czasowych umożliwiło nakładanie pewnego rodzaju okien czasowych do filtrowania sekwencji. Odkrywanie wzorców sekwencji polega ogólnie na znalezieniu w bazie danych sekwencji, podsekwencji występujących częściej niż zadany przez użytkownika próg częstości, zwany progiem minimalnego wsparcia (*ang. minsup*) w pewnym przedziale czasu.

5.10. Metody odkrywania klasyfikacji

Klasyfikacja (*ang. classification*) polega na zbudowaniu modelu przypisującego nowy, wcześniej nie znany obiekt, do jednej ze zbioru predefiniowanych klas. Przypisanie to następuje na podstawie wcześniejszego uczenia klasyfikatora (modelu umożliwiającego takie przypisanie) na zbiorze uczącym [97]. Najczęściej stosowanymi technikami do klasyfikacji są: klasyfikacja bayesowska, adaptywna sieć Bayesa, algorytmy indukcyjnych drzew decyzyjnych, algorytm *k* najbliższych sąsiadów, sieci neuronowe czy też algorytm SVM [108].

5.11. Metody odkrywania podobieństw w przebiegach czasowych

Odkrywanie podobieństw w przebiegach czasowych polega na odnalezieniu punktów wspólnych opisujących grupę wyselekcjonowanych przebiegów opisujących zadany proces trwający ciągle w czasie [109].

5.12. Metody wykrywania zmian i odchyleń

Wykrywanie zmian i odchyleń polega na znajdowaniu różnic pomiędzy aktualnymi a oczekiwanymi wartościami danych. Wykorzystywane jest podczas znajdowania anomalnych tj. niepasujących do trendu danych które od niego odstępują [109].

5.13. Metody odkrywania cech

Odkrywanie cech wykorzystywane jest najczęściej we wstępnych procesach (*ang. preprocessing*) eksploracji danych [3] w celu zmniejszenia wymiarowości rozpatrywanego problemu a więc i zwiększenia efektywności metod eksploracji danych. W celu zmniejszenia wymiarowości problemu stosuje się tzw. wybór cech (*ang. feature selection*) i odkrywanie cech

(*ang. feature extraction*) czy też analizę składowych głównych (*ang. principal components analysis – PCA*). Pierwsza z metod polega na wyselekcjonowaniu z grupy tych atrybutów tylko które posiadają istotną wartość informacyjną. Dwie następne metody polegają na połączeniu atrybutów i stworzeniu ich liniowej kombinacji w celu zmniejszenia liczby wymiarów i uzyskania nowych składowych głównych [7, 108, 111, 112]. Wybór i generacja nowych atrybutów może odbywać się w sposób nadzorowany lub bez nadzoru [111].

6. WNIOSKI

Eksploracja danych, stanowiąca jeden z etapów procesu np. odkrywania wiedzy z baz danych czy też traktowana jako dziedzina nauki, niewątpliwie jest zagadnieniem interdyscyplinarnym. Na jej interdyscyplinarność ma wpływ nie tylko szerokie spektrum jej aktualnych, opisanych w artykule, zastosowań ale także bogaty aparat matematyczny zaczerpnięty z różnych dziedzin nauki w celu pozyskiwania wiedzy z ogromnych zbiorów danych, które zazwyczaj są tylko częściowo ustrukturyzowane bądź wcale. Niezależnie od rodzaju danych, na których przeprowadzana jest eksploracja, wymagany jest zawsze dodatkowy nakład na skonstruowanie i opisanie samego celu badania jak i na określenie metody a następnie techniki oraz procesu do jego zrealizowania.

Skonstruowana i opisana klasyfikacja pozwala w łatwy sposób odnaleźć, umiejscowić i opisać własne badania w szerszym kontekście ED oraz umożliwia w łatwy sposób odnalezienie potrzebnej metody i techniki do ich realizacji. Ponadto przedstawiona klasyfikacja dostarcza początkowego usystematyzowanego słownika pojęć związanych z eksploracją danych, który w łatwy sposób można rozszerzać poprzez uzupełnianie go (odpowiednich gałęzi klasyfikacji) o własne definicje pojęć na danym polu zastosowań i badań naukowych.

Niektóre źródła danych mogą wymagać specyficznych metod oraz technik do przeprowadzenia na nich eksploracji danych. Wszystkie one mogą zostać dodane do wybranych gałęzi klasyfikacji według danych a następnie mogą zostać powiązane z odpowiednimi metodami oraz technikami przeprowadzania na nich eksploracji danych. Użycie takiego podejścia umożliwia więc kompleksowe, systematyczne i elastyczne klasyfikowanie nowych zastosowań i powstających w ich obrębie pojęć z zakresu eksploracji danych.

Literatura

[1] Wilk-Kołodziejczyk D. Pozyskiwanie wiedzy w sieciach komputerowych z rozproszonych źródeł informacji. In: Lesław H.H., editor. *Społeczeństwo informacyjne Wizja czy rzeczywistość?* [on-line] Kraków: Uczelniane Wydawnictwa Naukowo - Dydaktyczne, 2003, 30 maja. [dostęp: 16 listopada 2007] Dostępny w Internecie: http://winntbg.bg.agh.edu.pl/skrypty2/0095/285-295.pdf.

[2] Piatetsky-Shapiro G and Frawley JW. Knowledge Discovery in Databases. AAAI/MIT Press, 1991.

[3] Fayyad U, Piatetsky-Shapiro G and Smyth P. From Data Mining to Knowledge Discovery in Databases. AI Magazine, 1996.

[4] Chapman P, *Clinton* J, Kerber R, Khabaza T, Reinartz T, Shearer C, et al. *CRISP-DM 1.0* Step-by-step data mining guide. [on-line]. [dostęp: 1 czerwca 2008] Dostępny w Internecie: http://www.crisp-dm.org/CRISPWP-0800.pdf.

[5] *CRISP-DM*. [on-line] [dostęp: 1 czerwca 2008] Dostępny w Internecie: http://www.crisp-dm.org/.

[6] Metodologia Data Mining - model referencyjny CRISP-DM. [on-line] [dostęp: 1 czerwca 2008] Dostępny w Internecie: http://www.spss.pl/konsulting/konsulting_datamining_metodologia.html.

[7] Hand D, Mannila H and Smith P. *Eksploracja danych*. Wydanie 1. Warszawa: Wydawnictwo Naukowo-Techniczne, 2005.

[8] Mirończuk M. Eksploracja Danych w kontekscie procesu Knowledge Discovery In Databases (KDD) i metodologii Cross-Industry Standard Process for Data Mining (CRISP-DM). 2009.

[9] Fayyad UM, G Piatetsky-Shapiro and Smyth P. From Data Mining to Knowledge Discovery: An Overview. AAAI Press/MIT Press, s. 1-36.

[10] Krasuski A and Maciak T. Historia rozwoju Systemów zarządzania bazami danych. Bezpieczeństwo i Technika Pożarnicza: Wydawnictwo CNBOP Józefów, 2006. p. 213-226.

[11] Stanchev P. Using Image Mining For Image Retrieval. 2003. Dostępny w Internecie: http://paws.kettering.edu/~pstanche/mexico.pdf.

[12] Kotsiantis S, Kanellopoulos D and Pintelas P. Multimedia mining. WSEAS Transactions on Systems, No 3, 2004, s. 3263-3268.

[13] Leman M, Clarisse LP, Baets BD, Meyer HD, Lesaffre M, Martens JP, et al. Tendencies, Perspectives, and Opportunities of Musical Audio-Mining. 2002. [dostęp: 15 września 2009] Dostępny w Internecie: http://www.sea-acustica.es/Sevilla02/mus01002.pdf.

[14] Dai K, Zhang J and Li G. Video Mining: Concepts, Approaches and Applications. [Beijing]: Multi-Media Modelling Conference Proceedings, 2006 12th International, 2006.
[15] Divakaran A, Miyahara K, Peker KA, Radhakrishnan R and Xiong Z. Video Mining Using Combinations of Unsupervised and Supervised Learning Techniques. SPIE Conference on Storage and Retrieval for Multimedia Databases, 2004. p. 235-243.
[16] Ester M, Kriegel H-P and Sander J. Spatial Data Mining: A Database Approach. Springer, 1997.
[17] Woźniak J and Ferenc J. Budowa systemów geoinformacyjnych w zakładach górniczych. Prace Naukowe Instytutu Górnictwa Politechniki Wrocławskiej, No 106, 2004, s. 225-232
[18] Agrawal R and Psaila G. Active Data Mining.
[19] Wang W, Yang J and Muntz R. An Approach to Active Spatial Data Mining Based on Statistical Information.
[20] Górniak-Zimroz J, Woźniak J and Zimroz R. Możliwości metod data mining w geograficznych systemach informacyjnych zorientowanych na zarządzanie zasobami ziemi. Prace Naukowe Instytutu Górnictwa Politechniki Wrocławskiej No 113, 2005, s. 75-86.
[21] Gramacki J and Gramacki A. Dane przestrzenne w bazach relacyjnych. Wykorzystanie danych przestrzennych, systemy zarządzania danymi przestrzennymi.
[22] Gramacki J and Gramacki A. Dane przestrzenne w bazach relacyjnych. Model danych, zapytania przestrzenne.
[23] Gueting RH. An Introduction to Spatial Database Systems. Special Issue on Spatial Database Systems of the VLDB Journal, No 3, 1994.
[24] Ester M, Kriegel H-P and Sander J. Knowledge Discovery in Spatial Databases. [Bonn Germany]: Invited Paper at 23rd German Conf on Artificial Intelligence (KI '99), 1999.
[25] Santos M and Amaral L. Knowledge Discovery in Spatial Databases through Qualitative Spatial Reasoning. Portugal, 2000. [dostęp: 5 maja 2009] Dostępny w Internecie: http://repositorium.sdum.uminho.pt/bitstream/1822/5584/1/PADD2000_MS_LA.pdf.
.

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str. 47-50

ALGORYTM PSZCZELI W OPTYMALIZACJI MODELU PRZEPŁYWOWEGO SZEREGOWANIA ZADAŃ

Wiesław Popielarski

Akademia Górniczo-Hutnicza im. St. Staszica
Doktorant III roku
Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Al. Mickiewicza 30, 30-059 Kraków
e-mail: wieslaw.popielarski@sabre.com

Streszczenie: *Problem optymalizacji przy ograniczonych zasobach jest jednym z podstawowych tematów w informatyce. Algorytm pszczeli z szerszej grupy algorytmów stadnych, wynaleziony i przedstawiony w połowie ostatniej dekady, wydaje się być obiecującym narzędziem w optymalizacji kombinatorycznej. Artykuł przedstawia wyniki badań nad algorytmem w optymalizacji modelu przepływowego szeregowania zadań i zakreśla dalsze ich obszary.*

Słowa kluczowe: *Algorytm pszczeli, model przepływowy, szeregowanie zadań*

Bees algorithm in optimization of task scheduling for flow shop model

Abstarct: *Problem of optimization with limited resources is fundamental one in computer sciences. The bees algorithm from wider group of swarm algorithms, invented and implemented about 2005, seems to be a good candidate for next useful tool for combinatorial optimization. Article presents the results of the bees algorithm research in flow shop model of task scheduling and outlines areas of further exploration.*
Keywords: *Bees Algorithm, Flow Shop, Task Scheduling*

1. WPROWADZENIE

Dla wielu problemów algorytmicznych, które generalnie zaliczane są do klasy NP, nie spodziewamy się znaleźć rozwiązań, które dawałyby poprawne odpowiedzi w czasie wielomianowym dla dowolnego zbioru wejściowego. Dobrym przykładem mocnego algorytmu, który ma taką własność, jest metoda sympleks programowania liniowego. Z reguły jest on w stanie rozwiązać problemy w czasie wielomianowym jednakże możemy skonstruować takie zbiory danych, dla których to narzędzie będzie potrzebować wykładniczej ilości kroków, by podać odpowiedź. Ze względu na to, że programowanie liniowe jest algorytmem deterministycznym, odpowiedź zawsze będzie optymalna (pod warunkiem, że rozwiązanie optymalne istnieje, patrz „Podstawowe twierdzenie programowania liniowego" w [6]).

Zanim przejdziemy do głównego tematu artykułu, rozważmy przez chwilę problem zbioru danych wejściowych. Bardzo interesującą i sprzyjającą okolicznością jest fakt, że statystycznie zbiory niekorzystne, a więc nie dające się rozwiązać w czasie wielomianowym, występują stosunkowo rzadko. Oczywiście tylko wtedy, gdy rozwiązywany problem nie dotyczy albo nie ogranicza się do tych szczególnych danych. Co więcej, daje to możliwość zastosowania nie tylko algorytmów deterministycznych, ale także stochastycznych i chociaż w tym drugim przypadku znalezione rozwiązanie nie musi być rozwiązaniem optymalnym, to jest ono z reguły na tyle bliskie, że popełniany błąd jest kompensowany przez ilość wykonywanych kroków, dużo mniejszą jak w przypadku algorytmów deterministycznych.
Jedną z klas algorytmów stochastycznych są algorytmy stadne, do których zalicza się przedstawiany algorytm pszczeli. Jak można domniemywać, algorytm pszczeli jest procedurą przeszukiwania przestrzeni rozwiązań

naśladującą sposób poszukiwania i pozyskiwania pokarmu przez pszczoły. Pierwsze implementacje pojawiły się w drugiej połowie ostatniej dekady , a więc stosunkowo niedawno, i nadal są przedmiotem prac badawczych.

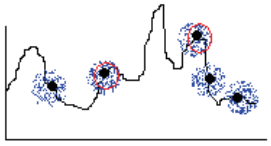
2. ALGORYTM PSZCZELI - STRUKTURA

Entomologia daje nam pewne wyobrażenie o strategii pozyskiwania pokarmu przez pszczoły. Początkowo część robotnic (zwiadowcy) poszukuje źródeł pokarmu wybierając w sposób losowy kierunek i obszar poszukiwań. Po znalezieniu potencjalnego źródła pożywienia, wraca do ula, by informacje o kierunku, odległości, obszarze i zasobności źródła pokarmu przekazać innym pszczołom w specyficznym rodzaju tańca (ang. waggle dance). Pozostałe pszczoły oceniają taniec zwiadowców i decydują się lub też nie na przyłączenie się do najlepiej tańczącego spośród nich. Generalnie im taniec jest dłuższy i bardziej ekspresyjny, tym więcej pszczół podąży za zwiadowcą. Zrobią to również te, które aktualnie eksploatują inne źródło, a które uznają, że nowo odnaleziony obszar rokuje na większą ilość pokarmu i efektywniejszy jego zbiór. W ten sposób kolonie pszczół rozwiązują problem maksymalizowania zbiorów przy ograniczonych zasobach.

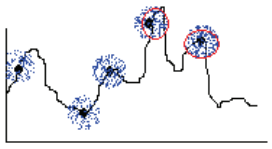
Algorytm pszczeli jest algorytmem iteracyjnym i z reguły jedynym kryterium stopu jest liczba iteracji podobnie, jak w przypadku innych algorytmów losowych. Wynika to przynajmniej z dwóch zasadniczych powodów. Po pierwsze, algorytm z reguły nie przeszukuje całej przestrzeni rozwiązań, a tylko jej część, a ponadto przeważnie ze względu na koszt, obliczane są jedynie pomocnicze wskaźniki jakości, które nie niosą pełnej informacji o zachowaniu się wskaźnika głównego. Również jak każdy algorytm losowy (genetyczny, symulowane wyżarzanie) algorytm pszczeli może być podatny na problem przedwczesnej zbieżności w zależności od przyjętej strategii przeszukiwania. Ogólne działanie algorytmu można opisać w następujących krokach:

1. Inicjalizacja populacji początkowej – wstępne określenie parametrów algorytmu takich jak wielkość populacji – *n*, liczba wyznaczonych rokujących obszarów w iteracji – *m*, liczba najlepszych rozwiązań w iteracji – *e* oraz inne; określenie wskaźnika jakości; przypisanie losowo wybranych obszarów przestrzeni rozwiązań poszczególnym zwiadowcom;
2. Ocena jakości znalezionych rozwiązań – wyznaczenie wskaźnika jakości dla każdej wartości znalezionej przez zwiadowcę; wskaźnik określa, czy zwiadowca odnalazł rokujący obszar; *m* najlepiej rokujących obszarów przechodzi do następnego kroku algorytmu;
3. Eksploracja najlepszych *m* rozwiązań – dla oznaczonych *e* spośród *m* rozwiązań przydzielamy większą ilość osobników do przeszukiwania otoczenia;
4. Wybór najlepszych *e* rozwiązań spośród *m* przeszukiwanych – rozwiązania te biorą udział w dalszej części algorytmu; tych *e* rozwiązań może się różnić od *e* początkowo znalezionych;
5. Eksploracja przestrzeni niezatrudnionymi osobnikami – rozlosowanie obszarów niezatrudnionym osobnikom;
6. Ocena jakości znalezionych rozwiązań - krok identyczny z krokiem 2
7. Iteracje – jeśli spełniono warunek stopu – zakończ; w przeciwnym wypadku skocz do kroku 3;

Na rysunku 1 pokazano przykład przeszukiwanego obszaru rozwiązań z zaznaczoną interpretacją wartości *m* oraz *e*.



a) punkty - reprezentują wylosowane wartości początkowe, chmury - przeszukiwane sąsiedztwo punktów, koła - najlepsze sąsiedztwa



b) trzy najgorsze obszary zostały zastąpione nowo wylosowanymi, zmieniły się też najlepsze sąsiedztwa - pozostało jedno z poprzedniej iteracji, drugie zostało znalezione w bieżącej

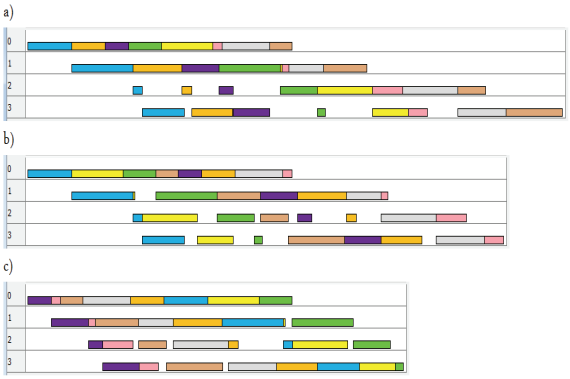
Rysunek. 1 Przykład przeszukiwanego obszaru dla *m=5* i *e=2*.

3. OPTYMALIZACJA MODELU PRZEPŁYWOWEGO

W modelu przepływowym *F* problemu szeregowania zadań (ang. flow shop) *n* zadań (ang. task) jest przetwarzanych na *m* maszynach. Każde zadanie *k* dzieli się na *m* prac (ang. job), z których każda wykonywana jest na innej maszynie. Stąd też każde z *n* zadań przetwarzane jest przez wszystkie maszyny dokładnie jeden raz. Istotnym założeniem dla modelu przepływowego jest kolejność maszyn, która jest taka sama dla wszystkich przetwarzanych zadań. W tym samym czasie maszyna może przetwarzać co najwyżej jedno zadanie oraz jedno zadanie może być przetwarzane przez co najwyżej jedną maszynę. Zadanie może być przetwarzane na następnej maszynie pod warunkiem, że jej aktualna praca została zakończona na maszynie bieżącej. W modelu dane są czasy t_{ij} wykonywania pracy zadania *j* na maszynie *i*. Optymalizacja polega na znalezieniu takiej permutacji zadań, dla której czas zakończenia ostatniego zadania na ostatniej maszynie jest najmniejszy. Czas ten oznaczany jest przez C_{max}. Równoważne jest to stwierdzeniu, że poszukiwana jest taka permutacja zadań, dla której sumaryczny czas bezczynności maszyn w oczekiwaniu na kolejne zadania jest minimalny.

Przykładowo zadanie *T1* wykonywane na czterech procesorach można zapisać jako: $T1=(j_0\ j_1\ j_2\ j_3)$, gdzie j_k oznacza pracę wykonywaną na procesorze *k*. Prace te wykonywane są zawsze w takiej samej kolejności, bez względu pozycję zadania *T1*. Stąd też czasy przetwarzania prac *j* mogłyby przyjmować wartości $t_{00}=a$, $t_{11}=a$, $t_{22}=a$, $t_{33}=a$, zakładając, że kolejność maszyn określona jest przez ciąg 0-1-2-3. Celem optymalizacji jest znalezienie takiej permutacji zadań *T1,…,Tn*, dla której całkowity czas przetwarzania będzie najkrótszy ([2] i [4]).

Rysunek 2 przedstawia przykłady trzech permutacji (uszeregowań) ośmiu zadań wykonywanych na czterech maszynach (stąd też każde zadanie składa się z czterech prac). Kolejność maszyn jest ustalona i określona w porządku 0-1-2-3 tak więc, każde zadanie najpierw musi być przetworzone przez maszynę 0, następnie 1, 2 i w końcu 3. Jeżeli przyjąć, że rysunek 2a) przedstawia uszeregowane zadania w kolejności 1-2-3-4-5-6-7-8, a rysunek 2c) przedstawia permutację optymalną, to kolejność optymalna wynosi 3-6-8-7-2-1-5-4.



Rysuek. 2 Przykładowe realizacje ośmiu zadań na czterech maszynach. Jednym kolorem oznaczono prace należące do tego samego zadania wykonywane na kolejnych maszynach (może zaskakiwać brak pracy zadania 5 oznaczonego kolorem żółtym na procesorze 1, w rzeczywistości czas przetwarzania jest proporcjonalnie niewielki, ale można też przyjąć bez straty ogólności, że wynosi on zero).

Zaprezentowany algorytm do optymalizacji problemu szeregowania zadań jest wzorowany na algorytmie ABC ([1]). Pierwszym jego krokiem jest znalezienie rozwiązań początkowych, będących z reguły wynikiem przeszukiwania obszarów rozwiązań wyznaczonych w sposób losowy. Sposób przeszukiwania pojedynczego obszaru wokół wyznaczonej losowo wartości określony jest przez sposób zdefiniowania sąsiedztwa. W problemach kombinatorycznych często określa się dwie permutacje jako sąsiednie, jeżeli różnią się między sobą pozycją dwóch sąsiednich elementów. Stąd przykładowo permutacje 1-**2**-**3**-4 i 1-**3**-**2**-4 są sąsiednie a **1**-2-**3**-4 i **3**-2-**1**-4 już nie. Następnie dla każdej sąsiedniej permutacji wyliczany jest wskaźnik jakości. Jeżeli jego wartość jest większa od obliczonej dla początkowo wyznaczonej, to permutacja ta staje się permutacją bazową (początkową) dla dalszych poszukiwań. Dodatkowo, aby obniżyć koszt wyznaczenia C_{max} sąsiedztwa zastosowano akcelerator [5]. Jeżeli w danym obszarze nie obserwuje się polepszenia wskaźnika, to poszukiwania w nim są przerywane i wyznaczany jest nowy. Kolejną ważną cechą algorytmu jest spamiętywanie testowanych permutacji, w celu uniknięcia ponownego ich wylosowania lub wyznaczenia. Przykładowe wyniki dla 20 maszyn, 500 zadań i 500 iteracji zestawiono w tabeli 1 (implementacja jest rozwinięciem pracy [3]). Czasy przetwarzania prac zostały wygenerowane losowo.

Przykład	Uruchomienie			Optimum	Max. błąd wzgl.
	1	2	3		
1	26429	26469	26511	26189	1.23%
2	26913	26903	26924	26629	1.11%
3	26739	26759	26764	26458	1.16%
4	26780	26750	26679	26549	0.87%
5	26628	26573	26587	26404	0.85%

Tabela. 1 Przykładowe wyniki zaimplementowanego algorytmu.

4. PODSUMOWANIE, DALSZE OBSZARY BADAWCZE

Użycie algorytmu pszczelego do znajdowania optymalnego rozwiązania problemu kombinatorycznego, jakim jest szeregowanie zadań (w szczególności model przepływowy) daje bardzo interesujące i obiecujące wyniki. Jednakże nie należy przy tym zapominać, że czasy poszczególnych prac wchodzących w skład zadań były generowane losowo. Z tego względu dalsze badania powinny skupić się na następujących problemach:

1. Analizy trudnych przypadków danych wejściowych; Należałoby zbudować zbiór takich danych i przeprowadzić testy dla tych danych;
2. Analizy i propozycji wyznaczania sąsiedztwa w inny sposób, jak domyślnie użyty w algorytmie;
3. Analizy i propozycji innych wskaźników jakości (funkcji dopasowania);
4. Analizy warunków początkowych i ich wpływu na znajdywane rozwiązania i ich zbieżność;
5. Analizy wartości początkowych liczności populacji, liczby rozwiązań oznaczonych m i najlepszych spośród nich e, i innych parametrów oraz ich zależności od wielkości i specyfiki zbioru danych wejściowych;
6. Analizy liczby uruchomień i weryfikacji statystycznej średniego błędu popełnianego przez algorytm;

Rozwiązanie tych zagadnień pozwoliłoby na określenie, czy algorytm pszczeli może stać się interesującą alternatywą dla innych algorytmów losowych, w tym genetycznych. Być może również dałoby się określić klasę bądź klasy problemów i powiązanych z nimi zbiorami danych wejściowych, dla których model kolonii pszczół dawałby najlepsze wyniki w ograniczonej liczbie kroków.

Literatura

1. Baykasoglu A., Ozbakor L., Tapkan P., „Artificial Bee Colony and Its Application to Generalized Assignment Problem", I-Tech Education and Publishing, Vienna, pp 29-33, 2007
2. Błazewicz J., Ecker K. H., Schmidt G., Węglarz J., „Scheduling in Computer and Manufacturing Systems", Springer-Verlag, Berlin, 1994
3. Librant S., Skrabacz M., „Zastosowanie algorytmu pszczelego w optymalizacji kombinatorycznej", Państwowa Wyższa Szkoła Zawodowa w Tarnowie, Instytut Politechniczny, praca inżynierska, Tarnów, 2009
4. Pinedo M., "Scheduling – theory, algorithms and systems", Pearson Education, Hong Kong, 2002
5. Smutnicki C., „Algorytmy szeregowania", Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2002
6. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., „Wprowadzenie do algorytmów", WNT, Warszawa, 2001

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str. 51-54

ZASTOSOWANIE ALGORYTMÓW GENETYCZNYCH DO PROJEKTOWANIA ROZDRABNIACZA WIELOTARCZOWEGO

Roksana Rama

Uniwersytet Technologiczno – Przyrodniczy w Bydgoszczy
Instytut Eksploatacji Maszyn i Transportu
ul.Kaliskiego 7/ p318,budynek 2.5, 85-789 Bydgoszcz
e-mail: roxi@utp.edu.pl

Streszczenie: *Zakres pracy obejmuje wiadomości z zakresu algorytmów genetycznych oraz rozdrabniaczy wielotarczowych. Przedstawia funkcjonowanie algorytmu genetycznego, metody optymalizacji oraz cel badań wykorzystania algorytmów genetycznych w projektowaniu rozdrabniaczy wielotarczowych. Program TarczeAG napisano w programie C++ Builder 7. Ograniczono się do projektowania tarcz tnących na podstawie istniejących tarcz. Stworzona została nowa grupa układów dwutarczowych, a symulacja umożliwiła dobór cech konstrukcyjnych tarcz.*

Słowa kluczowe: *Algorytmy genetyczne, krzyżowanie, tarcze*

Shredder multishield structure design with the use of genetic algorithms.

Abstarct: *This paper deals with the issue concerning genetic algorithms and multi disc mills. The article not only presents genetic algorithm operating and optimization methods but it also concentrates on genetic algorithms application possibilities in them of multi disc mills. AG Discs Software was written with a help of a software called: C++Builder 7. Main focus of interest concerned the cutting disc design process, involving existing ones. New class of double disc systems, whose discs were designed using a computer simulation.*

Keywords: *Genetic algorithms, crossover, shield*

1. WSTĘP

Algorytmy genetyczne znajdują bardzo szerokie zastosowanie w różnych dziedzinach życia codziennego. Obejmują one wiedzę o ludziach, maszynach i zwierzętach, których ogólną cechą jest posiadanie „inteligencji". Rozwój algorytmów genetycznych został zapoczątkowany przez J. Hollanda z Uniwersytetu w Michigan w roku 1975 pracą Adaptation in Natural, a później był kontynuowany przez D.E. Goldberga. Dzięki prostym założeniom i znacznej efektywności w ostatnim czasie rozwój algorytmów genetycznych wskazał nowe rozwiązania w różnych obszarach nauki i techniki.
Sposób działania algorytmów genetycznych naśladuje zjawiska występujące powszechnie w przyrodzie. W tym celu wykorzystuje się mechanizmy doboru naturalnego oraz dziedziczności. Dobór naturalny jest mechanizmem wyznaczającym jednostki, które przeżyją i wydadzą potomstwo. Zdefiniowane kryterium selekcji wyznacza pewną barierę. Przeżyją tylko te osobniki, które są najlepiej przystosowane. Mechanizm dziedziczności polega na przekazywaniu z pokolenia na pokolenie za pomocą genów, cech dziedzicznych w mniej lub bardziej niezmienionej postaci. Geny są nośnikami cech dziedzicznych, które zlokalizowane są w chromosomach. Poprzez krzyżowanie, które jest wymianą odpowiadających sobie odcinków między chromosomami obojga rodziców uzyskuje się „nowe" osobniki. Geny osobników rodzicielskich występują w chromosomach osobników potomnych w różnych kombinacjach. Materiał genetyczny może być także czasami zaburzany przez mutację, która występuje losowo z małym prawdopodobieństwem. Krzyżowanie i mutacja są naturalnymi procesami zabezpieczającymi przed wytworzeniem się jednorodnej genetycznie populacji, która byłaby niezdolna do dalszej ewolucji. Brak ewolucji oznaczałby brak przystosowania populacji do zmieniających się warunków zewnętrznych [4, 5, 6].

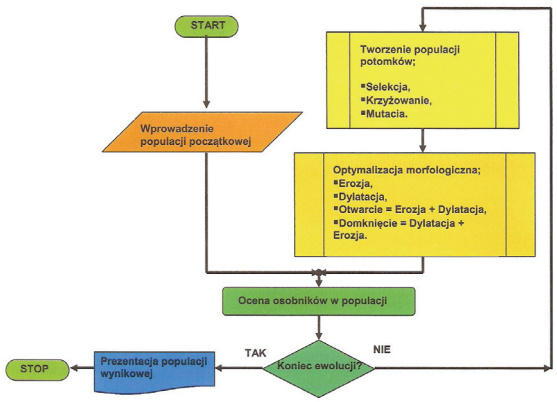
2. CEL PRACY

Celem pracy było opracowanie metody pozwalającej stworzyć nową populację tarcz w oparciu o już istniejące przy użyciu algorytmów genetycznych. Opracowana metoda pozwala także na optymalizację uzyskanych nowych rozwiązań konstrukcyjnych tarcz tnących rozdrabniacza wielotarczowego poprzez wybór najlepszych osobników, bądź eliminację najgorszych.

3. OPIS BADAŃ

Jednym z głównych problemów występujących w trakcie rozdrabniania jest jego duża energochłonność. W tym celu prowadzi się badania nad opracowaniem nowych cech konstrukcyjnych, pozwalających w sposób optymalny na uzyskanie odpowiedniego produktu. Motywacją do podjęcia prac nad algorytmami ewolucyjnymi były ograniczenia występujące w dotychczas stosowanych metodach optymalizacji. Obecnie algorytmy genetyczne stosuje się coraz częściej w budowie i eksploatacji maszyn.
Prezentowana praca jest częścią badań prowadzonych na Uniwersytecie Kazimierza Wielkiego w Bydgoszczy, dotyczących optymalizacji procesu rozdrabniania z użyciem rozdrabniaczy wielotarczowych. W opisanych badaniach doświadczenia przeprowadzano na modelu numerycznym, a ich weryfikację na modelu fizycznym. Badania podzielono na następujące etapy:
- budowa modelu numerycznego,
- weryfikacja modelu, opracowanie modelu fizycznego
- weryfikacja wyników badań na modelu fizycznym.

Weryfikację uzyskanych wyników badan przeprowadzano przy użyciu narzędzia do wirtualnego prototypowania COSMOSMotion.
W celu uzyskania nowych konstrukcji tarcz rozdrabniacza stworzony został program *TarczeAG*. Zasada jego działania opiera się na opracowanym algorytmie przedstawionym na rysunku 1.

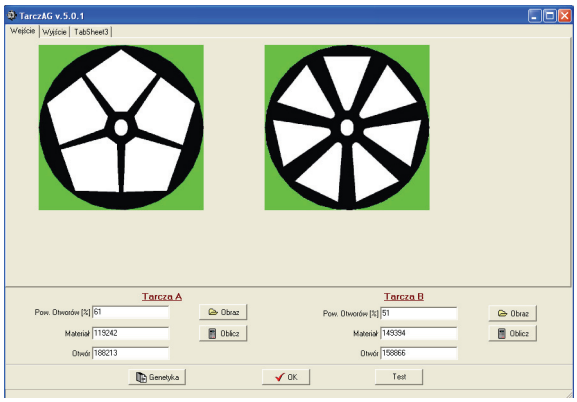


Rysunek. 1 Algorytm metody optymalizacji tarcz

Efektem działania programu jest powstanie rozwiązań konstrukcyjnych nowych układów dwutarczowych [1, 2, 3].

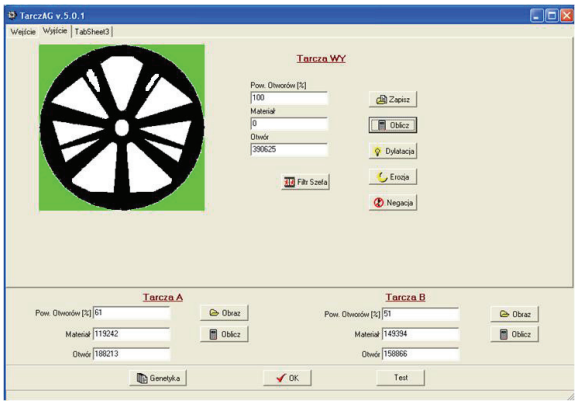
4. OPIS PROGRAMU *TarczeAG*

Wynikiem opracowania algorytmu metody optymalizacji tarcz jest utworzenie programu *TarczeAG,* który pozwala zaproponować z istniejących już tarcz-rodziców, poprzez ich krzyżowanie, nową populację tarcz-dzieci. Każdą nowo powstałą tarczę można także ulepszyć za pomocą operacji morfologicznych.
W implementacji, danymi wejściowymi (rysunek 2) jest układ tarcz tnących, który tworzy populację osobników. Funkcja przystosowania osobnika obliczana jest jako średnia przystosowania każdej z tarcz. Każda tarcza uzupełniona jest o informację opisującą zależność pomiędzy nimi.



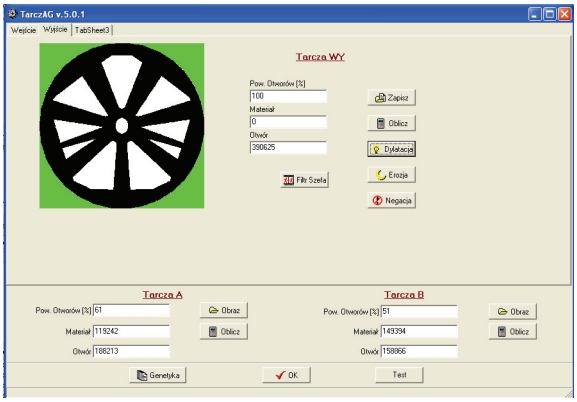
Rysunek. 2 Tarcze wejściowe przed operacją krzyżowania

Na rysunku 3 przedstawiono tarczę uzyskaną po operacji krzyżowania, w której widoczne są małe otwory nie wnoszące nic do konstrukcji. Celem optymalizacji kształtu otworów tnących jest użycie zaimplementowanych operacji morfologicznych.



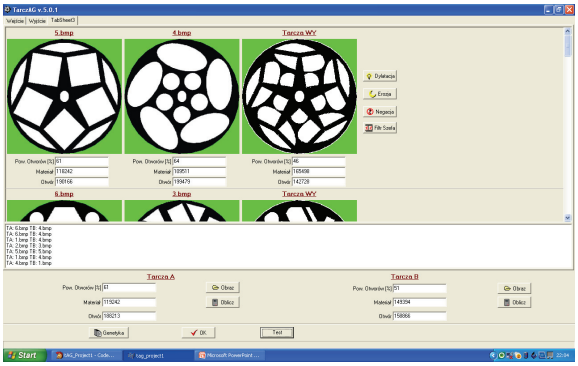
Rysunek. 3 Tarcza wyjściowa – nowy osobnik po operacji krzyżowania

Rysunek 4 przedstawia tą samą tarczę po operacji morfologicznej ***Dylatacja***, która **„**domyka" małe otwory.



Rysunek. 4 Osobnik po operacji: ***Dylatacja***

Dodatkowa zakładka, jaką jest *tAGSheet3* daje nam możliwość krzyżowania aż dwudziestu losowo wybranych osobników-rodziców, które tworzą nową populację. Każdego z tych nowo powstałych osobników można „leczyć", czyli poddać operacji morfologicznej, dzięki czemu otrzymujemy silniejszego osobnika (rysunek 5).



Rysunek. 5 Krzyżowanie wielu osobników

5. PODSUMOWANIE

W niniejszym artykule przedstawiono wycinek badań nad rozdrabniaczami tarczowymi, polegający na wirtualnym projektowaniu tarcz tnących. Zmodernizowano już istniejący program *tAG_Project1* Modernizacja programu polegała na poprawie jego algorytmu działania, zaimplementowano także nową metodę pozwalającą na tworzenie nowej populacji w oparciu o algorytmy genetyczne. Istnieje możliwość wyboru najodpowiedniejszych rozwiązań powierzchni tarcz tnących rozdrabniacza wielotarczowego w oparciu o „leczenie" powstałych używając do tego operacji morfologicznych. Zastosowanie programu pozwala na uzyskanie nowych konstrukcji par tarcz rozdrabniających w celu dalszych badań nad optymalizacją procesu rozdrabniania za pomocą rozdrabniaczy wielotarczowych.

Literatura

1. Arabas J., „*Wykłady z Algorytmów Ewolucyjnych"*, WNT, 2001
2. Czerniak J., „*Algorytmy genetyczne w praktyce"*, Software 2.0 Extra, Numer 10, Sztuczna inteligencja, 2004, ss. 60-65
3. Czerniak J., Macko M., „*Zastosowanie algorytmów genetycznych do optymalizacji konstrukcji tarcz tnących*", Instytut Badań Systemowych PAN, Badania systemowe – Teoria i zastosowania, Sesja sprawozdawcza, W-wa 2009
4. Goldberg D. E., „*Algorytmy genetyczne i ich zastosowanie"*, WNT Warszawa, 2003, s. 17
5. Gwiazda T.D., „*Algorytmy genetyczne - wstęp do teorii*", Warszawa 1995, s. 57

6. Flizikowski J., Kamyk W., „*Chromosomy algorytmów inżynierii rozdrabniania*", Inżynieria Maszyn-Inżynieria mechaniczna żywności, Bydgoszcz, luty 2004, s. 157
7. Murawski K., „*Obliczenia ewolucyjne – geneza i zastosowanie*" , Biuletyn Instytutu Automatyki i Robotyki WAT NR 15,2001, ss. 72-72
8. Syswerda G. *"Foundation of Genetic Algorithms",* (1991), ss. 94-101

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str. 55-58

SZYBKA DYSKRETNA TRANSFORMATA SINUSOWA

Robert Rychcicki

Zachodniopomorski Uniwersytet Technologiczny
Wydział Informatyki
ul. Żołnierska 49, 71-210 Szczecin
e-mail: rrychcicki@wi.ps.pl

Streszczenie: *Celem pracy jest zaproponowanie szybkiej metody obliczeniowej pozwalającej na wyznaczenie DST-IV (oraz transformaty odwrotnej) o złożoności O(n*lgn) pod względem liczby mnożeń. Wybór DST-IV podyktowany jest brakiem atrakcyjnych zależności w macierzy opisującej przekształcenie – większość prac polskich i zagranicznych [1,2,3] opisujących efektywne metody konstrukcji grafów przebiegu obliczeń opiera się o DST-II/DST-III, których analiza jest prostsza. Opracowana metoda zostanie przedstawiona w postaci matematycznej.*

Słowa kluczowe: *Transformata sinusowa, transformata dyskretna, DST*

Fast Discrete Sine Transform

Abstarct: *The aim of this work is to present a fast calculation method for DST-IV and inverse transform, whose complexity is O(n*lgn) with regard to multiplication count. DST-IV was chosen due to lack of attractive dependencies in the matrix describing the transformation. Most works (both Polish and foreign [1,2,3]) elucidating effective methods of producing graphs describing the calculation process are based on DST-II/DST-III, whose analysis is by far less complicated. The proposed method will be presented in a mathematical form..*

Keywords: *Sine transform, discrete transform, DST*

1. WSTĘP

W popularnej literaturze [2,4] można znaleźć wiele opracowań dotyczących efektywnych i szybkich algorytmów obliczających szybką dyskretną transformatę Fouriera, przydatną w wielu zastosowaniach związanych z cyfrowym przetwarzaniem sygnałów.
Nieco mniej miejsca poświęcono na szybką dyskretną transformatę cosinusową – używaną m.in. w kompresji obrazu i dźwięku.
Natomiast bliźniacza do niej szybka dyskretna transformata sinusowa nie okazała się atrakcyjnym tematem dla wielu naukowców.
Dlaczego? Powodów można wymienić kilka – przydatna własność funkcji cosinus, jaką jest jej parzystość cos(-x)=cos(x), węższe spektrum zastosowań [3,5] dla transformaty sinusowej (np. steganografia), stosunkowa łatwość przekształcania baz transformat Fouriera, sinusowej i cosinusowej oraz nieskomplikowane zależności matematyczne łączące te transformaty.
DCT i DST różni się od DFT przyjętymi warunkami brzegowymi – wszystkie te transformaty operują na skończonym zestawie próbek, jednak przyjmują inne założenia odnośnie „rozszerzenia" funkcji poza ograniczoną dziedzinę. Tak jak przy DFT zakłada się okresowe rozszerzenie dziedziny, tak przy DST przyjmuje się parzystość funkcji, a przy DCT – nieparzystość. Daje to aż 16 przypadków transformat trygonometrycznych – zależnie od parzystości/nieparzystości na lewym i prawym krańcu dziedziny oraz punkty symetrii dla każdego z krańców (skrajna próbka może być przy rozszerzeniu dublowana bądź nie). DCT (parzystość na lewym krańcu dziedziny) to grupa 8 z tych transformat (w szczególności I, II, V, VI – zachowują ciągłość na obu krańcach dziedziny po jej rozszerzeniu), a DST – pozostałe 8. Zależnie od rodzaju

analizowanego sygnału [3] i oczekiwanych efektów, stosujemy odpowiednią transformatę.

2. MATEMATYCZNY OPIS DST

Istnieje kilka podstawowych typów DST. DST-IV dana jest wzorem:

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right] \qquad k = 0,\ldots,N-1$$

W podobny sposób opisana jest bliźniacza transformata DCT-IV:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right] \qquad k = 0,\ldots,N-1$$

Transformatę można również opisać za pomocą iloczynu wektorowo-macierzowego:

$$\begin{bmatrix} X_0 \\ \vdots \\ X_{n-1} \end{bmatrix} = \begin{bmatrix} sin(\frac{\pi}{4n}) & \cdots & sin(2n-1\frac{\pi}{4n}) \\ \vdots & \ddots & \vdots \\ sin(2n-1\frac{\pi}{4n}) & \cdots & -sin(\frac{\pi}{4n}) \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ \vdots \\ X_{n-1} \end{bmatrix} = DST_n \cdot \begin{bmatrix} x_0 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

gdzie macierz DST_n składa się z czynników

$$S_{k,n}^N = \sin\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right]$$

$$k = 0,\cdots,N-1 \qquad n = 0,\cdots,N-1$$

W rozważanym przypadku 8-punktowego DST-IV:

$DST_8 =$

sin (1π/32)	sin (3π/32)	sin (5π/32)	sin (7π/32)	sin (9π/32)	sin (11π/32)	sin (13π/32)	sin (15π/32)
sin (3π/32)	sin (9π/32)	sin (15π/32)	sin (11π/32)	sin (5π/32)	sin (1π/32)	sin (7π/32)	sin (13π/32)
sin (5π/32)	sin (15π/32)	sin (7π/32)	sin (3π/32)	sin (13π/32)	sin (9π/32)	sin (1π/32)	sin (11π/32)
sin (7π/32)	sin (11π/32)	sin (3π/32)	sin (15π/32)	sin (1π/32)	sin (13π/32)	sin (5π/32)	sin (9π/32)
sin (9π/32)	sin (5π/32)	sin (13π/32)	sin (1π/32)	sin (15π/32)	sin (3π/32)	sin (11π/32)	sin (7π/32)
sin (11π/32)	sin (1π/32)	sin (9π/32)	sin (13π/32)	sin (3π/32)	sin (7π/32)	sin (15π/32)	sin (5π/32)
sin (13π/32)	sin (7π/32)	sin (1π/32)	sin (5π/32)	sin (11π/32)	sin (15π/32)	sin (9π/32)	sin (3π/32)
sin (15π/32)	sin (13π/32)	sin (11π/32)	sin (9π/32)	sin (7π/32)	sin (5π/32)	sin (3π/32)	sin (1π/32)

Tabela 1 Macierz opisująca transformatę DST(8).

3. DZIEL I ZWYCIĘŻAJ – OPIS MATEMATYCZNY METODY

Analiza macierzy definiującej przekształcenie DST-IV pozwala na wykorzystanie metody „dziel i zwyciężaj" podobnej do użycia „decymacji" przy obliczaniu FFT algorytmem Cooley-Tukey.

Macierz DST_8 z zaznaczonymi omawianymi fragmentami:

$DST_8 =$

sin (1π/32)	sin (3π/32)	sin (5π/32)	sin (7π/32)	sin (9π/32)	sin (11π/32)	sin (13π/32)	sin (15π/32)
sin (3π/32)	sin (9π/32)	sin (15π/32)	sin (11π/32)	sin (5π/32)	-sin (1π/32)	-sin (7π/32)	-sin (13π/32)
sin (5π/32)	sin (15π/32)	sin (7π/32)	-sin (3π/32)	-sin (13π/32)	-sin (9π/32)	sin (1π/32)	sin (11π/32)
sin (7π/32)	sin (11π/32)	-sin (3π/32)	-sin (15π/32)	-sin (1π/32)	sin (13π/32)	sin (5π/32)	-sin (9π/32)
sin (9π/32)	sin (5π/32)	-sin (13π/32)	-sin (1π/32)	sin (15π/32)	-sin (3π/32)	-sin (11π/32)	sin (7π/32)
sin (11π/32)	-sin (1π/32)	-sin (9π/32)	sin (13π/32)	sin (3π/32)	-sin (7π/32)	sin (15π/32)	-sin (5π/32)
sin (13π/32)	-sin (7π/32)	sin (1π/32)	sin (5π/32)	-sin (11π/32)	-sin (15π/32)	sin (9π/32)	sin (3π/32)
sin (15π/32)	-sin (13π/32)	sin (11π/32)	sin (9π/32)	sin (7π/32)	-sin (5π/32)	sin (3π/32)	-sin (1π/32)

Tabela 2 Zależności w transformacie DST(8).

Porównanie z macierzą DST_4:

$DST_4 =$

sin(2π/32)	sin(3π/16)	sin(5π/16)	sin(7π/16)
sin(3π/16)	sin(7π/16)	sin(1π/16)	-sin(5π/16)
sin(5π/16)	sin(1π/16)	-sin(7π/16)	sin(3π/16)
sin(7π/16)	-sin(5π/16)	sin(3π/16)	-sin(1π/16)

Tabela 3 Zależności w transformacie DST(4).

Argumenty funkcji sinus dla DST_8 odpowiadają argumentom w $DST_4 \pm \pi/32$. Nie jest to przypadek. Powracając do wzoru definiującego DST:

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right] \qquad k = 0,\cdots,N-1$$

poszczególne komórki macierzy dane są wzorem

$$S_{k,n}^N = \sin\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right]$$

$$k = 0,\cdots,N-1 \qquad n = 0,\cdots,N-1$$

Po porównaniu wartości poszczególnych komórek macierzy dla DST o rozmiarze *2N* i *N* uzyskujemy:

$$X_k^{2N} = \sum_{n=0}^{2N-1} x_n S_{k,n}^{2N} \qquad k = 0,\cdots,2N-1$$

Porównując to z odpowiednimi komórkami macierzy dla DST z 2N próbek otrzymamy:

$$S_{k,n}^{N} = \sin\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right]$$
$$k = 0, \cdots, N-1 \qquad n = 0, \cdots, N-1$$

$$S_{k,2n}^{2N} = \sin\left[\frac{\pi}{8N}(4n+1)(2k+1)\right]$$
$$S_{k,2n}^{2N} = \sin\left[\frac{\pi}{8N}(4n+2)(2k+1) - \frac{\pi}{8N}(2k+1)\right]$$

Ponieważ:

$$\sin(x \pm y) = \sin(x)\cos(y) \pm \cos(x)\sin(y)$$

uzyskujemy

$$S_{k,2n}^{2N} = \sin\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right]\cos\left[\frac{(2k+1)\pi}{8N}\right] -$$
$$-\cos\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right]\sin\left[\frac{(2k+1)\pi}{8N}\right]$$

Wprowadzając oznaczenie

$$C_{k,n}^{N} = \cos\left[\frac{2\pi}{8N}(2n+1)(2k+1)\right]$$
$$k = 0, \cdots, N-1 \qquad n = 0, \cdots, N-1$$

uzyskujemy

$$S_{k,2n}^{2N} = S_{k,n}^{N}\cos\left[\frac{(2k+1)\pi}{8N}\right] - C_{k,n}^{N}\sin\left[\frac{(2k+1)\pi}{8N}\right]$$

Analogicznie:

$$S_{k,2n+1}^{2N} = \sin\left[\frac{\pi}{8N}(4n+2)(2k+1) + \frac{\pi}{8N}(2k+1)\right]$$
$$S_{k,2n+1}^{2N} = S_{k,n}^{N}\cos\left[\frac{(2k+1)\pi}{8N}\right] + C_{k,n}^{N}\sin\left[\frac{(2k+1)\pi}{8N}\right]$$

Bardzo podobną zależność można znaleźć dla pozostałych współczynników wykorzystując fakt, że:

$$X_{2N-1-k}^{2N} = \sum_{n=0}^{2N-1} x_n(-1)^n C_{k,n}^{2N} \qquad k = 0, \cdots, N-1$$

wykonując analogiczne do poprzednich przekształcenia uzyskujemy natychmiast:

$$C_{k,2n}^{2N} = C_{k,n}^{N}\cos\left[\frac{(2k+1)\pi}{8N}\right] + S_{k,n}^{N}\sin\left[\frac{(2k+1)\pi}{8N}\right]$$
$$C_{k,2n+1}^{2N} = C_{k,n}^{N}\cos\left[\frac{(2k+1)\pi}{8N}\right] - S_{k,n}^{N}\sin\left[\frac{(2k+1)\pi}{8N}\right]$$

Obliczenie DST zostało zatem sprowadzone do obliczenia dwukrotnie mniejszego DST dla próbek będących sumami kolejnych par próbek oryginalnych oraz DCT dla kolejnych różnic par próbek.

$$X_k^{2N} = \sum_{n=0}^{N-1}\left[(x_{2n} + x_{2n+1})\,S_{k,n}^{N}\cos\left[\frac{(2k+1)\pi}{8N}\right] -\right.$$
$$\left. -\,(x_{2n} - x_{2n+1})\,C_{k,n}^{N}\sin\left[\frac{(2k+1)\pi}{8N}\right]\right]$$
$$X_{2N-1-k}^{2N} = \sum_{n=0}^{N-1}\left[(x_{2n} - x_{2n+1})\,C_{k,n}^{N}\cos\left[\frac{(2k+1)\pi}{8N}\right] +\right.$$
$$\left. +\,(x_{2n} + x_{2n+1})\,S_{k,n}^{N}\sin\left[\frac{(2k+1)\pi}{8N}\right]\right]$$

Z założenia, DCT (różnicy kolejnych par próbek $X^{(-)}$) liczymy jednym ze znanych szybkich algorytmów, a DST (sumy kolejnych par próbek $X^{(+)}$) dekomponujemy rekurencyjnie w ten sam sposób.

$$X_k^{2N} = {X^{(+)}}_k^N\cos\left[\frac{(2k+1)\pi}{8N}\right] - {X^{(-)}}_k^N\sin\left[\frac{(2k+1)\pi}{8N}\right]$$
$$X_{2N-1-k}^{2N} = {X^{(-)}}_k^N\cos\left[\frac{(2k+1)\pi}{8N}\right] + {X^{(+)}}_k^N\sin\left[\frac{(2k+1)\pi}{8N}\right]$$

Należy ustalić jeszcze warunek brzegowy rekurencji, np. dla danych o rozmiarze N=1 lub N=2.

$$\mathbf{DST_2} = \begin{vmatrix} \sin(1\pi/8) & \sin(3\pi/8) \\ \sin(3\pi/8) & -\sin(1\pi/8) \end{vmatrix} \qquad \mathbf{DCT_2} = \begin{vmatrix} \cos(1\pi/8) & \cos(3\pi/8) \\ \cos(3\pi/8) & -\cos(1\pi/8) \end{vmatrix}$$

Tabela 4 Warunek brzegowy dla propozycji DST-IV.

4. ZŁOŻONOŚĆ OBLICZENIOWA

Analiza złożoności obliczeniowej pod kątem ilości wykonanych mnożeń (rzeczywistych) przedstawionego algorytmu nie jest trudna – załóżmy, że DCT nie wymaga więcej mnożeń niż DST dla takiego samego rozmiaru danych (co zostało pokazane przy sprowadzanie DST do DCT).

$DST_{(N=2)}$ wymaga 3 mnożeń.
$DST_{(N=4)}$ składa się z $DST_{(N=2)}$, $DCT_{(N=2)}$ i ich łączenia zrealizowanego tymczasowo (nieoptymalnie) przy użyciu 8 mnożeń. Łącznie zostanie więc wykonanych 14 mnożeń.
$DST_{(N=8)}$ składa się z $DST_{(N=4)}$, $DCT_{(N=4)}$ i łączenia - zostaną wykonane 44 mnożenia.
$DST_{(N=16)}$ składa się z $DST_{(N=8)}$, $DCT_{(N=8)}$ i łączenia - Łącznie zostanie wykonanych 120 mnożeń.

Kontynuując obliczenia uzyskujemy następujące wyniki:

N	MW(N)	MP(N)	*MO(N)*
2	4	3	*3*
4	16	14	*14*
8	64	44	*39*
16	256	120	*82*
32	1 024	304	*176*
64	4 096	736	*380*
128	16 384	1 728	*820*
256	65 536	3 968	*1 764*
512	262 144	8 960	*3 780*
1024	1 048 576	19 968	*8 068*
2048	4 194 304	44 032	*17 156*
4096	16 777 216	96 256	*36 356*
8192	67 108 864	208 896	*76 804*

Tabela 5 Złożoność obliczeniowa.

Tabela zawiera dane: MW(N) - Liczba mnożeń przy użyciu iloczynu macierzowo-wektorowego, MP(N) - Pesymistyczna liczba mnożeń w proponowa–nej metodzie, MO(N) - Oczekiwana liczba mnożeń w propono–wanej metodzie

Wartości oczekiwane podano przy założeniu, że zastosujemy bardziej efektywne algorytmy obliczania DCT-IV – liczba ta ulegnie zmianie w zależności od wyboru metody.

5. WŁASNOŚCI ALGORYTMU

Zaproponowana metoda ma duże zalety:

- Działa dla dowolnego rozmiaru danych $N=2^C$ oraz innych rozmiarów, gdy dane wejściowe dopełni się zerami,
- Jej pesymistyczna złożoność obliczeniowa to *O(n*lgn)*, zarówno pod względem mnożeń, jak i dodawań, spełnia więc wymagania wystarczające, aby nadać jej status „szybkiego algorytmu" w tej klasie zadań,
- Na każdym z poziomów rekurencji można zastosować gotowy, opracowany moduł realizujący szybką transformatę dla określonego rozmiaru danych wejściowych, co dodatkowo zmniejszy liczbę operacji matematycznych,
- Możliwość stosowania różnych metod obliczeniowych na każdym z poziomów rekurencji,

Jest podatna na dalsze optymalizacje, ponieważ graf opisujący przeprowadzane obliczenia może zostać łatwo uproszczony

Literatura

1. Xuancheng Shao, Steven G. Johnson. Type-II/III DCT/DST algorithms with reduced number of arithmetic operations. 2008. Signal Processing Volume 88 , Issue 6 (June 2008), pp 1553-1564, ISSN:0165-1684
2. Markus Päuschel, José M. F. Mouray. The algebraic approach to the discrete cosine and sine transforms and their fast algorithms. SIAM Journal of Computing 2003, Vol. 32, No. 5, pp. 1280-1316
3. Vladimir Britanak, Patrick C. Yip, K. R Rao, Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations ISBN-13: 978-0123736246, Academic Press (2006);
4. Wolfram Research http://documents.wolfram.com/applications/wavelet/FundamentalsofWavelets/1.7.2.html (październik 2007)
5. Ross J. Anderson, Fabien A.P. Petitcolas, On The Limits of Steganography IEEE Journal of Selected Areas in Communications, 16(4):474-481, Maj 1998. Special Issue on Copyright & Privacy Protection. ISSN 0733-8716.

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str. 59-66

ZASTOSOWANIE ALGORYTMÓW POSZUKIWANIA Z TABU DO OPTYMALIZACJI UKŁADANIA PLANU ZAJĘĆ

Eliza Witczak
Uniwersytet Kazimierza Wielkiego
Instytut Techniki
ul. Chodkiewicza 30, 85-064 Bydgoszcz
e-mail: elizawitczak@go2.pl

Streszczenie: *TS jest mataheurystyką szukającą rozwiązania problemu poprzez nadzorowanie innych procedur heurystycznych lokalnego przeszukiwania, w celu eksploracji przestrzeni rozwiązań poza lokalne optimum. Proces przeszukiwania przestrzeni rozwiązań jest koordynowany za pomocą strategii opartych na mechanizmach pamięci, będących cechą charakterystyczną algorytmu TS. Niniejszy artykuł opisuje zastosowanie metody TS w procesie optymalizacji rozkładu zajęć.*

Słowa kluczowe: *Tabu serach, generowanie rozkładu zajęć*

Tabu search algorithms for timetabling optimization

Abstarct: *Tabu Search is a metaheuristic that guides a local heuristic search procedures to explore the solution space beyond local optimum. The process of searching the solutions space is coordinated by strategies based on the memory mechanisms, which are characters of the TS algorithm. This paper describes the method of TS in the timetable optimization process.*

Keywords: *Tabu search, timetabling*

1. WSTĘP

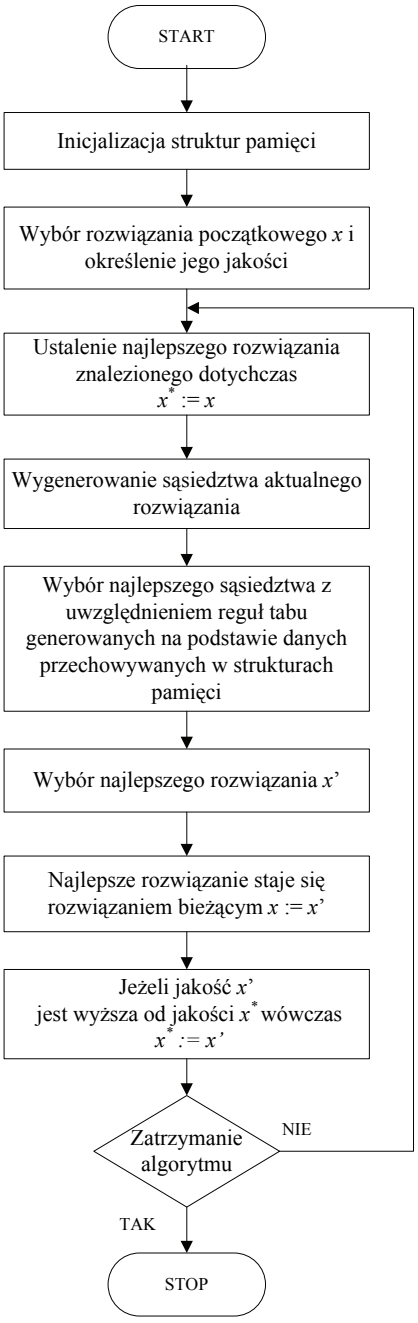
Tworzenie rozkładu zajęć polega na wyznaczeniu kombinacji wykładowca-grupa dla określonych okresów czasu, pedagogów i grup uczniów. Celem tego procesu jest zmniejszenie liczby konfliktów, które pojawiają się gdy kilku wykładowców musi prowadzić zajęcia z jedną grupą studentów w tym samym ograniczonym przedziale czasu lub gdy różne lekcje wymagają wspólnych sal. Główna trudność związana jest ze skalą problemu. Z jednej strony należy uwzględnić potrzeby nauczycieli, oczekujących zwartego planu zajęć i minimalnej liczby okienek, z drugiej strony natomiast higienę nauki uczniów. Często oba te cele nawzajem się wykluczają i jedynym rozwiązaniem jest znalezienie jak najlepszego kompromisu.

2. ALGORYTM TABU SEARCH

TS jest mataheurystyką szukającą rozwiązania problemu poprzez nadzorowanie innych procedur heurystycznych lokalnego przeszukiwania, w celu eksploracji przestrzeni rozwiązań poza lokalne optimum. Proces przeszukiwania przestrzeni rozwiązań jest koordynowany za pomocą strategii opartych na mechanizmach pamięci, będących cechą charakterystyczną dla algorytmu TS.

Algorytm TS dokonuje pełnego przeszukiwania sąsiedztwa aktualnego rozwiązania. Rozwiązanie bieżące zastępowane jest przez najlepsze rozwiązanie w sąsiedztwie, nawet jeśli powoduje to pogorszenie jakości. W procesie przeszukiwania wykorzystywany jest system ograniczeń nałożonych na zbiór należący do sąsiedztwa. Ma to na celu przeciwdziałanie możliwości zapętlenia się algorytmu i powracania do tych samych wąskich

obszarów rozwiązań. Ze zdefiniowanego sąsiedztwa usuwa się rozwiązania, które wcześniej były zaakceptowane jako rozwiązania bieżące tworząc tzw. zbiór tabu. Ograniczenia stosowane w TS te mają formę absolutnego zakazu lub pewnych restrykcji.



Rys. 1. Schemat działania algorytmu Tabu Search

Podstawowym i najbardziej charakterystycznym elementem mechanizmu tabu jest pamięć. W trakcie przeszukiwania gromadzone są w niej informacje o przeszukiwanej przestrzeni. Lokalne wybory są uzależnione od informacji zebranych podczas całego przeszukiwania. Na podstawie zapisanych w pamięci informacji tworzone są ograniczenia, które chronią algorytm przed powracaniem do przeszukanych już obszarom przestrzeni. Ograniczenia te zależą od:

- częstotliwości wykonywania zapisu określonych danych,
- aktualności zapisanych w pamięci danych,
- wpływu określonych danych na jakość uzyskiwanych wyników.

W kolejnych iteracjach algorytm przeszukuje sąsiedztwo aktualnie znalezionego rozwiązania w celu określenia nowego położenia rozwiązania bieżącego. W związku z tym dla zadanej przestrzeni poszukiwań musi zostać zdefiniowana relacja sąsiedztwa dla wszystkich jej elementów.

Struktura pamięci przechowuje przede wszystkim informacje o zrealizowanych przejściach. Może także zawierać inne informacje, np. o częstotliwości tych przejść lub czasie, który upłynął od wykonania każdego z przejść. W algorytmach TS mamy do czynienia z dwoma rodzajami pamięci: krótkoterminową i długoterminową. Każda z nich jest wykorzystywana przez charakterystyczne dla siebie strategie. Efekt działania ich obu jest widoczny jako modyfikacja sąsiedztwa $N(x)$ aktualnego rozwiązania x. Zmodyfikowane sąsiedztwo $N^*(x)$ jest rezultatem przechowywania informacji dotyczących dotychczasowego procesu przeszukiwania.

Pamięć krótkoterminowa jest wykorzystywana w każdej iteracji i służy do zapamiętywania ostatnio odwiedzonych rozwiązań i nałożonych na nie kar. Jej głównym zadaniem jest uniknięcie wyboru operatora ruchu, który może doprowadzić do zapętlenia się algorytmu w pewnym wąskim obszarze przestrzeni poszukiwań.

Pamięć długoterminowa przechowuje informacje o przebiegu procesu poszukiwania Pozwala na zapamiętanie najlepszych rozwiązań z odwiedzonych już obszarów przestrzeni poszukiwań a nie wyłącznie najlepszego rozwiązania z bieżącego sąsiedztwa.

W celu określenia relacji sąsiedztwa należy określić operator zmiany bieżącego rozwiązania, który zdefiniuje sąsiedztwo punktu: każdy punkt, który da się uzyskać z punktu bieżącego, wykorzystując operator zmiany, jest sąsiadem tego punktu bieżącego, a pozostałe punkty - nie.

Przy każdym kroku algorytmu musi zostać określony zbiór przejść do nowych położeń należących do sąsiedztwa rozwiązania aktualnego. Z nich zostanie wybrane jedno, które stanie się nowym bieżącym rozwiązaniem. Dla każdego z wygenerowanych przejść muszą zostać określone i sprawdzone wszystkie jago atrybuty. Jeżeli wartość takiego atrybutu jest określona choć jednym ograniczeniem tabu, atrybut ten jest tabu-aktywny. W przeciwnym wypadku atrybut jest tabu-nieaktywny. Po określeniu statusu wszystkich atrybutów każdego nowego przejścia generowane są reguły określające czy dane przejście jest tabu-aktywne czy nie. Status tabu-aktywny jednego atrybutu przejścia nie musi oznaczać, że status całego przejścia jest również tabu-aktywny Zależy to od reguł wykorzystujących status atrybutów. Jeśli przykładowo reguła mówi, że status przejścia do nowego położenia jest sumą logiczną dwóch określonych atrybutów, wówczas tabu-nieaktywność obu rozważanych atrybutów powoduje tabu-nieaktywność całego przejścia. Ustawienie statusu tabu-aktywnego dla określonego atrybutu wymaga zawsze określenia liczby iteracji, przez które status ten będzie utrzymywany. Po wykonaniu tej liczby iteracji status atrybutu ustawiany jest na tabu-nieaktywny. Jeżeli jednak wcześniej wykonany zostanie ruch tabu, ponownie uaktywniający ten atrybut, wówczas czas aktywności dla niego jest liczony od początku. Określenie długości kadencji tabu nie musi być jednakowa dla wszystkich atrybutów, powinna nawet być określona indywidualnie dla każdego z nich. Długość ta może być stała lub zmienna w czasie. W takim przypadku może być ustalona na sztywno (np. może być funkcją numeru aktualnie wykonywanej iteracji) lub zmieniać się losowo. Może być również funkcją współczynników określających postępy czynione przez algorytm. Każde przejście może zostać określone różnymi atrybutami. Może to być np. wartość kary, jaka jest mu przypisana, liczba bądź częstotliwość wykonywania tego przejścia w ostatnim czasie, okres tabu, tj. pozostałą liczbę iteracji, przez które będzie jeszcze obowiązywał zakaz jego wykonania. Wszystkie te informacje będą zapisywane do struktur pamięci w trakcie realizacji algorytmu. Te dwa typy zdarzeń różnią się między sobą: wysoka częstotliwość występowania pewnej wartości na określonej pozycji oznacza, że ta właśnie wartość atrybutu jest szczególnie pożądana. Natomiast wysoka zmienność wartości na określonej pozycji określa wysoką nieprzewidywalność tego atrybutu i oznacza, że jego rola jest prawdopodobnie większa przy końcowym dostrajaniu się do optimum, niż we wstępnej części procesu optymalizacji, tj. przy początkowej eksploracji w poszukiwaniu najbardziej obiecującego regionu dziedziny.

Poszukiwanie nowej wartości rozwiązania bieżącego wygląda następująco: operator przejścia do następnego położenia wybiera pewien podzbiór rozwiązań z sąsiedztwa aktualnego położenia i szuka wśród nich najlepszego, uwzględniając jednak przy określaniu wartości i dostępności rozwiązań reguły tabu. Po wygenerowaniu zbioru sąsiadów jest on analizowany i po uwzględnieniu wszystkich reguł tabu najlepszy znaleziony w tym zbiorze osobnik staje się nowym bieżącym rozwiązaniem algorytmu. Informacja o tym przejściu zapisywana jest do struktur pamięci.

Reguły tabu, którymi posługuje się operator przejścia, mają za zadanie niedopuszczenie do utknięcia procesu przeszukiwania w miejscowych optimach. Umożliwiają także wycofanie się z już odwiedzonych rejonów przestrzeni rozwiązań, dzięki czemu algorytm może dokonać przeszukiwania na większej przestrzeni. Reguły tabu mogą być różnorodne. Mogą one wprowadzać zakazy całkowite lub częściowe. Reguły te mogą także uwzględniać czas, np. zakaz przejścia między punktami może zostać anulowany po upływie ściśle określonego czasu (czyli liczby iteracji) od ostatniego takiego przejścia. Po upływie pewnego czasu mogą również maleć kary by w końcu zostać anulowane. Jednak po powtórnym wykonaniu takiego przejścia są one przywracane.

Ograniczenia tabu mogą zakazywać atrakcyjnych ruchów, nawet kiedy nie ma żadnego niebezpieczeństwa utknięcia w lokalnym optimum lub mogą prowadzić do stagnacji procesu przeszukiwania. Jest więc konieczne użycie narzędzi, które pozwolą odwołać tabu. Nazywane są one kryteriami aspiracji. Złagodzenie ograniczeń może zostać wdrożone poprzez kasowanie wybranego ograniczenia i dodawaniu do funkcji oceny pewnej kary za naruszanie zakazu. Istnieje kilka sposobów określania wartości kar. Jednym z nich jest zastosowanie tzw. samoregulującej kary – jej wartość zmienia się dynamicznie na podstawie historii procesu przeszukiwania

3. IMPLEMENTACJA SYSTEMU UKŁADANIA PLANU WSPOMAGANEGO TS

Proces układania planu lekcji został zaimplementowany w środowisku Matlab i polega na wygenerowaniu planu dla każdego z wykładowców i każdej z grup. Tworzony jest również rozkład wykorzystania poszczególnych sal. Zbiór wszystkich cząstkowych rozkładów zajęć tworzy plan wynikowy. Celem działania programu jest zmniejszenie liczby konfliktów, które pojawiają się gdy kilku wykładowców musi prowadzić zajęcia z jedną grupą studentów w tym samym ograniczonym przedziale czasu lub gdy różne lekcje wymagają wspólnych sal. Główna trudność związana jest ze skalą problemu. Oprócz dużej ilości grup, wykładowców, zajęć i sal musi zostać uwzględnionych wiele dodatkowych warunków i celów.

W tworzeniu rozkładu zajęć ograniczenia podzielone są na dwa rodzaje. Ograniczenia ciężkie (H) muszą zostać bezwzględnie spełnione. Warunki, które są uważane za pomocne ale nie konieczne w dobrym rozkładzie zajęć są tzw. ograniczeniami lekkimi (S). W im większym stopniu zostaną one spełnione, tym lepsza jakość rozkładu zajęć zostanie osiągnięta. Z tego powodu mają wpływ na wartość funkcji oceny.
Plan zajęć składa się ze zbioru m wykładowców $t_1,\ldots t_m$ prowadzących zajęcia z n grupami studentów $c_1,\ldots,c_n$ w ciągu okresów $p_1,\ldots,p_j$. Przydział nauczycieli do poszczególnych grup jest z góry ustalony a obciążenie pracą zapisane jest w matrycy wymagań. Rozwiązanie problemu polega na minimalizacji funkcji oceny f(x) uwzględniającej stopień spełnienia ograniczeń przez wygenerowane rozwiązanie.

Funkcja oceny wyraża się wzorem:

$$f(x) = \sum_r w_r f_r(x)$$

gdzie r jest kolejnym numerem ograniczenia.

Wartości wag w_r są zdefiniowane przez programistę i określają jaki znaczenie w całym procesie ma dane ograniczenie. Funkcje $f_r(x)$ są sumą wartości wszystkich rozwiązań x_{ita} dla parametrów określonych przez dane ograniczenie. Sumy cząstkowe związane z ograniczeniami ciężkimi określają poziom wykonalności natomiast sumy powiązane z ograniczeniami lekkimi są miarą zadowolenia.

Ograniczenia ciężkie, przyjęte przy tworzeniu programu, przedstawiają się następująco:

(Hl) Każda lekcja musi być przypisana do okresu albo pewnej liczby okresów, zależących o długości lekcji.
(H2) Żaden nauczyciel nie może prowadzić dwóch różnych zajęć w tym samych czasie.
(H3) Żadna grupa nie może uczestniczyć w dwóch różnych zajęciach w tym samym czasie.
(H4) Dwie równoczesne lekcje nie mogą odbywać się w jednej sali.
(H5) Sala wyznaczona do danej lekcji musi być wymaganego typu.
(H6) Okresy dostępności/niedostępności nauczycieli muszą być przestrzegane.
(H7) Lekcje poszczególnych grup jednego rocznika nie mogą kolidować z zajęciami wspólnymi dla tego rocznika.

Ograniczenia lekkie mają postać:

(S1) Rozkłady zajęć dla poszczególnych grup powinny mieć jak najmniej „okienek" by wyeliminować czasy bezczynności studentów.
(S2) Rozkłady zajęć dla poszczególnych nauczycieli powinny mieć jak najmniej „okienek" by wyeliminować czasy bezczynności nauczycieli.
(S3) Studenci nie powinni mieć więcej niż określoną liczbę godzin poszczególnych bloków zajęć (przedmiot/wykładowca/rodzaj zajęć).

Maksymalna liczba godzin ograniczenia (S3) określana jest przez użytkownika. Istnieje także możliwość zmiany maksymalnej ilości iteracji, ilości zjazdów dla wprowadzonej siatki godzin oraz rodzaju studiów. W przypadku studiów zaocznych zajęcia odbywają się od piątku do niedzieli, przy czym użytkownik ma możliwość wyboru godziny rozpoczęcia lekcji w pierwszy dzień zjazdu. Zajęcia studiów dziennych odbywają się w dni powszednie.

Stałymi parametrami algorytmu są:

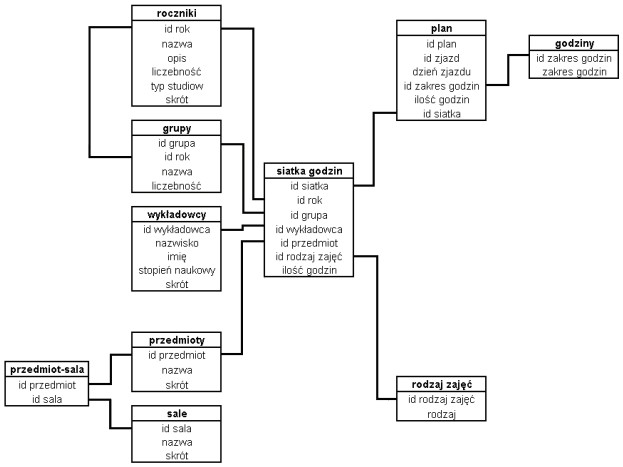
- czas tabu,
- wagi w_r funkcji oceny,
- ilość iteracji, po której następuje intensyfikacja/dywersyfikacja.

Generowanie rozkładu zajęć musi zostać poprzedzone wprowadzeniem podstawowych informacji. Opisują one:

- wykładowców,
- roczniki,
- grupy,
- przedmioty,
- sale,
- rodzaj zajęć.

Dane te zostają wykorzystane do wprowadzenia siatki godzin, tj. listy wszystkich zajęć odbywających się w określonym semestrze/roku. Na podstawie zawartych w niej informacji tworzony jest plan zajęć.

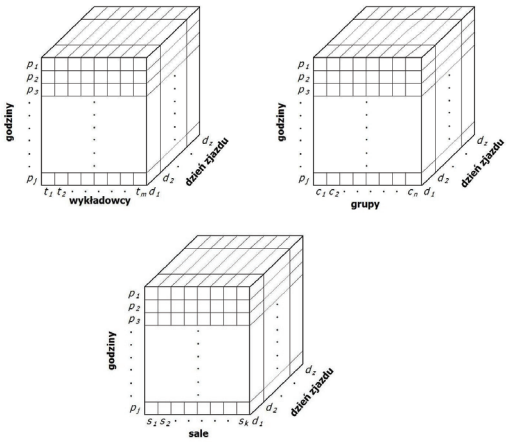
Wymienione powyżej dane przechowywane są w postaci tablic powiązanych ze sobą określonymi zależnościami, przedstawionymi na rys. 2.



Rys. 2. Struktura danych i powiązania między nimi

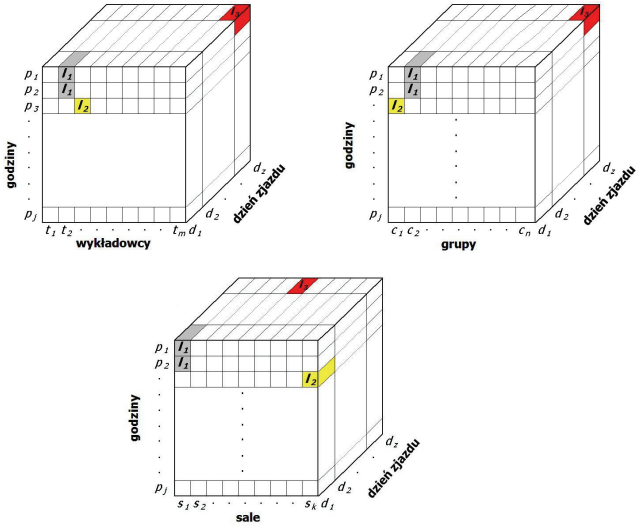
Przyjęta jednostka czasowa jest równa 15 minutom.

W omawianej implementacji do przechowywania poszczególnych planów cząstkowych wykorzystane zostały trzy macierze trójwymiarowe T, C i S. Pierwsza z nich zawiera informacje o planach wykładowców t, druga –grup c, trzecia - sal s.

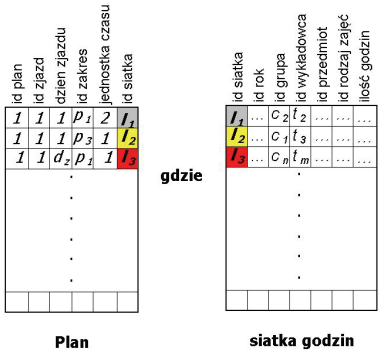


Rys. 3. Macierze przechowujące plany cząstkowe:
a) wykładowców, b) grup, c) sal

Tabele te są odrębnymi strukturami, łączy je jednak powiązanie z tabelą siatka godzin. Umieszczenie wylosowanej pozycji z macierzy siatka godzin (id siatka) w jednej z macierzy T, C lub S powoduje jednocześnie umieszczenie tego elementu również w analogicznych miejscach pozostałych dwóch macierzy. Z uwagi na fakt, że każdy element id siatka jednoznacznie definiuje wykładowcę, grupę i salę, których dotyczy, w poszczególnych komórkach macierzy T, C, S zapisywana jest tylko jedna wartość. W innym wypadku konieczne byłoby umieszczanie w komórkach macierzy większej ilości informacji. Rys. 4. i 5 ilustrują sposób zapisu informacji w macierzach T, C i S oraz tabeli Plan.



Rys. 4. Sposób zapisu informacji w macierzach T, C, S. Element l_i oznacza i-tą wartość id siatka



Rys. 5. Sposób zapisu poszczególnych pozycji generowanego planu zajęć na podstawie danych z macierzy T, C, S z rys. 4

Celem działania programu jest wygenerowanie planu lekcji o jak najmniejszej wartości funkcji oceny. Przy jej ustalaniu analizowany jest stopień spełnienia przez wygenerowane rozwiązanie ograniczeń określonych w punkcie 3.2. Zaimplementowany algorytm uniemożliwia stworzenie rozkładu zajęć łamiącego którekolwiek z ograniczeń ciężkich (H1)-(H7). Do wyznaczenia wartości funkcji oceny uwzględniane są więc tylko ograniczenia lekkie (S1)-(S3). Za każde złamanie ograniczeń przyznawany jest punkt karny. Wartość funkcji oceny wygenerowanego planu jest równa sumie wszystkich punktów karnych poszczególnych planów cząstkowych.

Proces tworzenia rozkładu zajęć rozpoczyna się od rozwiązania początkowego generowanego losowo. W trakcie każdej iteracji przeszukiwana jest przestrzeń rozwiązań X będąca zestawem rozwiązań spełniających ograniczenia (H1)-(H7).

Zbadanie całego sąsiedztwa rozwiązania x jest kosztowne obliczeniowo. Z tego powodu w programie zastosowana została strategia listy potencjalnych obiektów. Każda określona kombinacja zajęcia-wykładowca-grupa-sala (id siatka) jest wybierana przypadkowo jednak przypadkowość jest ograniczona. Rozwiązanie x′ musi spełniać ograniczenia ciężkie wobec czego nie jest konieczne przeszukiwanie całego sąsiedztwa N(x).

W pamięci programu przechowywana jest lista tabu oraz atrybuty zaakceptowanych ruchów. Zapisywana jest także liczba powtórzeń, dla której ruch ma pozostać zakazany. Przy każdej kolejnej iteracji liczba ta jest zmniejszana, a gdy przyjmie wartość równą 0 ruch zostaje usunięty z listy. Do wykonania wybrany zostaje natomiast ruch generujący rozwiązanie x' z najlepszą wartością funkcji oceny lepszą niż wartość funkcji oceny najlepszego otrzymanego dotychczas rozwiązania.

Po wykonaniu ruchu, rozwiązanie z najlepszą wartością funkcji oceny w nowym regionie zostaje zapisane w pamięci. Jeżeli rozwiązanie lepsze od zapisanego nie zostanie otrzymane po ustalonej liczbie powtórzeń, zapisane rozwiązanie będzie badane ponownie (intensyfikacja). Dzięki temu poszukiwanie skupia się na sąsiadach dobrych rozwiązań. Jeżeli rozwiązanie lepsze niż najlepsze regionalne nie zostanie znalezione w ciągu określonej liczby iteracji, algorytm powróci do najlepszego regionalnego rozwiązania.

Wszystkie opisane powyżej kroki algorytmu są wykonywane do momentu spełnienia jednego z kryteriów końca: znalezione zostanie rozwiązanie z wartością funkcji oceny równą 0 lub ilość iteracji osiągnie zadaną wartość.

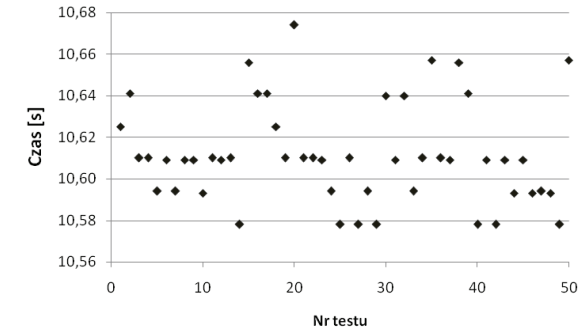
4. WYNIKI

Eksperyment został podzielony na dwie części. W pierwszej z nich dokonano pomiaru czasu niezbędnego do wygenerowania planu lekcji w sposób losowy bez użycia algorytmów optymalizacyjnych. Zbadano skuteczność metody oraz jakość otrzymanego rozkładu zajęć. Pomiar skuteczności przeprowadzony został poprzez uruchomienie aplikacji 100 razy i określeniu ilości poprawnie wygenerowanych planów zajęć. Czas generacji rozkładów oraz ich jakość określony został na podstawie 50 poprawnie wylosowanych planów. W drugiej części eksperymentu dokonano analogicznych pomiarów dla planów lekcji utworzonych przy zastosowaniu algorytmu TS. Przyjęty okres tabu był równy 5 bez możliwości warunkowego wykonania zakazanego ruchu. Porzucenie aktualnie przeszukiwanego sąsiedztwa na rzecz nowych obszarów następowało po upływie 10 iteracji. Wartości wag w_r funkcji oceny miały wartość równą 0.01, zarówno dla cząstkowych wartości funkcji oceny związanych z planami poszczególnych wykładowców, jak i poszczególnych grup. Wprowadzenie jednakowych wartości wag powoduje, że każdy czynnik odgrywa taką samą rolę w ocenie jakości planu. Maksymalna ilość iteracji wynosiła 500. Plan generowany był dla studiów dziennych.

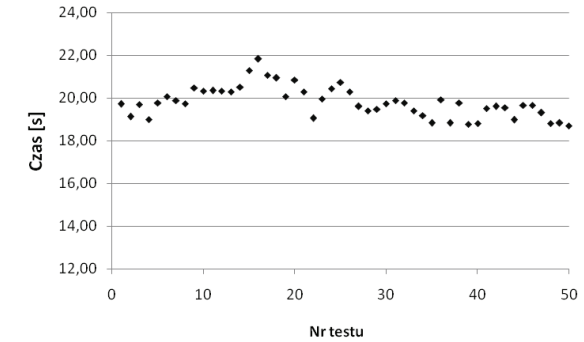
Wyniki eksperymentu omówionego w punkcie 3.6 zostały zilustrowane na rys. 6-8.

Czas niezbędny do losowego wygenerowania planu zajęć jest krótszy dla procesu losowego (rys. 6).

a)



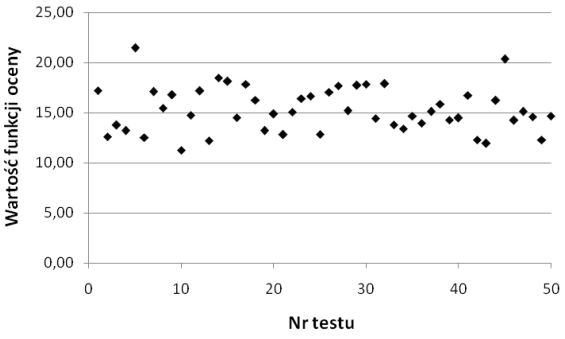
b)



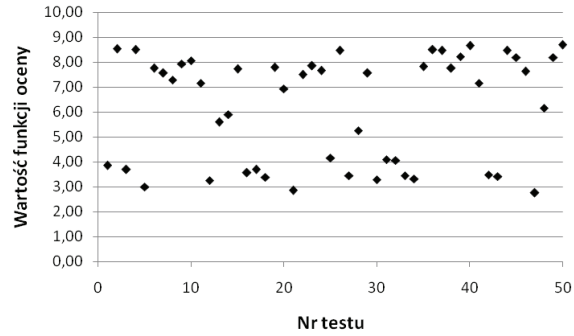
Rys. 6. Czas niezbędny do wygenerowania planu zajęć: a) generowanie losowe, b) z wykorzystaniem TS

Pierwsze wygenerowane rozwiązanie staje się rozwiązaniem końcowym, niezależnie od wartości funkcji oceny, co znajduje swoje potwierdzenie na rys. 7 a, b. Jakość planu wynikowego otrzymanego przy wykorzystaniu algorytmu TS jest średnio dwa razy lepsza niż w przypadku tworzenia go w sposób losowy.

a)



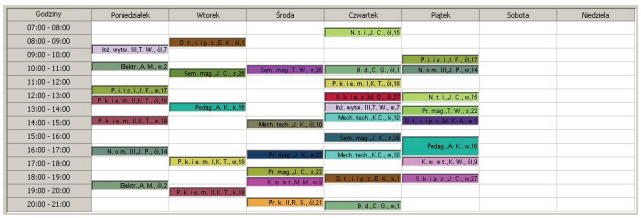
b)



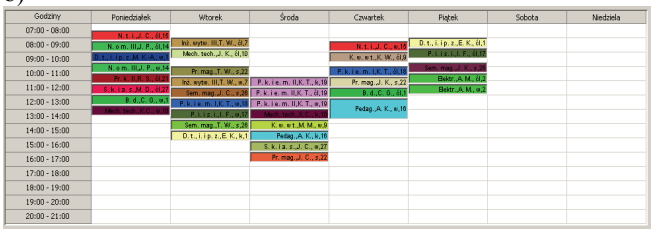
Rys. 7. Jakość wygenerowanego planu zajęć: a) generowanie losowe, b) z wykorzystaniem TS

Wartość funkcji oceny ma decydujący wpływ na ostateczną postać wynikowego planu zajęć. Różnice przedstawione w postaci wykresów na rys. 7 a i b wyraźnie widać na graficznej wersji harmonogramów.

a)



b)



Rys. 8. Przykładowe plany zajęć: a)generowanie losowe, b) z wykorzystaniem TS

5. PODSUMOWANIE

Przedstawione powyżej wyniki dowodzą, że zastosowanie algorytmu TS wpływa na znaczne zwiększenie jakości ostatecznego rozwiązania w stosunku do wyników generacji losowej. Dzięki zastosowaniu systemu ograniczeń możliwe jest sprawdzenie wszystkich dostępnych wariantów opracowywanego planu i określenie, który z nich w największym stopniu spełnia określone z góry założenia.

Literatura

1. Aladağ Ç. H,. Hocaoğlu G., *A Tabu Search Algorithm to Solve a Course Timetabling Problem*, Hacettepe Journal of Mathematics and Statistics, vol. 36(1), 2007, s.53-64
2. Glover F., Laguna M., *Tabu Search*, Kluwer Academic Publishers, Norwell, MA, 1997, ISBN:079239965X, s. 53
3. Glover, F., *Tabu Search — Part I*, ORSA Journal on Computing, vol. 1,no 3, 1989, s. 190-206
4. Glover, F., *Tabu Search — Part II*, ORSA Journal on Computing, vol. 2,no 1, 1990, s. 4-32

Studia i Materiały Informatyki Stosowanej, Tom 2, Nr 2, 2010
str.67-72

WYBRANE PROBLEMY BEZSTRATNEJ KOMPRESJI OBRAZÓW BINARNYCH

Urszula Wójcik, Dariusz Frejlichowski

Zachodniopomorski Uniwersytet Technologiczny
Wydział Informatyki
ul. Żołnierska 49, 71-210 Szczecin
e-mail: uwojcik@wi.zut.edu.pl, dfrejlichowski@wi.zut.edu.pl

Streszczenie: *W artykule przedstawiono podstawowe pojęcia i algorytmy związane z bezstratną kompresją obrazów binarnych. Dokonano przeglądu istniejących algorytmów bezstratnej kompresji, które są lub mogą być stosowane do obrazów binarnych, takich jak: kodowanie Huffmana, Golomba, Rice'a, kompresja JBIG, kodowanie długości sekwencji, kodowanie arytmetyczne oraz wybrane algorytmy predykcyjne.*

Słowa kluczowe: *Kompresja bezstratna, obrazy binarne*

A discussion on selected problems of lossless binary digital images compression

Abstarct: *In the paper some basic notions and algorithms devoted to the problem of lossless digital binary images compression were provided. A brief survey on existing lossless algorithms was performed, concentrated on the ones that are or can be applied to binary images, e.g. Huffman, Golomb, Rice, JBIG, RLE, arithmetic coding, as well as the selected predicting approaches.*

Keywords: *Lossless compression, binary images*

1. WPROWADZENIE

Niniejsze opracowanie, dokonujące przeglądu pojęć, metod i algorytmów z zakresu bezstratnej kompresji obrazów, jest próbą wstępnego ogarnięcia tego szerokiego obszaru. O zakresie i popularności problemu kompresji obrazów, w tym bezstratnej, może świadczyć mnogość pozycji książkowych jemu poświęconych (m.in. [1-5]).

W kolejnych rozdziałach niniejszego artykułu przedstawione zostaną wybrane teoretyczne podstawy kompresji, najważniejsze pojęcia związane z bezstratną jej wersją oraz przegląd miar jakości stosowanych do jej oceny. Trzeci rozdział dokonuje przeglądu istniejących metod kompresji bezstratnej obrazów w kontekście ich przyszłego zastosowania dla poszczególnych typów obrazów binarnych. W punkcie tym zawarta jest wstępna ocena porównawcza metod kompresji obrazów binarnych. Celem prowadzonych badań jest wskazanie zależności pomiędzy obiektem badań (obrazami binarnymi należącymi do różnych dziedzin zastosowań), a wyborem najskuteczniejszej metody kompresji.

2. PODSTAWOWE POJĘCIA

Kompresja jest sposobem reprezentacji informacji w zwięzłej postaci, na podstawie znajomości jej własności strukturalnych. Jedna z definicji mówi, że ([5]): „kompresja danych to proces przekształcenia pierwotnej reprezentacji sekwencji danych w reprezentację o mniejszej liczbie bitów". Za twórcę pojęcia kompresji uważa się Claude'a Shannnona, który zajmował się metodami przesyłania i przechowywania danych. W tatach czterdziestych rozwinął on dziedzinę wiedzy zwaną teorią informacji ([6]). Chociaż można wskazać wiele grup metod kompresji danych, np. na bazie charakteru kompresowanych danych lub stosowanego podejścia, najczęściej wyróżnia się dwa jej rodzaje: stratną i bezstratną.

Kompresja bezstratna odznacza się tym, że zrekonstruowany ciąg danych jest identyczny z sekwencją źródłową, a więc nie mamy do czynienia z utratą informacji. Ten rodzaj kompresji ma zastosowania głównie tam, gdzie wymagana jest wierna rekonstrukcja danych m.in. w obrazach medycznych, satelitarnych, historii operacji finansowych na kontach bankowych, czy kompresji tekstu. Natomiast kompresja stratna dopuszcza pewną utratę informacji, co nie pozwala na odtworzenie danych źródłowych z dokładnością do jednego bitu (tak jak w poprzednim przypadku), a jedynie w sposób przybliżony. Dzięki temu można uzyskać lepszy stopień kompresji, co znajduje zastosowanie na przykład w kompresji multimediów.

Głównymi cechami współczesnych metod kompresji są zdolność wyboru ilości, jakości i postaci informacji wyjściowej oraz elastyczność. Algorytm powinien realizować takie zadania przy możliwie skromnych kosztach obliczeniowych i sprzętowych. Metody kompresji można oceniać według różnych kryteriów. Najważniejsze to:

- **złożoność**;
- **szybkość działania**;
- **pamięć potrzebna do implementacji**;
- **rozmiar kompresji** - proporcja liczby bitów reprezentujących dane przed i po kompresji, wyrażona ułamkiem bądź procentowo;
- **stopień kompresji CR** (ang. *compression ratio*) – stosunek liczby bitów potrzebnych do reprezentacji danych przed kompresją do liczby bitów potrzebnych do reprezentacji danych po kompresji, wyrażonej w postaci *n:1*;
- **średnia bitowa BR** (ang. *bit rate*) – średnia liczba bitów potrzebna do reprezentowania pojedynczego symbolu z alfabetu źródłowego;
- **procent kompresji CP** (ang. *compression percentage*) - stopień kompresji wyrażony w formie danych procentowych. Procent kompresji jest określany wyrażeniem ([5]):

$$CP = \left(1 - \frac{1}{CR}\right) * 100\%$$

- **współczynnik kompresji** – dostarcza informacji o ilości usuniętej nadmiarowej informacji, czyli określa w jakim stopniu dane źródłowe zostały zredukowane ([8]):

$$k = \frac{Lwe - Lwy}{Lwe}$$

Przy tym: L_{we} – rozmiar danych wejściowych, a L_{wy} – rozmiar danych wyjściowych;

- **entropia wejścia**, czyli współczynnik chaosu zbioru, określa „podatność" danego zbioru na kompresję. Entropia (w teorii informacji) jest średnią ilością informacji, za pomocą której dane źródło może być zakodowane w ustalonym modelu probabilistycznym. Entropia opisywana jest wzorem ([4]):

$$E_{we} = -\sum_{i=1}^{n} P(z_i) * \log_2(P(z_i))$$

Przy tym: $P(z_i)$ – prawdopodobieństwo zajścia zdarzenia z_i, natomiast n – ilość różnych znaków.
Ponadto musi być spełniony warunek ([8]):

$$\sum_{i=1}^{n} P(z_i) = 1$$

- **minimalny rozmiar danych po kompresji**, możemy obliczyć za pomocą wzoru ([8]):

$$L_{min} = \frac{L_{we} * E_{we}}{8[bitów]}$$

- **minimalny współczynnik kompresji**, możemy obliczyć za pomocą wzoru ([8]):

$$k_{min} = \frac{Lwe - Lmin}{Lwe}$$

3. PRZEGLĄD METOD BEZSTRATNEJ KOMPRESJI OBRAZÓW BINARNYCH

Ogólnie bezstratne metody kompresji możemy podzielić na takie, które muszą posiadać wiedzę o danych wejściowych przed rozpoczęciem procesu kodowania oraz takie, które kodują dane na bieżąco lub dostosowują się już w procesie kodowania (tzw. metody adaptacyjne). Uproszczony schemat procesu kompresji metodami bezstratnymi pokazano na rys.1.

Rysunek. 1 Uproszczony schemat kompresji bezstratnej, źródło: [8].

Wybór odpowiedniej metody do konkretnego zadania nie jest łatwy, gdyż niektóre algorytmy cechują się lepszymi współczynnikami kompresji, inne zaś dużą szybkością kodowania i dekodowania. Niektóre metody są bardzo czułe na błędy spowodowane niedokładnym przesłaniem określonej partii danych. Czasami wystarczy zmiana nawet jednego bitu, aby dalsza partia danych była błędnie zdekodowana, podczas gdy w innych podejściach zmiana nawet całego bajtu powoduje mało znaczące straty. Przed podjęciem decyzji o wyborze sposobu kompresji ważną sprawą jest precyzyjne określenie typu danych wejściowych.

W kolejnych podrozdziałach przedstawiony zostanie krótki opis wybranych metod, które są lub mogą być stosowane w kompresji obrazów binarnych. Warto przy tym zaznaczyć, że obrazy binarne mogą być przed kompresją traktowane różnorodnie. Można się skupić na pojedynczych wartościach pikseli (0 lub 1) albo też łączyć je w większe grupy, np. po 8, 16, itd. i dopiero wtedy poddawać je kodowaniu.

3.1. Kodowanie Huffmana

Omawiany algorytm został zaproponowany w 1952 przez Davida Huffmana jako część rozwiązania zadania na zajęciach dla studentów. Metoda ta jest jedną z najbardziej znanych technik kodowania binarnego stosowaną w kompresji obrazów. Jest ona oparta na rozkładzie prawdopodobieństwa wystąpienia znaków. Kody generowane przez ten algorytm nazywane są kodami Huffmana i są to kody prefiksowe, będące optymalnymi dla przyjętego modelu probabilistycznego ([2]).

Kodowanie Huffmana jest algorytmem kompresji statycznej. Oznacza to, że program kompresujący musi przynajmniej raz przejrzeć cały kompresowany plik w celu stworzenia histogramu. Kodowanie Huffmana jest sposobem dopasowania zapisu znaków do rodzaju kompresowanych danych. Często wykorzystywane jest jako ostatni etap w różnych systemach kompresji stratnej oraz bezstratnej (np. w kompresji JPEG jako końcowy etap przetwarzania).

Algorytm kodowania wykorzystuje strukturę drzewa binarnego do wyznaczenia słów kodowych, których długość jest odwrotnie proporcjonalna do prawdopodobieństw p_i wystąpienia poszczególnych symboli s_i, oraz do kodowania znaków zmienną długością kodu. Konstrukcja drzewa binarnego rozpoczyna się od liści w kierunku korzenia. Algorytm Huffmana polega na łączeniu dwóch liści o najmniejszych wagach w poddrzewo z wierzchołkiem rodzica o wadze równej sumie wag liści. Następnie wśród liści i wierzchołków poddrzew wyszukiwane są i łączone kolejne wierzchołki, które mają najmniejsze wagi. W ten sposób tworzone są kolejne poziomy poddrzewa, do momentu znalezienia rodzica każdego z wierzchołków drzewa. Istnieją również metody adaptacyjne, które tworzą drzewo w czasie kodowania i nie potrzebują czasochłonnego procesu przeglądania całego pliku.

Na efektywność kompresji wpływa przede wszystkim częstość występowania znaków. Kodowanie Huffmana cechuje się uniwersalnością i wynikami bliskimi optymalnym. Bardzo dobrze radzi sobie ono z grafiką, czy tekstem.

Niestety operacja kompresji i dekompresji jest stosunkowo czasochłonna, gdyż wykonywane są operacje na strukturze drzewa. Kolejna wada to duża czułość na zniekształcenia przesyłanej zakodowanej informacji: wystarczy zmiana jednego bitu, aby spowodować nieodwracalne błędy podczas dekompresji. Dlatego też w kodowaniu metodą Huffmana należy stosować dużą liczbę zabezpieczeń przed błędami (np. kompresowanie mniejszych bloków i zapis sumy kontrolnej po każdym bloku), co zmniejsza współczynnik kompresji. W celu zwiększenia efektywności kodowania, stosuje się efekt polegający na rozszerzeniu alfabetu źródła. Utrudnia to jednakże realizację etapu modelowania.

3.2. Kody Golomba

Kod Golomba (G_m) jest unarnym kodem symboli, który wykorzystuje kod dwójkowy stałej długości. Jest to kod o nieskończonym alfabecie źródła, który opisywany jest dodatnią liczbą całkowitą m, zwaną rzędem kodu ([5]). Kody Golomba - Rice'a służą do kodowania liczb całkowitych przy założeniu, że im większa liczba, tym mniejsze prawdopodobieństwo jej wystąpienia.

Słowo kodu Golomba składa się z dwóch części:
- przyrostka, czyli odległości d symbolu (wyrażonej słowem kodu dwójkowego stałej długości) od początku przedziału ($B_m(d)$);
- przedrostka, czyli numeru przedziału symbolu zapisanego w kodzie unarnym ($U(u)$).

Słowo kodu Golomba można zapisać w postaci $G_m(i) = U(u)\ B_m(d)$, gdzie:
u – numer przedziału, $u=[i/m]$;
m – rząd kodu;
i – liczba całkowita ($i >= 0$), $i = u\ m+d$;
d – odległość od początku przedziału, $d = i \bmod m$.

Kod Golomba cechuje optymalna postać zakodowanej reprezentacji danych przy mniejszych kosztach obliczeniowych. Duża jest też efektywność, porównywalna z kodowaniem metodą Huffmana. Kod zawiera nieskończony zbiór słów kodowych. Stopień

kompresji zależy nie tylko od częstości wystąpień symboli, ale też od ich „ustawienia". Kod Golomba ma zastosowanie m.in. w metodzie odwracalnej kompresji JPEG-LS ([16]), oraz w metodzie adaptacyjnie dobieranych kodów Golomba ([15]).

3.3. Kody Rice'a

Kod Rice'a jest odmianą kodu Golomba, którą można traktować jako dynamiczny kod Golomba. Modyfikacja polega tu na dzieleniu zbioru liczb na podprzedziały o stałej długości (zwanej rzędem kodu), która jest potęgą dwójki. Kodowanie Rice'a z adaptacyjnym dobieraniem rzędu jest stosowane między innymi w algorytmie kompresji bezstratnej JPEG-LS oraz FLAC. Metoda ta uchodzi za bardzo wygodną w implementacji i pozwala na prostą i szybką realizację procesu kompresji.

3.4. Kodowanie arytmetyczne

Początki kodowania arytmetycznego sięgają lat sześćdziesiątych. Za jego twórców uważa się C.E. Shannona i P. Eliasa ([6]). Kodowanie arytmetyczne jest użyteczne, gdy rozmiar alfabetu jest mały, prawdopodobieństwa są zróżnicowane oraz gdy proces kodowania i modelowania są od siebie oddzielone. W opisywanym kodowaniu generowany jest unikalny znacznik, kodujący cały ciąg wejściowy, który odpowiada prawdopodobieństwu wystąpienia kodowanego ciągu, wyrażonego binarnie.

Zasada kodowania arytmetycznego opiera się na kodowaniu komunikatu jako liczby z lewostronnie domkniętego przedziału [0,1). Znajdowanie tej liczby polega na zwiększaniu jej precyzji poprzez stopniowe zawężanie przedziału, przy użyciu prawdopodobieństwa aktualnie przetwarzanego znaku ([1]).

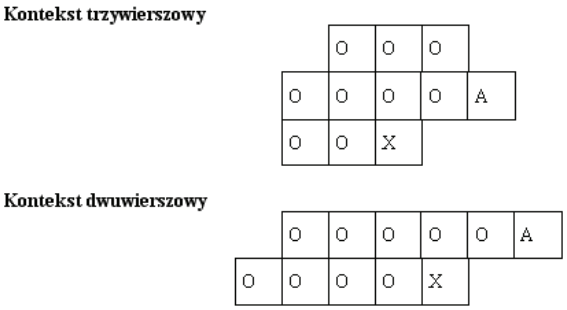
Algorytm kodowania arytmetycznego działa na dowolnym źródle danych, zarówno dla danych tekstowych, obrazów, jak i dla plików dźwiękowych. Jest to bardzo efektywna metoda kompresji, która daje lepsze wyniki niż metoda Huffmana (od kilku do kilkunastu procent). Jednakże efektywność algorytmu zależy jedynie od rozkładu prawdopodobieństwa symboli. Ponadto nie wykorzystuje on cech właściwych dla poszczególnych typów danych. Przy praktycznych rozwiązaniach stosuje się dodatkowo struktury drzewa i kopca binarnego, co przyspiesza nawet kilkunastokrotnie działanie algorytmu.

W metodzie tej nie ma potrzeby stosowania rozszerzonego alfabetu źródła (tak jak w przypadku metody Huffmana). Pozwala to zbudować model na podstawie bogatszej statystyki danych źródła. Wygenerowanie znacznika dla konkretnego ciągu nie wymaga wyznaczania bądź pamiętania znaczników innych ciągów. W porównaniu z algorytmem Huffmana, kodowanie jest znacznie wolniejsze, natomiast dekodowanie jest operacją dużo prostszą. Podejście bywa jednak uważane za mało praktyczne – kosztowna arytmetyka zmiennoprzecinkowa i złożony algorytm dają w efekcie dużą złożoność obliczeniową. W przypadku długiej sekwencji danych źródłowych liczba kodowa może przekroczyć pojemność zmiennych (rejestrów) danej implementacji (procesora). Można przy tym zauważyć nieznaczne pogorszenie współczynnika kompresji w przypadku kodowania obrazów binarnych. Więcej informacji na temat kodowania arytmetycznego można znaleźć m. in. w [17].

3.5. Kompresja JBIG

Standard kodowania obrazów binarnych JBIG (*ang. Joint Bi-level Image experts Group)* stanowi połączenie dwóch algorytmów ([2]): algorytmu kompresji bezstratnej i transmisji progresywnej kodowania obrazów binarnych.

Sposób kodowania piksela jest określony na podstawie „kontekstu", na który składa się 9 bitów (oznaczonych na rys.2. literą **O),** położonych w sąsiedztwie kodowanego piksela. Znajdują się one na lewo i/lub wyżej (zakładamy kompresję od lewej do prawej i z góry na dół), zatem piksele te znane są dekoderowi w momencie dekodowania. Wartości kolejnych pikseli kodowane są w oparciu o kontekst pikseli sąsiednich. Dodatkowo na kontekst składa się dziesiąty bit (litera **A** na rys.3), również na lewo i/lub w górę, którego położenie względem kodowanego symbolu może być dowolne (ale takie samo w obrębie całego obrazka; jest ono przesyłane wraz z plikiem) – służy to uwzględnieniu periodyczności w obrazku, nie mieszczących się w kontekście.



Rysunek. 2 Rodzaje kontekstów. Oznaczenia: O – stały element kontekstu, A – zmienny element kontekstu, X – kodowany piksel, źródło: [2].

Po kompresji obrazu następuje drugi etap – transmisja progresywna. Polega ona na tworzeniu małych, mniej dokładnych wersji obrazu. Obrazy te mogą być zastosowane np. do szybkiego przesłania podglądu pliku. Zadanie to realizuje się poprzez zastępowanie kolejnych bloków pikseli jednym pikselem, wyznaczonym jako średnia wartość czterech pikseli składających się na dany blok. Schemat ten można stosować wieloetapowo, tworząc w ten sposób kilka obrazów, każdy o mniejszej rozdzielczości. Obrazy te nazywane są poziomami.

Liczba koderów używanych przy kompresji JBIG wynosi od 1024 do 4096. Zależy ona od tego, czy wykonywane jest kodowanie o niskiej, czy wysokiej rozdzielczości. W standardzie JBIG wszystkie 1024 kodery są wariantami kodera arytmetycznego nazywanego koderem QM ([2]). Kompresja JBIG umożliwia uszczegóławianie obrazu, bez zwiększenia rozmiaru danych (względem obrazu oryginalnego).

3.6. Kompresja JBIG2

Standard JBIG2 opiera się na tym samym algorytmie kodowania co JBIG. Jest jednakże rozszerzeniem swojego poprzednika, w którym różne fragmenty obrazu mogą być kodowane różnymi metodami.

Koder JBIG2 dzieli obraz binarny na tzw. **regiony**. Wyróżnia się **regiony tekstowe**, **regiony w półtonie** oraz **regiony ogólne**. Kodowanie bloków ogólnych opiera się na tym samym algorytmie kodowania arytmetycznego, którego używano w JBIG ([5]). Regiony tekstowe kompresowane są metodą słownikową. W segmencie umieszczany jest słownik symboli, który utworzony jest z map bitowych. Kodowanie polega na zapisaniu informacji o położeniu znaków oraz indeksu do elementu w słowniku symboli. W przypadku kompresji stratnej symbole nie muszą dokładnie pokrywać się z danymi na obrazie. Dzięki temu można osiągnąć mniejszy rozmiar słownika. Procedura kodowania regionów w półtonie jest także oparta na metodzie słownikowej. Kodowanie przebiega podobnie jak w przypadku segmentów tekstowych, z tym, że słownik zamiast symboli przechowuje wzorce półtonu o określonym rozmiarze.

W przeciwieństwie do JBIG, w omawianym algorytmie możliwy jest wybór metody kodowania, która daje najlepszą kompresję dla przetwarzanego typu danych. Dodatkowo można stosować różne metody kompresji dla różnych fragmentów obrazu.

3.7. CALIC

Istnieją dwa rodzaje adaptacyjnych modeli predykcji: adaptacja w przód (ang. *forward adaptation*) oraz wstecz (ang. *backward adaptation*). Adaptacja w przód wykorzystuje dostęp do całej sekwencji źródłowej oraz importuje parametry modelu predykcji, które dynamicznie zmodyfikowano w trakcie kodowania. Algorytm kompresji (dekompresji) jest niesymetryczny. Dane wejściowe są dzielone na kilka części. W przypadku obrazów są to często bloki prostokątne lub sąsiednie fragmenty danych. Adaptacja wstecz polega na poszukiwaniu efektywnego modelu na podstawie informacji zakodowanej lub zdekodowanej. W tym przypadku model predykcji nie wykorzystuje całej informacji źródłowej, przez co jest znacznie uboższy.

Metoda CALIC (ang. *context-based adaptive image coder*) wykorzystuje pełne sąsiedztwo najbliższych punktów podczas kodowania pikseli obrazu, co zapewnia większą efektywność kompresji obrazów.

3.8. JPEG-LS

Kodowanie stosowane w formacie JPEG-LS polega na przesyłaniu par liczb – wartości próbki i liczby powtórzeń tej wartości dla kolejnych próbek. Podejście takie jest szczególnie efektywne dla obrazów nienaturalnych, np. zrzutów ekranu komputera. Podczas modelowania kontekstu określany jest rozkład prawdopodobieństw użyty do kodowania aktualnej wartości. Istnieją dwa rodzaje trybów kodowania:

- tryb regularny, jeśli jest małe prawdopodobieństwo wystąpienia serii próbek o identycznej wartości;
- tryb sekwencji próbek identycznych, jeśli jest duże prawdopodobieństwo wystąpienia serii próbek o takiej samej wartości.

3.9. Metoda kodowania długości serii (RLE)

Metoda kodowania długości serii - RLE (*ang*. *Run-Length Encoding*) należy do grupy metod predykcyjnych. Jest to prosta metoda bezstratnej kompresji danych. Algorytm polega na wyszukiwaniu ciągów tych samych znaków i opisywaniu ich za pomocą licznika powtórzeń. Metoda najbardziej skuteczna okazuje w kompresji obrazów binarnych. Stosuje się ją również do kodowania dokumentów przesyłanych za pomocą faksu (najczęściej w jednym z formatów: PCX, BMP, TGA). Sprawdza się wszędzie tam, gdzie występuje duża powtarzalność tych samych elementów następujących po sobie, takich jak kolor tła, czy inne duże obszary jednolitego zabarwienia. Metoda charakteryzuje się dużą szybkością kodowania i dekodowania.

Istnieje wiele metod zwiększania efektywności kompresji RLE. Przykładem jest wektorowe kodowanie długości serii (ang. *vector run-length encoding*).

Kodowane są tutaj serie bloków pikseli o wymiarach $m \times m$, zamiast pojedynczych pikseli binarnych. Można dzięki temu uzyskać nawet kilkudziesięcioprocentową poprawę efektywności kodowania w porównaniu z jednowymiarową metoda RLE ([5], [10]).

4. PODSUMOWANIE

W artykule dokonano przeglądu podstawowych pojęć i metod z zakresu bezstratnej kompresji obrazów binarnych. W przyszłej pracy omawiane algorytmy będą badane na rzeczywistych klasach obiektów binarnych, takich jak semacody, odciski palców, czy obrazy radarowe. Dobór materiału badawczego celowo koncentruje się na aktualnych problemach badawczych, takich jak bezpieczeństwo czy różnego rodzaju zastosowania techniczne. Na rys. 3 pokazano przykłady danych testowych, należące do pięciu klas zastosowań. Będą one poddawane działaniu różnych algorytmów kompresji, by sprawdzić, jaki wpływ na ich efektywność ma charakter danych wejściowych.

Nazwa grupy	Dane
 Odciski palców	Ilość próbek: 51 Rozdzielczość: 256 x 256 px Typ plików: bmp Źródło pochodzenia: fingerp Wielkości plików: 8,06 KB Głębia w bitach: 1
 Obrazy radarowe	Ilość próbek: 402 Rozdzielczość: 228 x 228 px oraz 225 x 225 px Typ plików: bmp Źródło pochodzenia: zbiory własne Wielkości plików: 51,8 KB oraz 51,1 KB Głębia w bitach: 1
 Semacody	Ilość próbek: 50 Rozdzielczość: 216 x 216 px oraz 184 x 184 px Typ plików: bmp Źródło pochodzenia: QR-Code Generator Wielkości plików: 137 KB oraz 100 KB Głębia w bitach: 1
 Skany tekstu	Ilość próbek: 30 Rozdzielczość: 5104 x 7016 px Typ plików: bmp Źródło pochodzenia: zbiory własne Wielkości plików: 34,1 MB Głębia w bitach: 1
 Kontury kształtów	Ilość próbek: 110 Rozdzielczość: 128 x 128 px oraz 100 x 100 px Typ plików: bmp Źródło pochodzenia: zbiory własne Wielkości plików: 17 KB oraz 10,8 KB Głębia w bitach: 1

Rysunek. 3 R Przykłady klas danych obrazów binarnych do testowania algorytmów kompresji, źródło: opracowanie własne.

Literatura

1. Drozdek A., "Wprowadzenie do kompresji danych", WNT, Warszawa 1999
2. Sayood K., "Kompresja danych wprowadzenie", RM, Warszawa 2002
3. Skarbek W., „Metody reprezentacji obrazów cyfrowych", Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993
4. Skarbek W. (red.), „Multimedia algorytmy i standardy kompresji", Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998
5. Przelaskowski A., "Kompresja danych, podstawy, metody bezstratne, kodery obrazów", Wydawnictwo BTC, Warszawa 2005
6. Shannon C. E., "A Mathematical Theory of Communication", Bell System Technical Journal, 1948, vol. 27, pp. 379-423
7. Starosolski R., "Algorytmy bezstratnej kompresji obrazów", Studia Informatica, 2003, vol. 24, Nr 1(52), ss. 138-158
8. Heim K., "Metody kompresji danych", Wydawnictwo MIKOM, Warszawa 2000
9. Huffman D. A., "A method for the construction of minimum-redundancy codes", Institute of Radio Engineers , 1952, vol. 40, no.9, pp. 1098-101
10. Wang, Y., Wu, J.-M., "Vector run-length coding of Bi-level images", Proc. of Data Compression Conference, 1992, pp. 289 – 298
11. Knuth D. E., "Dynamic Huffman coding", Journal of Algorithms, 1985, vol. 6, pp. 163-180
12. Starosolski R., "Algorytmy bezstratnej kompresji obrazów", Studia Informatica, 2002, Vol. 23, Nr 4(51), ss. 277-300
13. Pennebaker W. B., Mitchell J. L., "JPEG: Still Image Data Compression Standard", Springer, 1993
14. Stateczny A., "Nawigacja porównawcza", Gdańskie Towarzystwo Naukowe, Gdańsk 2001
15. Seroussi, G., Weinberger, M.J., "On adaptive strategies for an extended family of Golomb-type codes", Proc. of the Data Compression Conference, 1997, pp. 131 – 140
16. Weinberger M., Seroussi G., Sapiro G., "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS", Hewlett-Packard Laboratories Technical Report No. HPL-98-193R1, 1998
17. Langdon G.G., "An introduction to arithmetic coding", IBM Journal Research and Development, 1984, vol. 28, pp. 135-148

UNIWERSYTET KAZIMIERZA WIELKIEGO

Wydział Matematyki, Fizyki i Techniki
ul. Chodkiewicza 30

fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264

www.simis.ukw.eu.pl

Studia i Materiały Informatyki Stosowanej

Czasopismo młodych pracowników naukowych, doktorantów i studentów

Wprowadzenie

W dzisiejszej technice i naukach ścisłych trudno obejść się bez wspomagania procesów badawczych i technologicznych metodami i narzędziami informatycznymi. Nowe czasopismo Studia i Materiały Informatyki Stosowanej skierowane jest do naukowców wykorzystujących szeroko rozumiane metody informatyczne w swoich pracach badawczych na polu różnych dyscyplin technicznych. Pozwoli ona skonsolidować różne w swoich obszarach badawczych dyscypliny techniczne z punktu widzenia podobnych metod komputerowych wspomagających te dziedziny. Do aktywnego udziału zaproszeni są wszyscy pracownicy, doktoranci i studenci, szczególnie ci reprezentujący szeroko rozumianą technikę, którym bliska naukowo jest dziedzina Informatyki Stosowanej. Szczególne zaproszenie kierujemy do młodych pracowników naukowych, do osób prowadzących koła naukowe. Naszym pragnieniem jest aby SiMIS stało się forum gdzie doktoranci, czy studenci-dyplomanci opublikują swoje pierwsze publikacje, często z promotorem, i wkroczą na drogę działalności naukowej.

Zakres tematyczny

- Różne dziedziny Informatyki (*Sztuczna Inteligencja, Kryptologia, Inżynieria Oprogramowania, Algorytmika, Programowanie Równoległe, Sieci komputerowe, itp.)*,
- Komputerowe wspomaganie projektowania CAx,
- Symulowanie komputerowe procesów w technice,
- Symulacje i modelowanie w naukach ścisłych,
- Przetwarzanie danych pomiarowych,
- Mechatronika,
- Wszelkie zastosowania technik komputerowych w nauce i technice.

Ważne daty

31.10.2010 Ostateczny termin nadsyłania artykułów

30.11.2010 Ostateczna akceptacja artykułów i zakończenie recenzji

20.12.2010 Rozpoczęcie procesu wydawniczego

UNIWERSYTET KAZIMIERZA WIELKIEGO

Wydział Matematyki, Fizyki i Techniki
ul. Chodkiewicza 30

fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264

www.simis.ukw.eu.pl

Dla autorów

Studia i Materiały Informatyki Stosowanej są czasopismem ukazującym się co pół roku. Możliwe jest wyłącznie nadsyłanie artykułów, których wcześniej nie publikowano, wszystkie artykuły są recenzowane.

- Studia i Materiały Informatyki Stosowanej są wydawnictwem recenzowanym
- Autorem może być nauczyciel akademicki, doktorant lub student oraz każda inna osoba zainteresowana tematyką.
- Format artykułu w załączniku, grafika będzie konwertowana do skali szarości
- Język publikacji: polski z obowiązkowym streszczenie w j.angielskim, lub całość artykułu w j.angielskim.
- Zgłoszenie artykułu następuje przez wysłanie maila z wypełnionym plikiem „SiMIS2010 zgłoszenie" na adres: `jczerniak@ukw.edu.pl`
- Artykuły napisane z użyciem szablonu „SiMIS2010 szablon" prosimy przesłać na adres: `jczerniak@ukw.edu.pl`

Kontakt

- dr inż. Jacek Czerniak, kontakt: tel. (052) 341 92 64, p.216A, jczerniak@ukw.edu.pl
- dr inż. Marek Macko, kontakt: tel. (052) 341 93 30, p.04, mackomar@ukw.edu.pl

KAZIMIERZ WIELKI UNIVERSITY IN BYDGOSZCZ

Faculty of Mathematics, Physics
and Technical Science
ul. Chodkiewicza 30

fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264

www.smacs.ukw.edu.pl

Studies and Materials in Applied Computer Science

Journal of young researchers, PhD students and students

Introduction

Modern engineering and science cannot do without computer-aided research and technologic processes. The new journal, Studies and Materials in Applied Computer Science, is aimed at scientists who, in their research work on diverse engineering branches, use computer science methods in their broad sense. With that journal, it will be possible to consolidate different engineering branches that are aided by common computer-related methods. We would like all employees, PhD students, students and specially those who represent engineering in its broad sense and whose research works are closely related to Applied Computer Science to feel invited to actively cooperate with the journal. We address that invitation especially to young researches and to leaders of scientific circles. We would like SMACS to become a forum where PhD students or MSc. candidates publish their first papers, often together with their thesis supervisor, and thus they enter the world of scientific activity.

The editorial staff is going to publish original papers of experts of world renown as well as young scientists both in Polish and in English language.
An important part of our publishing programme will be special editions prepared by world-renown authors.
We also plan to publish review papers presenting the state of knowledge in a given branch of science as well as special papers written by most outstanding Polish and foreign scientist, who present original, sometimes controversial attitude towards challenges that are put out to computer science in its broad sense by the modern world. Those specially ordered papers will be published in a series called “Meetings with Masters”. We particularly want young computer science specialists who enter the world of science or begin their professional career to learn opinion of well-known experts of their branch. Hence the “master-student” relationship that has been present in science and in culture since the dawn of time will be continued by our journal.

Range of topics

- Different Computer Science branches (*Artificial intelligence, Cryptology, Software Engineering, Theoretical Computer Science, Concurrent programming, Computer networks, etc.)*,
- Computer Aided Design CAx,
- Computer Aided technological process simulation,
- Simulation and modelling in science,

KAZIMIERZ WIELKI UNIVERSITY IN BYDGOSZCZ
Faculty of Mathematics, Physics and Technical Science
ul. Chodkiewicza 30
fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264
www.smacs.ukw.edu.pl

- Measurement data processing,
- Mechatronics,
- Any other computer technology applications in science and engineering.

Important deadlines

31.10.2010	Final deadline to receive papers
30.11.2010	Final acceptance of papers and completion of review process
20.12.2010	Beginning of the publishing process by publisher

To authors

Studies and Materials in Computer Science is a journal that comes out twice a year. Papers sent to us by their authors cannot be published before. All the articles will be subject to review. By now, research studies were published under the name Applied Computer Science Symposium, a compact publication in monograph form and it included papers presented during meetings and symposiums.

- Studies and Materials in Applied Computer Science is a publication subject to review
- The author of a paper can be an employee, a PhD student, a student or any other person interested in this subject.
- See the attachment with the format of the paper. Be aware that graphics will be converted into greyscale.
- Language of a publication: Polish plus mandatory summary in English or an entire paper in English.
- Papers should be submitted by sending an e-mail including completed file „SMACS2009_request" to the address: jczerniak@ukw.edu.pl
- Papers written using the „SMACS2009_template" template should be sent to the address: jczerniak@ukw.edu.pl

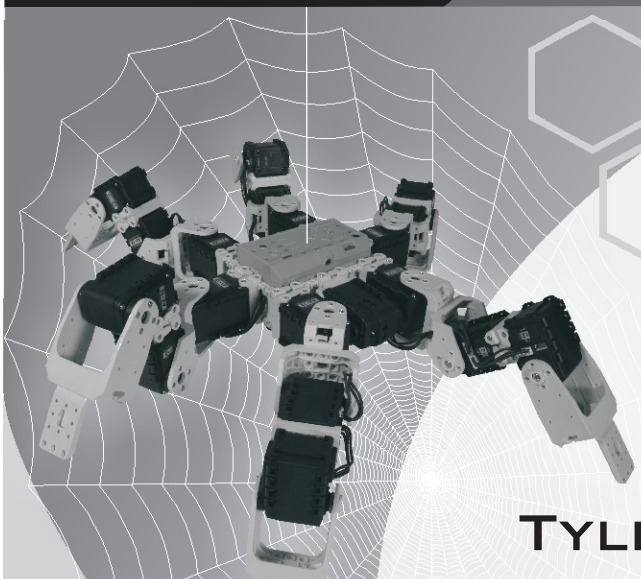
Contact:

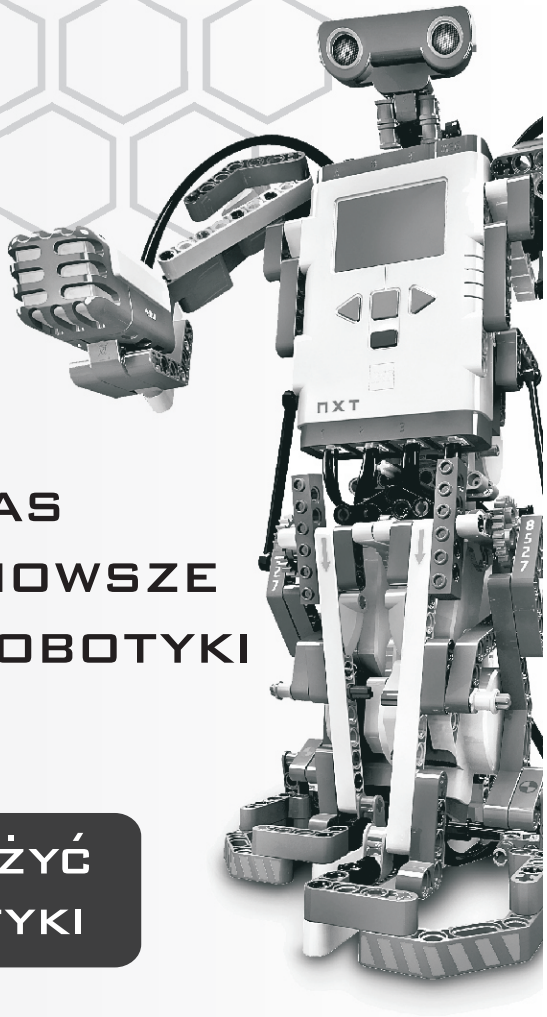
- PhD. eng. Jacek Czerniak, contact: Phone: (+4852)341-92-64, room 216A, jczerniak@ukw.edu.pl
- PhD. eng. Marek Macko, contact: Phone: (+4852)341-93-30, room 04, mackomar@ukw.edu.pl

Robotyka dla przyszłości

Www.roboSHOP.pl

sklep dla robotyków



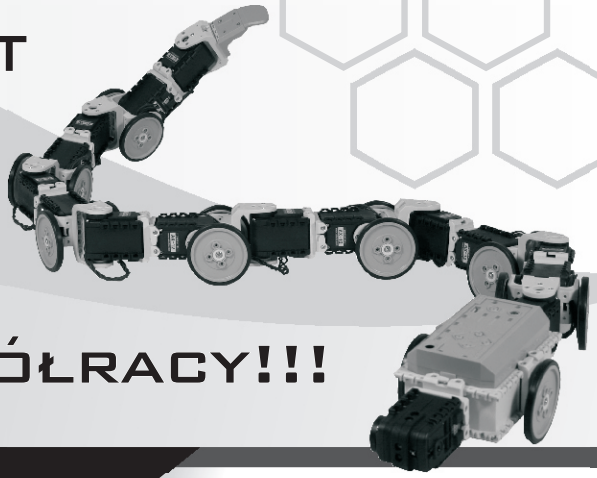


Tylko u Nas
Znajdziesz najnowsze
produkty ze świata robotyki

Jesteśmy w stanie wyposażyć
każde laboratorium robotyki

Szczególnie polecamy zestawy edukacyjne:

- BIOLOID
- LEGO Mindstorms NXT



Zapraszamy do współracy!!!

RoboSHOP.pl
ul. Trzy lipy 3 I 80-172 Gdańsk
tel./fax (058) 739 61 61 I biuro@roboshop.pl
www.roboshop.pl

ROBO SHOP.PL