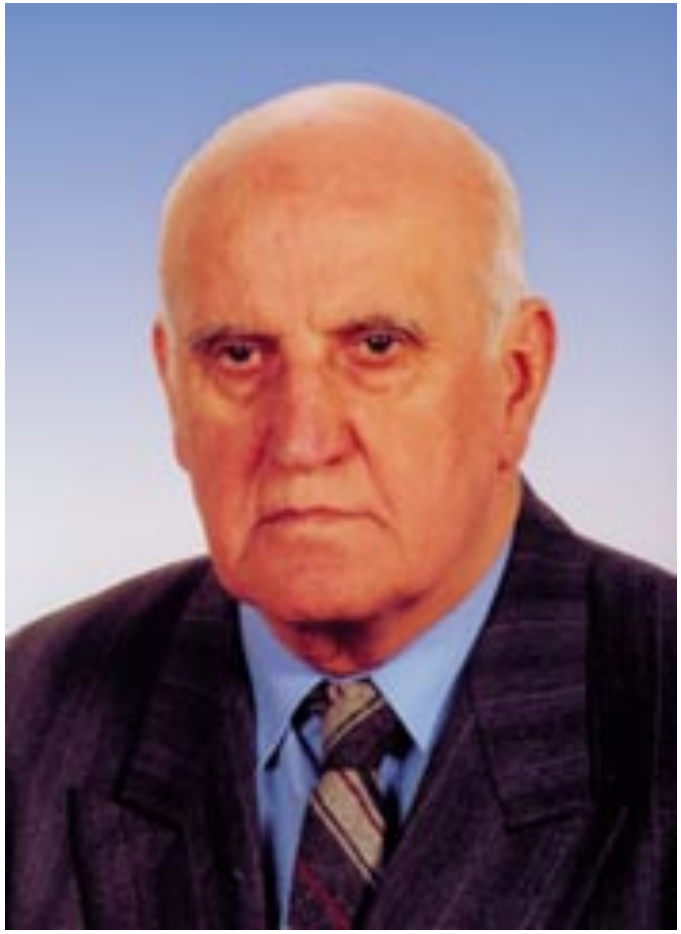
Teoria Zbiorów Przybliżonych

dr inż. Jacek Czerniak

prof. dr hab. inż. Zdzisław Pawlak
Uniwersytet Warszawski, Polska

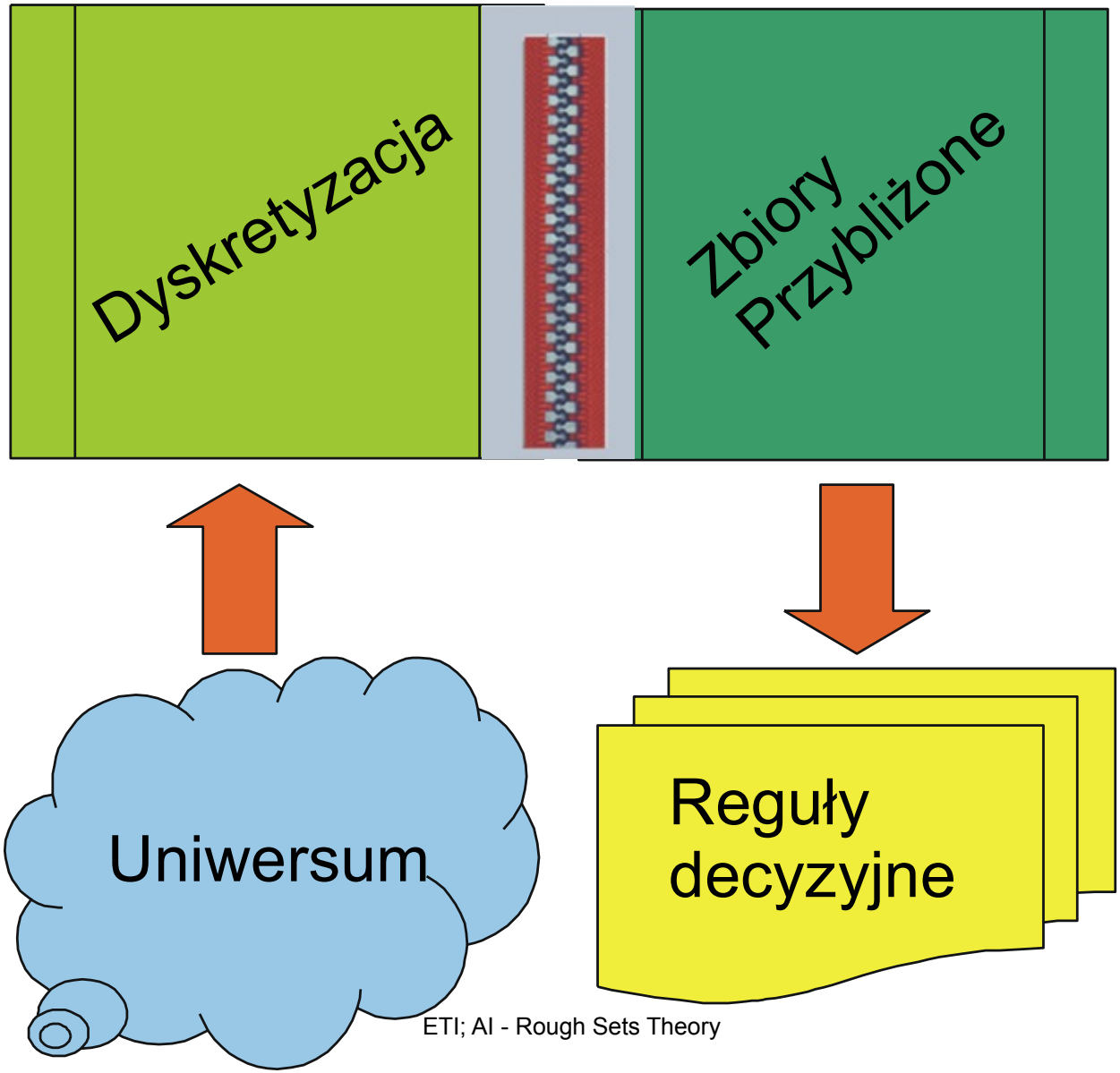


- Twórca teorii zbiorów przybliżonych
- Członek rzeczywisty Polskiej Akademii Nauk
- Laureat Nagrody Państwowej Zespołowej II stopnia
- Odznaczony Krzyżami: Kawalerskim i Oficerskim Orderu Odrodzenia Polski
- Doktor Honoris Causa Politechniki Poznańskiej

Teoria Zbiorów Przybliżonych

- Angielska nazwa *Rough Sets Theory*
- Pozwala na wykrywanie reguł w wycinkach świata rzeczywistego tzw. *uniwersum*
- Należy do dziedziny wiedzy zwanej *Knowledge Discovery in Databases*
- Dostarcza narzędzi matematycznych do formalnego opisu wiedzy, w tym wiedzy niepełnej i niedokładnej
- Wiedza rozumiana jest tu jako zdolność klasyfikowania obiektów badanego świata, a do jej reprezentacji proponuje się systemy informacyjne
- Algorytmy decyzyjne w teorii zbiorów przybliżonych budowane są w oparciu o ww. systemy informacyjne, a dokładniej o szczególny ich rodzaj zwany tablicami decyzyjnymi

Zbiory Przybliżone c.d.



Skrótowe przedstawienie problemu

Na początku rozważań posłużmy się wyjątkiem z definicji *Knowlage Discowery in Databases* podanej przez S.Nguyen'a „Typowy przebieg KDD składa się z następujących kroków:

- zgromadzenie i udostępnienie danych (ten proces jest ściśle związany z problematyką baz danych),
- przekształcanie i transformacja danych,
- wykrywanie reguł i wzorców danych,
- interpretacja i ilustracja wyników. ”

W dalszej części prezentacji zagadnienie to będzie dokładniej wyjaśnione.

Definicje

Jako główne problemy badawcze w KDD postrzega się zwykle reprezentację wiedzy i klasyfikacje obiektów. Problemy te można opisać w następujący sposób. Rozważmy uniwersum obiektów **Ư**. Obiekty z **Ư** są przydzielone do różnych kategorii zwanych klastrami decyzyjnymi wg pewnej nieznanej nam zasady. Znalezienie funkcji decyzyjnej, która odpowiada ze przypisanie każdemu obiektowi jego klasę decyzyjna, jest głównym celem procesu klasyfikacji. Problem klasyfikacji można opisać następująco:

"*Jeżeli dane jest uniwersum* **Ư** *oraz funkcja decyzyjna* **dec** *opisana na skończonym podzbiorze* $U \subset$ **Ư**. *Znaleźć taki algorytm decyzyjny, działający na zbiorze atrybutów* **A** *opisujących obiekt* **O**, *który potrafi z dużą dokładnością przypisać obiekt* **O** *do określonej klasy decyzyjnej*"

Tablica decyzyjna

Problem klasyfikacji może być opisany za pomocą *tablicy decyzyjnej*. Tablicą decyzyjną nazywamy strukturę **Ă**$=(U, A \cup \{dec\})$ gdzie

- U nazywa się zbiorem obiektów,
  $U = \{u_1, \ldots, u_n\}$
- A jest zbiorem atrybutów postaci
  $A : U \rightarrow \{a_1, \ldots, a_k\}$
- *dec* jest specjalnym atrybutem zwanym decyzją
  $dec : U \rightarrow \{d_1, \ldots, d_j\}$

Tabela decyzyjna c.d

Przykład tablicy decyzyjnej pokazany został w tabeli Tab1.

	a1	a2	...	dec
1	213	10	...	2
2	23	12	...	2
...	...	...	...	...
n	22	34	...	1

Tab 1. Przykład tablicy decyzyjnej.

Pojęcia szczegółowe

Wprowadźmy jeszcze kilka dodatkowych pojęć:

Klasy decyzyjne: dec definiuje podział uniwersum U

$$U = DEC_1 \cup \ldots \cup DEC_d \text{ gdzie}$$

$$DEC_K = \{x \in U : dec(x) = k\}$$

Zbiór DEC_i nazywamy i-tą klasa decyzyjną.

Rozróżnialność: Dane są obiekty $x,y \in U$ oraz zbiór atrybutów $B \subset A$. Mówimy, że x,y są rozróżnialne przez B wtedy i tylko wtedy, gdy istnieje $a \in B$ taki, że $a(x) \neq a(y)$. Tablica decyzyjna nazywamy *sprzeczną,* jeśli istnieją obiekty nierozróżnialne przez A, które należą do różnych klas decyzyjnych.

Pojęcia szczegółowe c.d.

Zbiór atrybutów $B \subset A$ nazywamy *reduktem* tablicy A wtedy i tylko wtedy, gdy

-dla dowolnych obiektów x,y $\in U$

jeżeli $dec(x) \neq dec(y)$ i x,y są rozróżnialne przez A,

to są również rozróżnialne przez B (B *zachowuje rozróżnialność A)*

-B jest nieredukowalny (tzn. żaden właściwy podzbiór B nie zachowuje rozróżnialności zbioru A)

Przykład tablicy decyzyjnej

Poniższa tabela przedstawia przykład tablicy decyzyjnej opisującej stan posiadania pewnej firmy developerskiej X.

Budynek	Ładna okolica	Bezpieczna okolica	Przy ruchliwej ulicy	Nowa dzielnica	Wynik finansowy
	A1	A2	A3	A4	D1
1	Ładna	Bezpieczna	Nie	Tak	Strata
2	Ładna	Bezpieczna	Nie	Nie	Strata
3	B. ładna	Bezpieczna	Nie	Tak	Zysk
4	Przeciętna	Supuper b.	Nie	Tak	Zysk
5	Przeciętna	Niezbyt b.	Tak	Tak	Zysk
6	Przeciętna	Niezbyt b.	Tak	Nie	Strata
7	B. ładna	Niezbyt b.	Tak	Nie	Zysk
8	Ładna	Super b.	Nie	Tak	Strata
9	Ładna	Niezbyt b.	Tak	Tak	Zysk
10	Przeciętna	Super b.	Tak	Tak	Zysk
11	Ładna	Super b.	Tak	Nie	Zysk
12	B. ładna	Super b.	Nie	Nie	Zysk
Nowy 1	B. ładna	Bezpieczna	Tak	Tak	?
Nowy 2	Przeciętna	Super b.	Nie	Nie	?

Teoria Zbiorów Przybliżonych (*TZP*)

Zanim przystąpimy do analizy tych danych, możemy sobie postawić pytania:

- czy tablica posiada jakiś redukt, który nie zawiera wszystkich atrybutów?

Odpowiedź na to pytanie jest pozytywna, gdyż takim reduktem byłby zbiór $\{a_1,a_2,a_4\}$. Oznacza to, że nie potrzebujemy wszystkich atrybutów do analizy. Ale powstają przy tym inne pytania:

- czy istnieje redukt zawierający 2 atrybuty?
- znaleźć redukt o najmniejszej liczbie atrybutów (tzw. redukt minimalny)

Ogólniej, chcemy znaleźć rozwiązanie dla następujących problemów związanych z reduktami:

- czy istnieje redukt zawierający *k* atrybutów?
- znaleźć redukt o najmniejszej liczbie atrybutów.

Przykładowe Zastosowania *TZP*

Poniżej przedstawiono zagadnienia opisane w literaturze przedmiotu do analizy których użyto zbiorów przybliżonych:

Informatyka	**Ekonomia**
Dobór optymalnego trasowania w sieciach WAN Rozwiązywanie konfliktów w modelach interakcji i kooperacji agentów Analiza nakładu pracy w projektach informatycznych	Wycena przedsiębiorstw Polityka kredytowa banku Bankructwa 66 nowojorskich firm
Medycyna	**Sterowanie**
Schorzenia wątrobowe Zakażenia układu moczowego Badania angiograficzne tętnic wieńcowych	Odwrócone wahadło Piec obrotowy do wypalania klinkieru Reaktor chemiczny
Poligrafia	**Inne dziedziny**
Dobór odpowiedniego typu płyt *CtP* na podstawie rodzajów dotychczasowych publikacji Analiza kosztów materiałów eksploatacyjnych w produkcji poligraficznej	Dane dotyczące hodowli kwiatów Irys Dane techniczne 55 łożysk tocznych Dane agregatów stosowanych przy produkcji betonu

Światowe ośrodki badań nad „Rough Sets Theory”

Obecnie na świecie pięć ośrodków prowadzi intensywne badania nad Rough Sets wliczając w to prace nad dyskretyzacją.

Są to grupy pod kierunkiem;

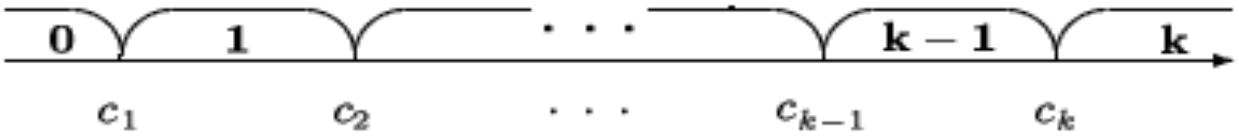
- prof. J.W. Grzymala-Busse, University of Kansas, USA
- prof. J.Komorowski, Norwegian University of Science and Technology, Norwegia,
- prof. R.Słowiński, prof. J.Stefanowski, Politechnika Poznańska,
- prof. A.Skowron, prof. Z.Pawlak, Uniwersytet Warszawski,
- prof. L.Polkowski, Polsko-Japońska Wyższa Szkoła Technik Komputerowych

Dyskretyzacja

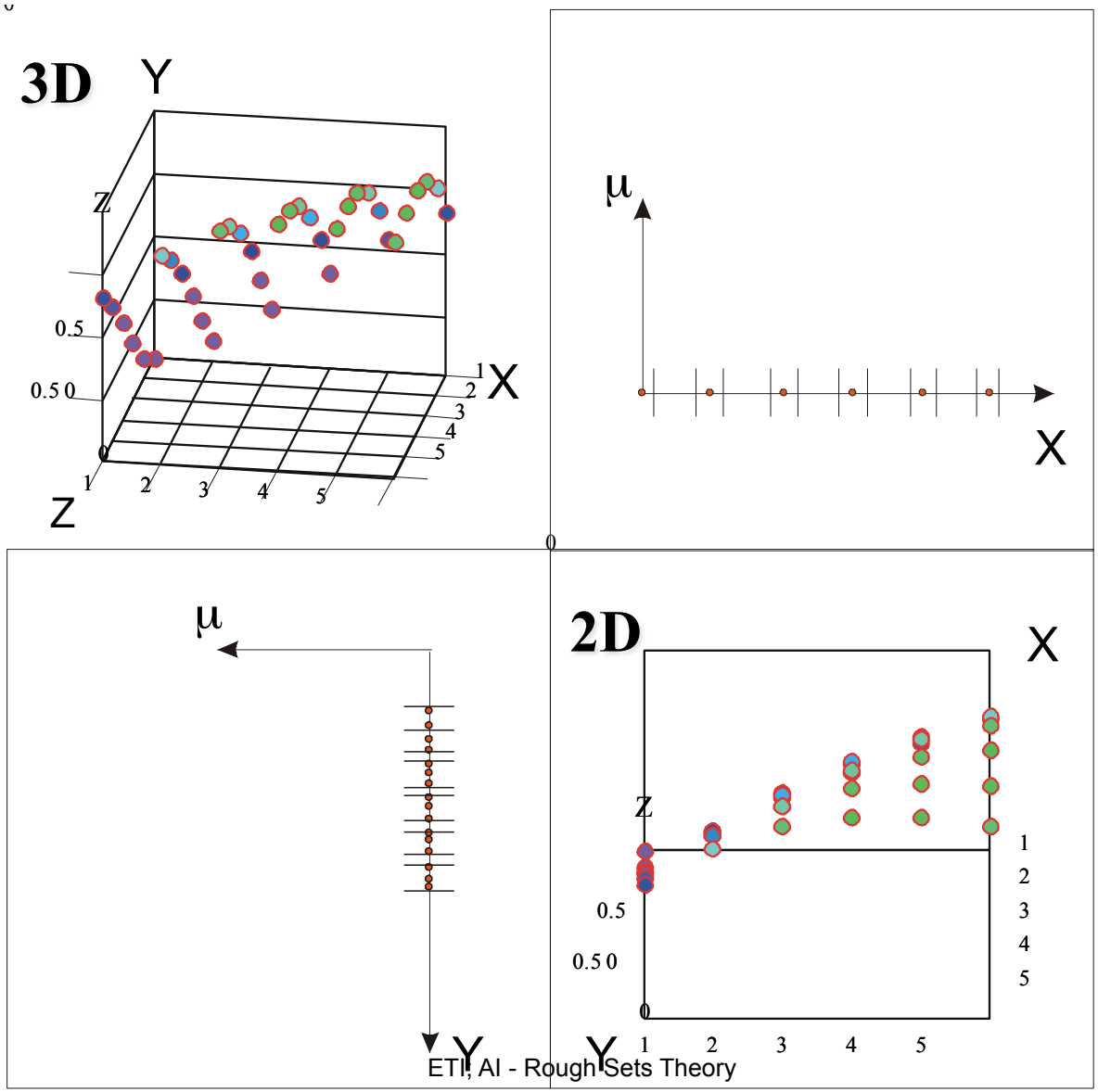
Na początku rozważań posłużmy się cytowanym już wyjątkiem z definicji *Knowledge Discovery in Databases* podanej przez S.Nguyen'a „Typowy przebieg KDD składa się z następujących kroków:

- zgromadzenie i udostępnienie danych,
- przekształcanie i transformacja danych,
- wykrywanie reguł i wzorców danych,
- interpretacja i ilustracja wyników. ”

Standardowo przyjmuje się że dyskretyzacja danych następuje zwykle na drugim i trzecim etapie. Dyskretyzacja atrybutu o wartościach rzeczywistych polega na znalezieniu zbioru końców przedziałów zwanych cięciami. Cięciem nazywamy każda parę (a,c), gdzie a jest atrybutem, zaś c jest dowolną wartością rzeczywistą.



Dyskretyzacja c.d.



Dyskretyzacja c.d.

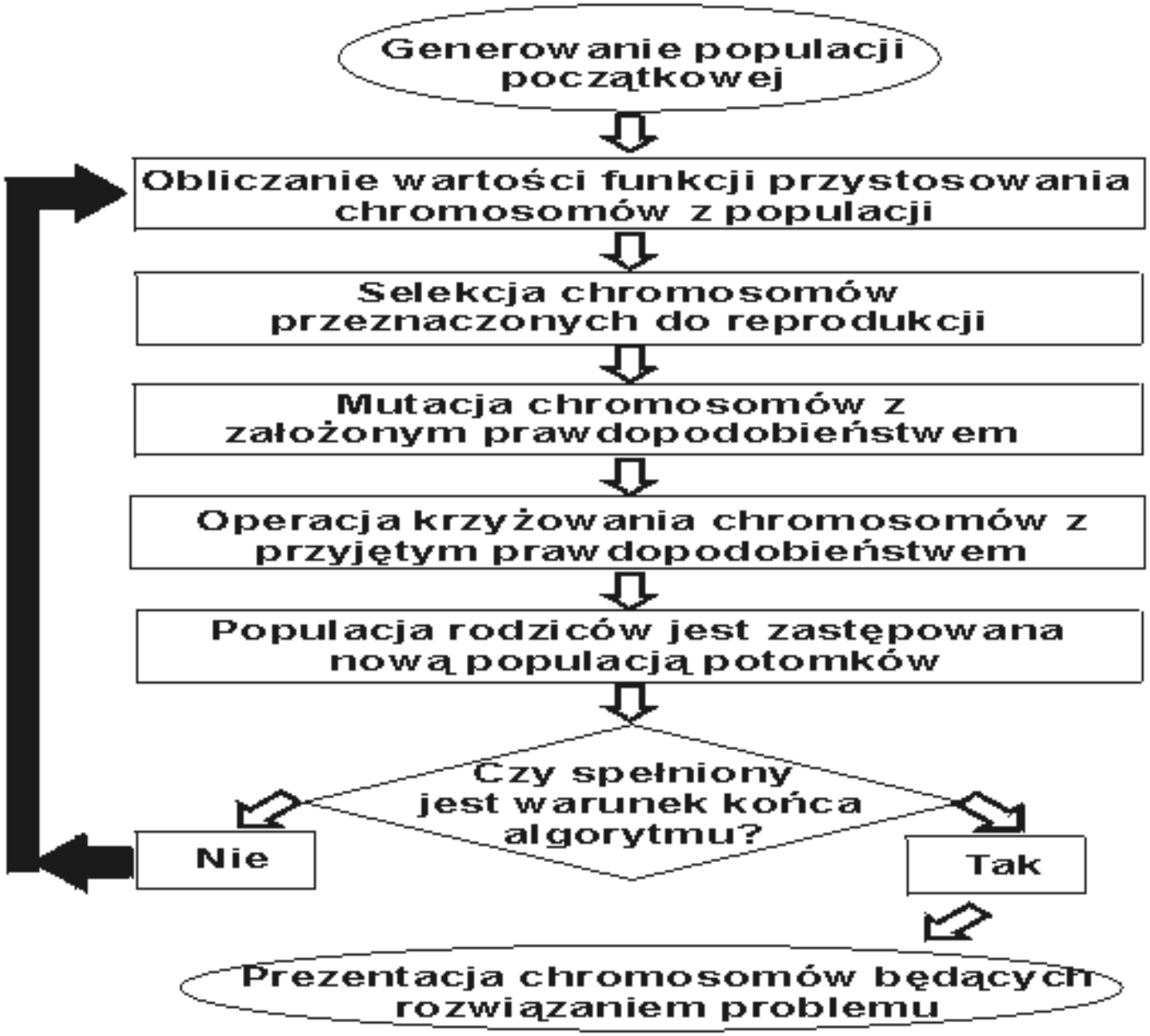
•Ze względu na złożoność problemu większość metod analizy tablic decyzyjnych ogranicza się do przypadku, gdy atrybuty mają mało możliwych wartości. W przypadku atrybutów o wartościach rzeczywistych, dane ulegają procesowi dyskretyzacji.

• Niestety, dyskretyzacji atrybutów rzeczywistych nie można przeprowadzać w dowolny sposób. Autorzy zajmujący się problematyką zbiorów przybliżonych (*ang. Rough Sets*) zwrócili uwagę iż zależnie od wybranej formy dyskretyzacji atrybutów otrzymują znacznie różniące się zbiory reguł decyzyjnych. W rozprawie przedstawiono przegląd współczesnych prac nad tym zagadnieniem.

Własna metoda rozwiązania problemu

- Przedstawienie metody LDGen
- Wykazanie przewagi metody LDGen nad dotychczasowymi
- Przedstawienie publikacji doktoranta

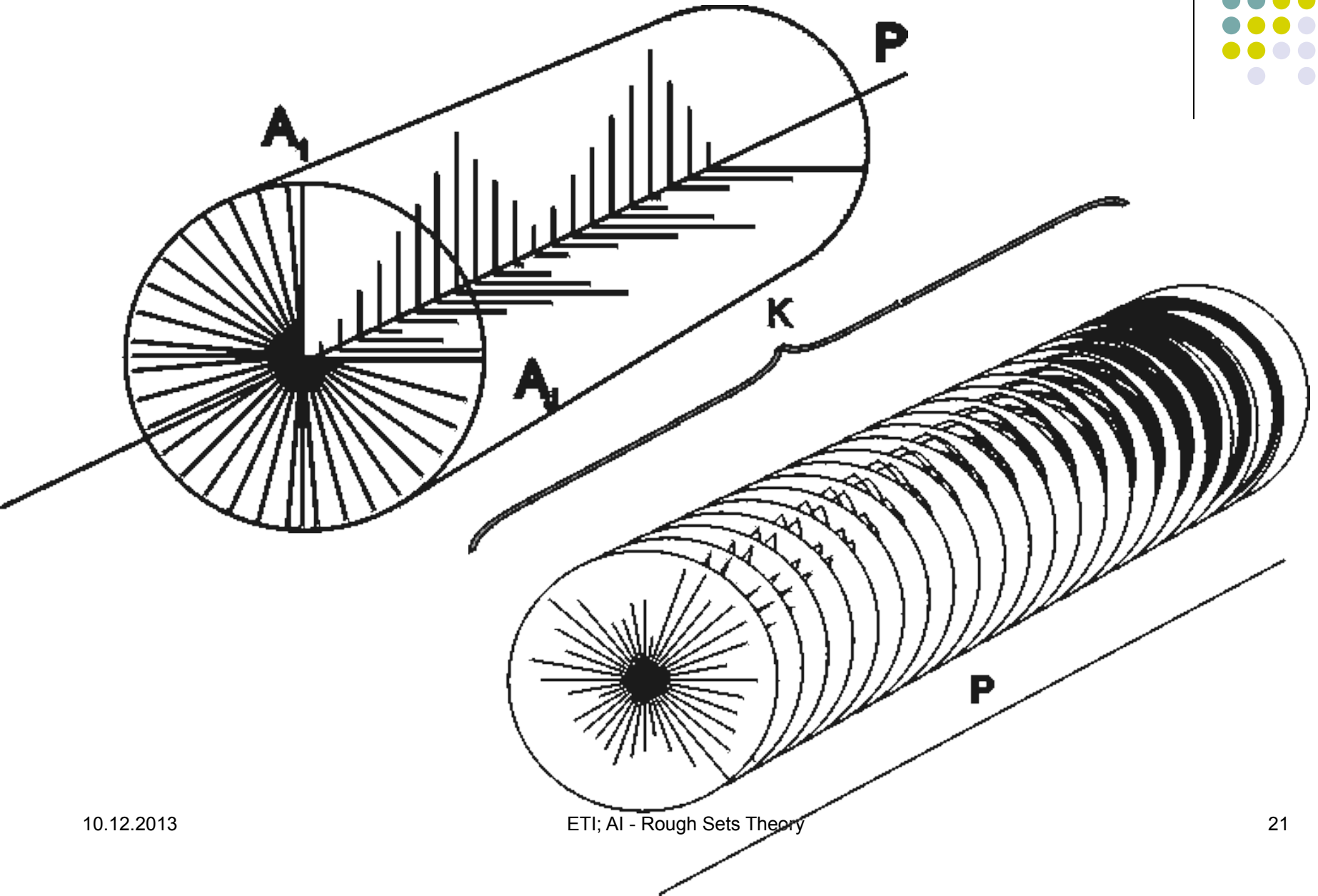
Metoda LDGen



Wprowadzenie

W celu pełniejszego i bardziej systematycznego przedstawienie poszczególnych etapów metody LDGen chciałbym zacząć od nakreślenia metody wizualizacji wielowymiarowej, której jestem autorem, a która to metoda stała się podwaliną mechanizmów algorytmu genetycznego, a w szczególności operatorów krzyżowania. Wspomniana metoda nosi nazwę „Rury Próbek”, a omówię ją posługując się schematem z kolejnego slajdu...

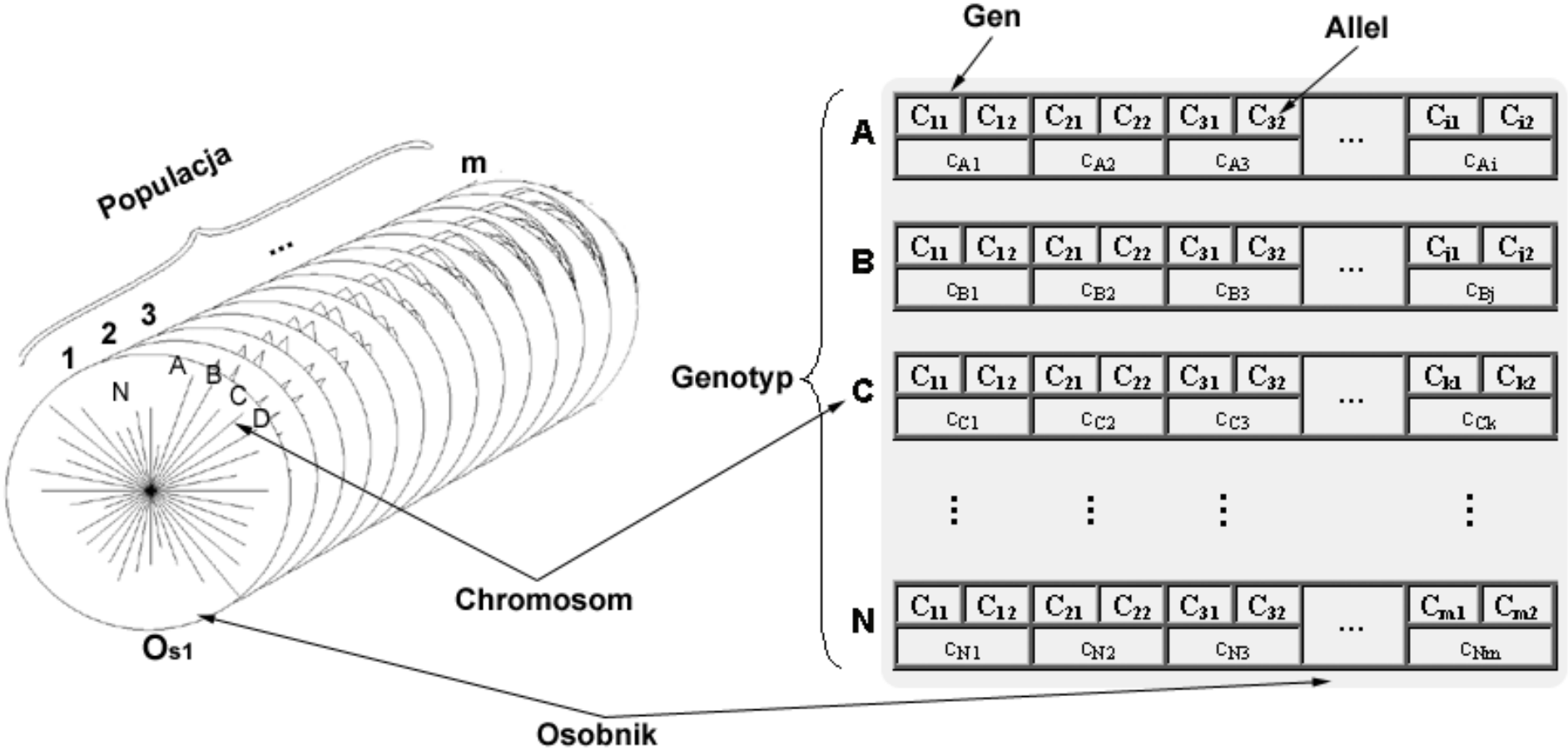
„Rura próbek”



Podstawowe pojęcia

- **Populacja** - skończony zbiór osobników,
- **Osobnik** - element populacji, stanowi rozwiązanie problemu czasem określany mianem punktu z przestrzeni poszukiwań, w algorytmie genetycznym występuje w postaci genotypu,
- **Genotyp** - struktura złożona z chromosomów, zbiór informacji opisujący osobnika,
- **Fenotyp** - zbiór konkretnych wartości odpowiadających danemu genotypowi, posiada go każdy osobnik,
- **Chromosom** - uporządkowany ciąg genów, zwykle tożsamy jest z pewnym ciągiem kodowym opisującym pewną własność osobnika,
- **Gen** - pojedynczy element c_{Ai} chromosomu A, opisującego najmniejszą część osobnika,
- **Allel** - wartość danego genu, jest częścią składową fenotypu

Struktury danych w LDGen



Przykład osobnika

$$Osobnik = \begin{bmatrix} (C_{11}, C_{12}, c_{A1}) & (C_{21}, C_{22}, c_{A2}) & (C_{31}, C_{32}, c_{A3}) & (C_{41}, C_{42}, c_{A4}) \\ (C_{11}, C_{12}, c_{B1}) & (C_{11}, C_{12}, c_{B2}) & & \\ (C_{11}, C_{12}, c_{C1}) & (C_{11}, C_{12}, c_{C2}) & & \\ (C_{11}, C_{12}, c_{D1}) & (C_{11}, C_{12}, c_{D2}) & & \\ (C_{11}, C_{12}, c_{E1}) & (C_{11}, C_{12}, c_{E2}) & & \\ (C_{11}, C_{12}, c_{F1}) & (C_{11}, C_{12}, c_{F2}) & & \end{bmatrix}$$

$$Fenotyp = \begin{bmatrix} (36,37,"n") & (37,38,"p") & (38,40,"g") & (40,\infty,"w") \\ (0.0,0.5,"nie") & (0.5,1.0,"tak") & & \\ (0.0,0.5,"nie") & (0.5,1.0,"tak") & & \\ (0.0,0.5,"nie") & (0.5,1.0,"tak") & & \\ (0.0,0.5,"nie") & (0.5,1.0,"tak") & & \\ (0.0,0.5,"nie") & (0.5,1.0,"tak") & & \end{bmatrix}$$

Format danych wejściowych

```
struct ciecia {
  float f_od;
  float f_do;
  String s_decyzje;
  String kod_cut;
};
```

Dane dotyczące „cięć”. Traktowane jako przedział, domyślnie lewostronnie domknięty. Dodatkowo również informacja z jakimi decyzjami związany jest przedział

```
struct chromosom {
  struct ciecia wartosc[max_cut];
  unsigned long int no_ch;
  float start;
  float stop;
  int ile_ciec;
  };
```

unikalny numer chromosomu
start domeny
stop domeny
cięcia pomiędzy start i stop

```
struct genotyp {
 struct chromosom fenotyp[max_gen];
  int  sila;
  unsigned long int no_os;
  int rodzic;
};
```

wartość funkcji dopasowania
unikalny numer osobnika
ile razy osobnik był rodzicem

Generowanie populacji startowej

Wygenerowanie populacji początkowej zależne jest od ilości i wartości atrybutów decyzyjnych. Następuje ono wyniku przejścia poniższych kroków;

- sortowanie próbek wg unikalnego zestawu decyzji
- dla każdej decyzji w każdym atrybucie osobno
  sortowanie atrybutów
- domyślnie jedna decyzja to jedna przedział ale jeżeli liczebność wartości atrybutów w decyzji jest większa niż 10% liczebności atrybutów to za każde zaczęte 10% jest dodatkowe cięcie.

Generacja populacji początkowej c.d.

• do każdego przedziału w tabeli `ciecia wartosc[max_gen]` w polu `s_decyzje` wpisywany jest łańcuch odpowiadający do jakiej decyzji przedział przynależy.

Przedziały w ww tabeli są posortowane. Każdorazowo wpisanie nowego przedziału dla decyzji wiąże się z możliwością wystąpienia trzech sytuacji.

• przedziały rozłączne, nie ma problemu ,

• przedziały zawierające się. Wtedy tworzy się dwa lub trzy przedziały rozłączne,

• przedziały zachodzące na siebie. Wtedy tworzy się dwa rozłączne przedziały zewnętrzne oraz dwa przedziały poprzez podzielenie na dwa części wspólnej.

Generowanie populacji c.d.

Każda klasa decyzyjna otrzymuje pewien "priorytet". Unikalny i stały podczas tworzenia jednego osobnika - dyskretyzacji. Priorytety nadawane są losowo z możliwością powtarzania w różnych przebiegach (dla różnych osobników). Potencjalne konflikty przy dodawaniu nowych cięć rozwiązywane są w zależności od priorytetu. Jeżeli nowe cięcie próbuje rozciąć stare to porównanie priorytetów skojarzonych z klasami i pośrednio z cięciami pozwala lub nie na modyfikację. Konkretnie wynik=[stary priorytet]/[nowy priorytet]*100, jeżeli wynik>=100 to dodaje się cięcie, jeżeli nie to następuje losowanie z przedziału [1,100] i jeżeli liczba jest większa niż wynik to nie nastąpi modyfikacja, a jeżeli mniejsza to nastąpi.

Obliczenie wartości funkcji przystosowania

W LDGen zastosowano wartość funkcji przystosowania mierząca względną ilość niesprzecznych opisów przykładów w dyskretyzowanej tabeli danych. Oznacza to zgodnie z teorią zbiorów przybliżonych, że ten sam układ wartości atrybutów nie daje różnych decyzji. Oczywistym jest fakt, iż najlepsze efekty eliminacji atrybutów osiągnięte są na niesprzecznych zbiorach danych dlatego też tak postawione kryterium wydaje się być optymalne.

Metoda selekcji osobników

Selekcja chromosomów przeznaczonych do reprodukcji odbywa się zgodnie ze zmodyfikowaną metodą koła ruletki. *De facto* proces selekcji przebiega zgodnie z poniższym schematem:

- obliczenie sumy wszystkich wartości funkcji przystosowania.

```
for(int i=0;i<max_pop;i++)
{
  suma=suma+ ptr_P[i].sila;
}
```

- przesunięcie w dalszy zakres wartości sumy. Dzieje się tak aby generator liczb pseudolosowych mógł zadziałać w sposób bardziej zbliżony do rzeczywistej losowości.

```
suma=suma*100;
```

- sortowanie chromosomów wg wartości funkcji przystosowania. Chodzi o uzyskanie porządku rosnącego.

Metoda selekcji osobników c.d.

- losowanie liczby z przedziału (0, suma) w celu wyodrębnienia osobnika.

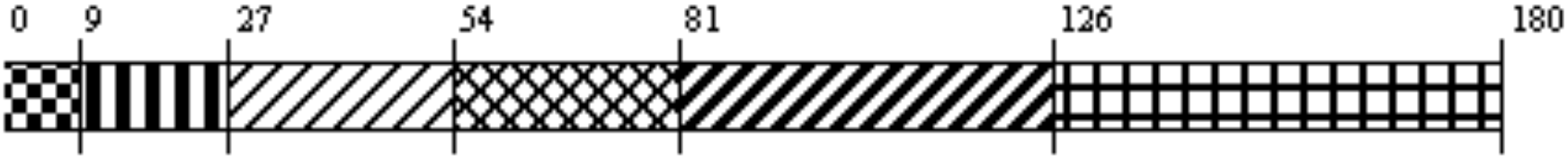
```
traf =random(suma);
```

- na koniec następuje wybranie osobnika wg poniższego wzoru.

```
for(i=0;i<max_pop;i++)
{
  suma=suma+ptr_P[i].sila;
  if(traf<=suma*100) //trafiles w wycinek
   {
      ptr_P[i].rodzic++;
      wynik= ptr_P[i];
      break;
   }
}
```

Metoda selekcji osobników c.d.

Należy przy tym dodać, że sytuacja przypomina np. oś odciętych naznaczona podziałką. Zaczynając od zera w stronę sumy następuje sprawdzenie czy wylosowana poprzednio liczba mieści się w przedziale. Jeżeli jest większa poszukiwanie jest kontynuowane, jeżeli nie oznacza to iż wyselekcjonowano osobnika do prokreacji.



Operacja mutacji

Mutacja odbywa się (lub nie) zależnie od założonego prawdopodobieństwa. W tym konkretnym przypadku założono prawdopodobieństwo wystąpienia mutacji zgodnie z sugestiami licznych autorów na poziomie `pm=0.01`. Mutacja chromosomów polega na losowym zwiększeniu ilości cięć. Konkretnie wygląda to tak że dla populacji złożonej z 20-tu wielowymiarowych rodzin atrybutów losowany jest jeden lub więcej osobników które zostaną zmutowane. Ilość wylosowanych chromosomów jest zależna od prawdopodobieństwa mutacji `pm` wg. następującego wzoru: `ile= INT(pm*licznosc)`.

Operacja mutacji c.d.

Następnie gdy wiadomo ilu osobników będzie mutowanych procedura toczy się jak poniżej;

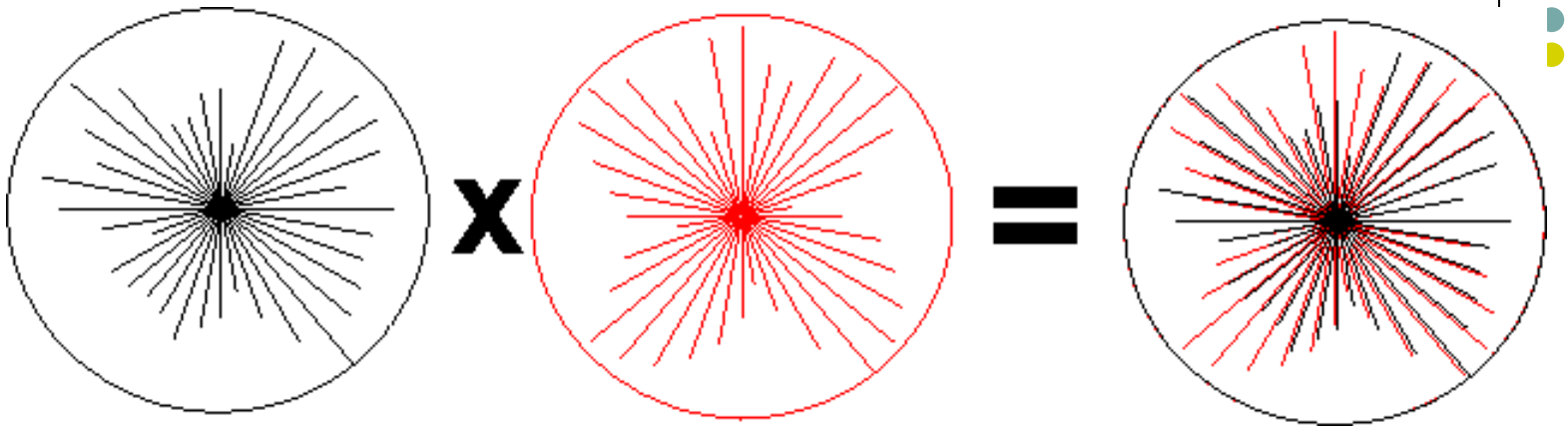
- wylosować osobnika do mutacji `traf=random(licznosc)`
- wyszukać osobnika wylosowanego transponując `traf` na dwie współrzędne lokalizujące
- odczytać ilość przedziałów dyskretyzacji z parametru `ile_ciec`
- wyliczyć liczbę naturalną jako 5% z ilości cięć odpowiednio zaokrąglając
- wylosować tyle przedziałów ile obliczono wyżej i podzielić je na pół.

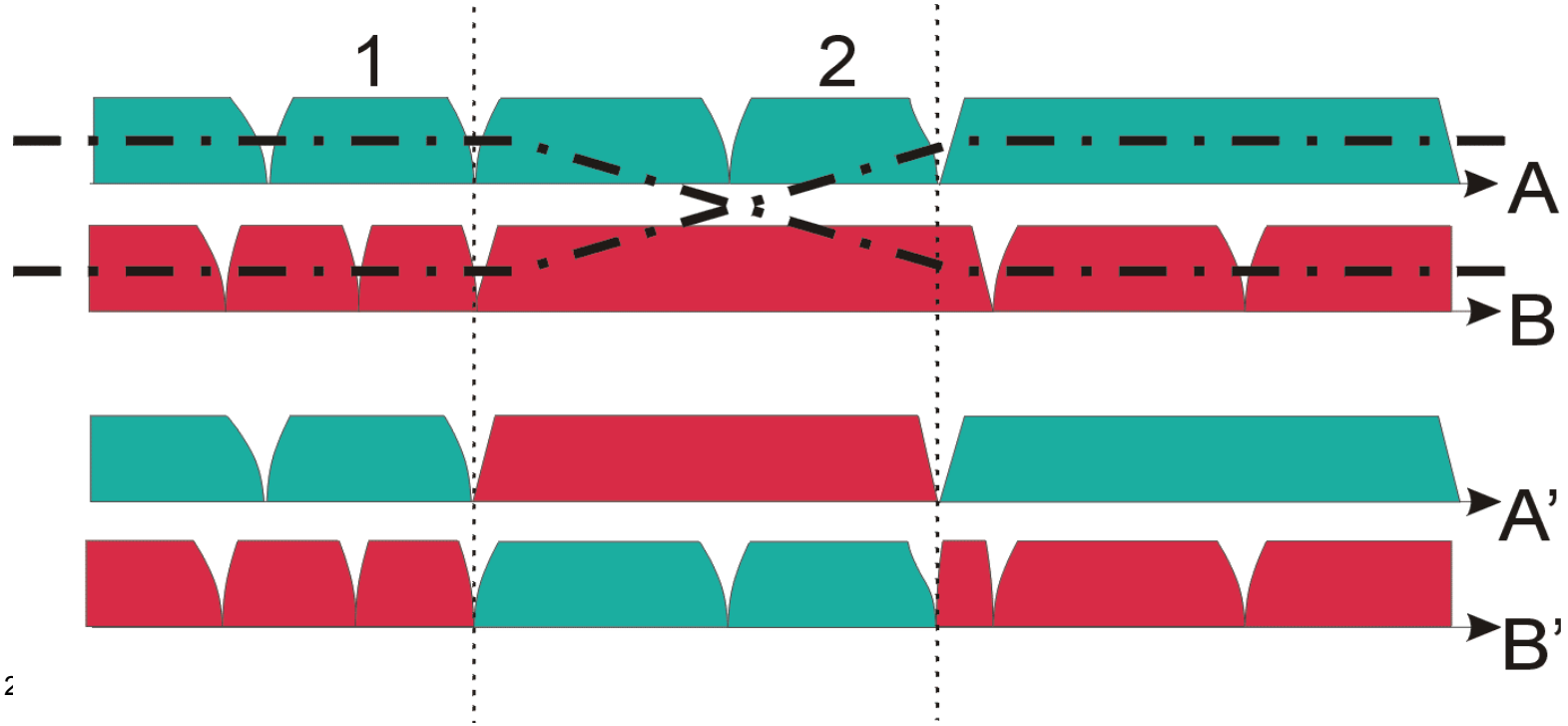
Operacja krzyżowania

Krzyżowanie chromosomów w celu uzyskania nowej populacji jest jedna z ciekawszych części tego algorytmu. Należy tutaj nadmienić raz jeszcze, że algorytm nie jest binarny i działa na liczbach rzeczywistych. Nigdzie nie następuje konwersja na liczby binarne. Krzyżowanie następuje na poziomie atrybutu poprzez ingerencje w wartości zapisane w tablicy `ciecia wartosc[max_gen]`.

Pierwotne próby krzyżowania na poziomie próbek wielowymiarowych poprzez wymianę całych dyskretnych atrybutów pomiędzy wielowymiarowymi osobnikami przynosiło gorsze efekty i zostało zarzucone.

Operacja krzyżowania c.d.





Operacja krzyżowania c.d.

Na podstawie eksperymentów został doświadczalnie dobrany algorytm który można streścić w następujących krokach;

- wybór osobników do krzyżowania (zgodnie z opisanym mechanizmem selekcji)
- wybór punktów krzyżowania w ilości domyślnie dwa ale użytkownik może zdecydować na początku inaczej.
- losowanie które przedziały (cięcia) z pierwszego chromosomu posłużą jako punkty krzyżowania
- złożenie informacji o cięciach naprzemiennie z wylosowanych osobników.

Tam gdzie przedziały zachodzą na siebie tworzone są nowe granice. Domyślnie przyjmuje się, że faktycznym miejscem krzyżowania jest prawa strona wylosowanego przedziału.

Operacja krzyżowania c.d.

- wybór osobników do krzyżowania

$$\begin{cases} \exists_{i \in [1, max_pop]} \left[i = select(P); P = \{Ch_1, \ldots, Ch_{max_pop}\} \right]; \\ \exists_{j \in [1, max_pop]} \left[j = select(P); P = \{Ch_1, \ldots, Ch_{max_pop}\} \right]; \\ i \neq j \end{cases}$$

- losowanie które cięcia posłużą jako punkty krzyżowania

$$\begin{cases} \exists_{c_{ik} \in C_i} \left[c_{ik} = random(C_i); C_i \subset Ch_i; k \in [1, n] \right]; \\ \exists_{c_{iz} \in C_i} \left[c_{iz} = random(C_i); C_i \subset Ch_i; k \in [1, n] \right]; \\ c_{ik} \neq c_{iz} \end{cases}$$

Zastąpienie populacji rodziców

Jako, że w tej wersji algorytmu LDGen założono iż jest on tzw. algorytmem bez pamięci więc populacja rodziców zostanie bezpowrotnie utracona. W trywialnej pętli następuje nadpisanie starej populacji właśnie uzyskaną populacją dzieci.

Ta klasyczna zasada została przyjęta celem uniknięcia zbyt wczesnej zbieżności algorytmu co bywa często skutkiem uboczny modyfikacji sztucznie faworyzujących najsilniejsze osobniki.

W kolejnych wersjach algorytmu można zbadać jaki wzrost efektywności uzyskany zostanie jeżeli nową populację utworzy się z najsilniejszych osobników z obu populacji, tak rodziców jak i dzieci.

Kontrola spełnienia warunku końca

Zakończenie działania algorytmu następuje gdy przynajmniej jeden z osobników osiągnie założoną sprawność funkcji przystosowania. Użytkownik może zmienić wartość, która domyślnie ustawiana jest na 10% ilości decyzji. Znaczy to iż poszukiwanie zostanie przerwane gdy w populacji znajdzie się przynajmniej jedna wielowymiarowa próbka tak zdyskretyzowana, że 90% równań decyzyjnych jest niesprzeczne. Oczywiście nie zawsze jest to możliwe i wtedy działa zabezpieczenie awaryjne przerywające algorytm po wygenerowaniu ilości populacji równej ilości próbek wymnożonej przez 100 `(ile_probek*100)`. Jeżeli proces został przerwany awaryjnie użytkownik jest o tym informowany, oczekiwany wynik jest osłabiany o 5% i wszystko zaczyna się od początku.

Metoda LDGen w ujęciu zwartym

Zastosowane w kodzie symbole oznaczają odpowiednio:

- P – populacja,
- Os – osobnik,
- Pp– populacja potomków,
- P – populacja,
- K – zbiór klas decyzyjnych,
- A – zbiór atrybutów,
- F – całkowite przystosowanie populacji,
- d – założona dokładność.

Metoda LDGen w ujęciu zwartym

```
Function LDGen
        (input:Z={A,K} zbiór przykładów, d założona dokładność
         output:Os zwycięski osobnik)
{
 define function populacja_startowa(input: A,K; output: P)
                {
                        for i=1 to max_pop do
                        {
                                set_priorytet_K(K)
                                sort(A,K|K)
                                P=P∪{O_i}
                        }
                }
 define function sila(input: P,K; output: ile niesprzecznych)
                {
                        sort(P,K|K)
                        test_ile_sprzecznych
                        sila=obl_proc_nsprz
                }
```

```
define function selekcja(input: P; output: Os)
        {
                F=suma(sila(P))
                sort(P|sila)
                losuj(r∈[0,F])
                znajdz(Os,r)
        }
define function krzyzowanie(input: P; output: Os1',Os2')
        {
                Os1=selekcja(P)
                Os2=selekcja(P)
                znajdz_pk(Os1)
                for Os1',Os2'
                 for i=1 to pk do
                 {
                        losuj(wariant)
                        case wariant
                         1:weź_frg(Os1)
                         2:weź_frg(Os2)
                         3:wchłoń(Os1,Os2)
                 }
                Pp= Pp∪{Os1',Os2'}
        }
```

```
43.  define function mutacja(input: P)
44.                 {
45.                         losuj(Os)
46.                         losuj(ilosc_ciec)
47.                         znajdz_i_podzil;
48.                 }
49.  define function nadpisanie(input: Pp; output:P)
50.                 {
51.                         for i=1 max_pop do
52.                                 P[i]=Pp[i]
53.                 }
54.  define function test_konca(input: P,K,d; output:{T,F})
55.                 {
56.                         if max(sila(P))≤d
57.                           then test_konca=T //rozwiązanie
58.                           else test_konca=F //dalsze poszukiwania
59.                         if ile_pop>stop_pop
60.                           then d=d-5%
61.                 }
```

```
main()
{
 populacja_startowa(A,K)
 while not test_konca(P,K,d) do
        {       ile_pop++
                krzyzowanie(P);
                mutacja(PP);
                nadpisanie(P,PP);
        }
}//main
}
```

Eksperymenty

Eksperymenty z LDGen przeprowadzono na pewnej grupie plików z danymi. Są to pliki różnego pochodzenia, ale większość była już wielokrotnie sprawdzana gdyż używano ich przy okazji badania wielu metod analizy danych. Wszystkie atrybuty w każdym z tych plików mają charakter numeryczny.

Lp.	Plik danych	Próbki	Atrybuty	Klasy decyzji	Dziedzina
0	Diagnosis	120	6	4	Medycyna
1	Bank	66	5	2	Ekonomia
2	Bupa	345	6	2	Medycyna
3	Buses	76	8	2	Komunikacja
4	Iris	250	4	3	Kwiaty
5	Meldb1	55	10	2	Mechanika
6	Pustak	216	10	2	Budownictwo

Metody zastosowane w porównaniu

- 1R - metoda 1R,
- AB - metoda oparta na algebrze Boole'a,
- AK - medianowa analiza klastrowa,
- Chi2 - metoda Chi2,
- CM - metoda ChiMerge,
- KO - metoda uwzględniająca koszty,
- MDLP - metoda minimalnej długości opisu,
- MKE - metoda minimalnej klasy entropii,
- OM - metoda odległości Mantaras,
- PRS - metoda przedziałów o równej szerokości,
- RC - metoda przedziałów o równej częstotliwości.

Przedstawienie wyników

Metoda	Quality	#Cuts	#Partitions	#Atoms	#Redund	Cluster-Ratio	Occupy-Ratio	Redund-Ratio
AB	0.96	9	96	17	2	0.89	0.18	0.22
MKE	0.94	13	320	37	9	0.76	0.12	0.69
AK	0.93	12	256	26	7	0.83	0.10	0.58
CM	0.88	15	420	40	10	0.74	0.10	0.67
Chi2	0.87	13	320	30	8	0.81	0.09	0.62
KO	0.86	12	256	23	6	0.85	0.09	0.50
RC	0.79	12	256	21	6	0.87	0.08	0.50
1R	0.77	12	225	13	5	0.92	0.06	0.42
PRS	0.75	14	400	23	9	0.85	0.06	0.64
OM	0.73	21	1344	56	13	0.63	0.04	0.62
MDLP	0.72	23	1960	64	10	0.58	0.03	0.43

Drugi etap eksperymentu

- AB - metoda oparta na algebrze Boole'a,
- AK - medianowa analiza klastrowa,
- MKE - metoda minimalnej klasy entropii,

- LDGen – metoda autorska

Charakterystyki dyskretyzacji MKE

Plik	Quality	#Cuts	#Partitions	#Atoms	#Redund	Cluster-Ratio	Occupy-Ratio	Redund-Ratio
Bank	1	10	243	34	10	0.49	0.14	1
Bupa	0.87	53	472392	285	29	0.17	0	0.55
Buses	1	8	256	20	8	0.75	0.08	1
Iris	0.94	12	256	36	4	0.77	0.14	0.33
Meldb1	1	30	1048576	33	28	0.41	0	0.93
Pustak	0.98	40	9765625	115	33	0.47	0	0.83

Charakterystyki dyskretyzacji AK

Plik	Quality	#Cuts	#Partitions	#Atoms	#Redund	Cluster-Ratio	Occupy-Ratio	Redund-Ratio
Bank	0.92	6	32	21	0	0.69	0.66	0
Bupa	0.97	34	70560	316	10	0	0	0.29
Buses	1	3	8	5	1	0.95	0.63	0.33
Iris	0.88	10	144	30	5	0.81	0.21	0.5
Meldb1	0.93	13	1080	20	2	0.65	0.02	0.15
Pustak	0.86	17	2400	76	1	0.65	0.03	0.06

Charakterystyki dyskretyzacji AB

Plik	Quality	#Cuts	#Partitions	#Atoms	#Redund	Cluster-Ratio	Occupy-Ratio	Redund-Ratio
Bank	1	3	8	7	0	0.91	0.88	0
Bupa	0.93	16	2048	285	0	0.17	0.14	0
Buses	1	2	4	4	0	0.96	1	0
Iris	0.93	4	9	7	0	0.96	0.78	0
Meldb1	0.96	7	96	14	0	0.76	0.15	0
Pustak	0.88	6	36	24	0	0.89	0.67	0

Charakterystyki dyskretyzacji LDGen

Plik	Quality	#Cuts	#Partitions	#Atoms	#Redund	Cluster-Ratio	Occupy-Ratio	Redund-Ratio
Bank	1	4	9	8	0	0.89	0.88	0
Bupa	0.96	14	1792	267	0	0.17	0.15	0
Buses	1	2	5	4	0	0.96	0.80	0
Iris	0.95	6	8	7	0	0.97	0.87	0
Meldb1	0.98	13	45	13	0	0.77	0.28	0
Pustak	1	7	28	22	0	0.90	0.78	0

Omówienie wyników

Z analizy wynika, że algorytmy *AK* i *AB* są znacznie lepsze od *MKE*. Zwłaszcza ilość cięć i redundancja jest w obu przypadkach znacznie mniejsza. Należy podkreślić, że metoda MKE należy do grupy dyskretyzacji lokalnych, co tłumaczy raczej słabą wydajność tejże dyskretyzacji, gdy bierzemy pod uwagę współczynniki takie jak *RedundRatio* (współczynnik redundancji) lub *OccupyRatio* (współczynnik zajętości).

Omówienie wyników c.d.

Analiza ujawnia również, że metoda *AB* tworzy lepsze rozwiązania w porównaniu, do tych które generowane są przez metodę *AK*. Wskazuje na to nie tylko mniejsza liczba punktów rozcinających przy jednocześnie wyższych wartościach stopnia dyskryminacji w dyskretyzacjach generowanych przez metodę *AB*, ale również istnienie zbytecznych (redundantnych) rozcięć w dyskretyzacjach tworzonych przez *AK*. Metoda *LDGen* wykazuje lepszą jakość niż *AB*, co jest czasami okupione nieznacznie większą ilością cięć. Obie metody wykazują brak redundancji. Metoda *AB* ma generalnie niższe niż *LDGen* współczynniki *RedundRatio* oraz *OccupyRatio*, co tym lepiej świadczy o algorytmie *LDGen*.

Epilog

Na koniec warto dodać, że droga dojścia do wyniku w metodzie LDGen jest za każdym razem inna. Wynika to z istoty podejścia, którego rdzeń jest oparty na algorytmie genetycznym. Eksperymentalnie stwierdzono, że średnio na 10 dyskretyzacji tego samego zbioru ok. 2 są znacząco słabsze, tyleż samo widocznie skuteczniejsze, a pozostałe utrzymują podobny poziom przedstawiony w tabeli z poprzedniego paragrafu. Gdyby zatem wybierać najlepszą z 10 dyskretyzacji, algorytm LDGen miałby jeszcze lepsze wyniki niż konkurenci ponieważ jest z natury algorytmem adaptacyjnym.

Elementy nowości w pracy

1. Zastosowanie algorytmów genetycznych do procesu dyskretyzacji danych wejściowych przetwarzanych przy użyciu zbiorów przybliżonych,
2. Opracowanie złożonych, autorskich operatorów krzyżowania i mutacji jakie zostały użyte w algorytmie,
3. Zastosowanie własnej metody selekcji będącej autorską modyfikacja metody "koła ruletki".

Lista publikacji doktoranta

- 1. J.Czerniak, H.Zarzycki, Application of rough sets in the presumptive diagnosis of urinary system diseases, Artifical Inteligence and Security in Computing Systems, ACS'2002 9th International Conference, Kluwer Academic Publishers, str. 41-51
- 2. J.Czerniak, Wizualizacja wielowymiarowości za pomocą 'Rury próbek' - ogólna koncepcja zagadnienia, Materiały VII Sesji Naukowej Informatyki, Politechnika Szczecińska, Wydział Informatyki, Szczecin, Grudzień 2002, tom II, str. 375-381
- 3. J.Czerniak, H.Zarzycki, Application of the LDGen Genetic Algorithm in a Discretization of Numerical Attributes, ACS'2003 10th International Multi-Conference on Advanced Computer Systems, Międzyzdroje, Październik 2003
- 4. J.Czerniak, Metoda wielowymiarowego krzyżowania zastosowana w algorytmie LDGen, Materiały VIII Sesji Naukowej Informatyki, Politechnika Szczecińska, Wydział Informatyki, Szczecin, Grudzień 2003, str. 403-406
- 5. J.Czerniak, Metoda selekcji osobników zastosowana w algorytmie LDGen, Materiały VIII Sesji Naukowej Informatyki, Politechnika Szczecińska, Wydział Informatyki, Szczecin, Grudzień 2003, str. 397-401
- 6. J.Czerniak, M.Dramski, Zarządzanie kodem w korporacji produkującej oprogramowanie, Materiały VIII Sesji Naukowej Informatyki, Politechnika Szczecińska, Wydział Informatyki, Szczecin, Grudzień 2003, str. 407-412
- 7. J.Czerniak, M.Dramski, Przetwarzanie danych w gabinecie lekarskim, Materiały VIII Sesji Naukowej Informatyki, Politechnika Szczecińska, Wydział Informatyki, Szczecin, Grudzień 2003, str. 413-418
- 8. J.Czerniak, W.Śpiewak, Large-scale Software Development in the Multinational Software Corporation, Pismo Naukowe PTI Pro Dialog 18 Computer Programming and Applications,No. 18,2004, str. 61-70

Lista publikacji doktoranta c.d.

- 8a J.Czerniak, W.Spiewak, Large-Scale Software Development in a Multinational Software Corporation, UPGRADE The European Journal for the Informatics Professional, <http://www.upgrade-cepis.org/>,Vol. V, No. 4, August 2004, pp.63-70 (przedruk z ProDialog na mocy porozumienia wydawców)
- 9. J.Czerniak, Porównanie metody LDGen z wiodącymi metodami dyskretyzacji wstępnej stosowanymi w kontekście teorii zbiorów przybliżonych, Roczniki Informatyki Stosowanej, tom 6, Politechnika Szczecińska, Wydział Informatyki, Szczecin, Grudzień 2004, (w trakcie recenzowania)
- 10. J.Czerniak, Klasyfikacja metod dyskretyzacji stosowanych w indukcji wiedzy metodami zbiorów przybliżonych, Roczniki Informatyki Stosowanej, tom 6, Politechnika Szczecińska, Wydział Informatyki, Szczecin, 2004,str.49-58
- 11. J.Czerniak, W.Spiewak, The architecture of applications and services .NET, (ed.) Guy Pujolle, IFIP Open Conference on Metropolitan Area Networks Architecture, protocols, control and management, Ho Chi Minh city, Vietnam, April 11-13, 2005, str.28-34
- Publikacje popularnonukowe:
- 12. J.Czerniak, Algorytmy genetyczne w praktyce, Software 2.0 Extra, Numer 10, Sztuczna Inteligencja, 2004, str. 60-65 (ukazał się po polsku w nakładzie 6000egz, oraz po hiszpańsku, niemiecku i francusku)
- Publikacje dydaktyczne:
- 13. J.Czerniak, T.Karasiewicz, M.Kociszewski, skrypt pt. "Podstawy programowania strukturalnego", (*zgłoszony do planu wydawniczego Wydawnictwa Uniwersytetu Kazimierza Wielkiego*)

Literatura dotycząca TZP

1. J.Mulawka, Systemy ekspertowe, Wydawnictwa Naukowo-Techniczne, 1996
2. J. Chromiec, E. Strzemieczna, Sztuczna inteligencja. Metody konstrukcji i analizy systemów eksperckich, PLJ, 1996
3. L.Rutkowski, Metody i techniki sztucznej inteligencji, PWN, 2006
4. D.Hand, H.Mannila, P.Smyth, Eksploracja danych, WNT, 2003
5. A. Niederliński "Regułowo-Modelowe Systemy Ekspertowe", PKJS marzec 2006,
6. M.Owoc, Elementy systemów ekspertowych. Cz.I: Sztuczna inteligencja i systemy ekspertowe, Wydawnictwo Akademii Ekonomicznej we Wrocławiu, 2006
7. Grzech A. (red.): Inżynieria wiedzy i systemy ekspertowe, Oficyna Wydawnicza Politechniki Wrocławskiej, 2006
8. B.Stefanowicz, Systemy eksperckie Przewodnik (Wydanie III rozszerzone), WSISiZ, 2003
9. J.Jagielski, Inżynieria wiedzy, Oficyna Wydaw. Uniwersytetu Zielonogórskiego, 2005
10. J.Jagielski, Inżynieria wiedzy w systemach ekspertowych, Lubuskie Towarzystwo Naukowe, 2001

Dziękuję za uwagę

dr inż. J.Czerniak

jczerniak@ukw.edu.pl