

ROMAN SŁOWIŃSKI

## Techniczne systemy informatyki jako narzędzia odkrywania wiedzy i symulacji życia

Informatyka jest nauką badającą procesy algorytmiczne, które opisują i przetwarzają informację (Denning i in., 1989; Węglarz, 1998). Dokonuje się to w systemach technicznych, biologicznych, nanotechnicznych i kwantowych (Węgrzyn, 2003). Techniczne systemy informatyki stanowią dziś dominujący dział informatyki – ich celem jest realizacja procesów obliczeniowych i procesów gromadzenia, przetwarzania i przekazywania danych. Od przeszło pięćdziesięciu lat rozwój technicznych systemów informatyki postępuje w tempie rozwoju technologii. Gordon Moore, jeden z założycieli firmy Intel, sformułował w 1965 r. prawo, według którego liczba tranzystorów, jakie można umieścić w pojedynczym układzie scalonym – a co za tym idzie, moc obliczeniowa komputerów – podwaja się co 18 miesięcy. Ewolucja układów scalonych przeszła od jednego tranzystora na płycie do 100 milionów w ciągu zaledwie 40 lat. Kolejnym punktem zwrotnym tej ewolucji może być przeniesienie obliczeń na poziom atomowy, a nawet subatomowy. Szybkość komunikacji – przewodowej i bezprzewodowej – podwaja się co 12 miesięcy.

Postęp w zdolności gromadzenia i przesyłania informacji był jednak niewspółmiernie duży w porównaniu z postępem w metodyce przetwarzania zgromadzonych danych, co doprowadziło do przerostu „biernej” funkcji technicznych systemów informatycznych, skoncentrowanej na archiwizacji danych i organizowaniu dostępu do baz danych. Nie sprawdziła się tu „zasada”, że ilość przechodzi w jakość. Przepaść, jaka się ciągle tworzy między generowaniem danych a ich zrozumieniem, sprawia, że następuje inflacja informacji, a rośnie koszt *wiedzy*, która jest wynikiem jej uogólnienia i przetworzenia do postaci użytecznej. *Odkrywanie wiedzy z danych* stało się zatem wielkim wyzwaniem dla informatyki – takie podejście do przetwarzania danych wymaga rozwoju „aktywnej” funkcji technicznych systemów informatyki, nazywanej także inteligencją maszynową lub sztuczną inteligencją. Najkrócej mówiąc, aktywność (lub inteligencja) technicznych systemów informatyki przejawia się w zdolności uczenia się i automatycznej adaptacji maszyn do potrzeb (preferencji) użytkownika.

Związek informatyki i sztucznej inteligencji podkreśla definicja inteligencji podana przez Norberta Wienera (1961): „inteligencja jest procesem pozyskiwania i przetwarzania informacji dla osiągnięcia wyznaczonych celów”.

W niniejszym artykule pragniemy zwrócić uwagę na rolę technicznych systemów informatyki w odkrywaniu wiedzy z danych. Dlatego też nazywamy te systemy narzędziami odkrywania wiedzy. Dodajmy, że nie chodzi tu o wiedzę typu encyklopedycznego, lecz jedynie o zespół wniosków, zwanych wzorcami, wydrukowanych z danych. Odkrywanie wiedzy jest ukierunkowane na wspomaganie decyzji, rozumiane z jednej strony – jako wyjaśnianie sytuacji decyzyjnej opisanej przez dane, a z drugiej – jako zalecanie akcji i zachowań spójnych z systemem preferencji decydenta. Zależnie od kontekstu wspomaganie decyzji, pod pojęciem decydenta występuje albo człowiek, albo grupa ludzi, albo sztuczny agent. W części końcowej artykułu zwrócimy uwagę na jeszcze inny aspekt aktywnych systemów informatycznych – odkrywanie wiedzy połączone z działaniem sztucznego agenta w środowisku, z którego dane pochodzą. Domknięcie pętli „środowisko-informacja-agent-czynność-środowisko” doprowadziło do powstania dziedziny *sztucznego życia*, która korzysta z doświadczeń informatyki (odkrywanie wiedzy), optymalizacji (algorytmy ewolucyjne) i robotyki (konstrukcja agenta). W tym sensie techniczne systemy informatyki są również narzędziem symulacji życia.

Sposób ukazania w tym artykule zastosowań technicznych systemów informatyki do odkrywania wiedzy i symulacji życia jest z pewnością naznaczony osobistymi doświadczeniami badawczymi autora i jego zespołu, wobec czego artykuł ten nie pretenduje do roli przeglądu stanu wiedzy w całej bogatej dziedzinie.

### Odkrywanie wiedzy z danych

Odkrywanie wiedzy (ang. *knowledge discovery*) z danych jest procesem identyfikowania prawdziwych, nietrywialnych, potencjalnie użytecznych i bezpośrednio zrozumiałych wzorców w danych. Przez wzorzec rozumie się regułę, trend, zjawisko, relację przyczynowo-skutkową, prawidłowość, anomalie, hipotezę itp. Wzorce są użyteczne dla wyjaśniania sytuacji opisanych przez dane oraz do predykcji przyszłych sytuacji.

Odkrywanie wiedzy może być:

- nieukierunkowane („powiedz mi coś interesującego o moich danych”),
- ukierunkowane („scharakteryzuj klientów, którzy odpowiedzieli na ofertę promocyjną”),
- testowaniem lub uściśleniem hipotez („czy to prawda, że jest związek między sposobem odżywiania a zapadalnością na chorobę X?”).

Metodologia odkrywania wiedzy jest uwarunkowana takimi cechami danych, jak:

- duże rozmiary bazy danych (hurtownie danych),
- dynamiczna zmienność danych,

- niedokładność i niepewność danych,
- niespójność danych,
- z jednej strony częściowy brak danych, a z drugiej redundancja.

Dodajmy, że na te cechy danych nakłada się jeszcze sprawa percepcji danych przez człowieka. Rozumowanie w kategoriach liczb jest dla człowieka o wiele mniej naturalne niż w kategoriach pojęć, porządku, symboli. Pojęcia agregują dane numeryczne pochodzące z pomiaru, czyli dokonują granulacji informacji i sprawiają, że reprezentacja wiedzy wchodzi na wyższy poziom abstrakcji, bliższy rozumieniu człowieka.

Obecnie dominują dwa sposoby reprezentacji wiedzy (wzorców):

- reguły decyzyjne,
- drzewa decyzyjne.

Przykłady tych reprezentacji podamy w następnym paragrafie.

Metodyka odkrywania wiedzy z danych jest uwarunkowana ich charakterem. Ze względu na ten aspekt wyróżnimy następujące dane i podejścia metodyczne:

- ilościowe i jakościowe – indukcyjna eksploracja danych (ang. *data mining*),
- tekstowe – eksploracja tekstów lub zasobów Internetu (ang. *text mining*, *Web mining*),
- obrazowe i dźwiękowe – rozpoznawanie wzorców (ang. *pattern recognition*).

W dalszym ciągu scharakteryzujemy te trzy podejścia metodyczne. Z uwagi na charakter artykułu i rozległość tematu będziemy przedkładać zrozumiałość tekstu przez jak najszersze grono odbiorców nad formalną ścisłość i szczegóły, które cechuje cytowana literatura.

### Odkrywanie wiedzy z danych ilościowych i jakościowych

Dane ilościowe i jakościowe opisują obiekty, tj. sytuacje, transakcje, osoby, akcje, itp., w języku przyjętych atrybutów, tj. zmiennych, cech, wskaźników, symptomów, itp. Innymi słowy, dla każdej pary obiekt-atrybut znana jest pewna wartość (niekoniecznie deterministyczna) ilościowa lub jakościowa. Atrybut przyjmujący wartości ilościowe nazywa się numerycznym, a atrybut przyjmujący wartości jakościowe – nominalnym.

Ponadto, jeśli dziedzina wartości danego atrybutu jest uporządkowana według preferencji (od najgorszej do najlepszej), to taki atrybut nazywa się kryterium. Dziedzina wartości kryterium tworzy zatem pewną skalę: porządkową (jeśli odległość na tej skali nie ma charakteru ilościowego, a jedynie porządkowy, np. oceny szkolne), przedziałową (jeśli zero na tej skali nie ma charakteru bezwzględnie, a porównywanie przedziałów wartości ma znaczenie ilościowe, np. temperatura w skali Celsjusza) lub ilorazową (jeśli zero na tej skali ma znaczenie bezwzględne i odległość względem niego ma charakter ilościowy, np. ciężar).

Odkrywanie wiedzy z danych odbywa się na drodze indukcji, będącej szczególnym typem wnioskowania w tzw. uczeniu maszynowym (ang. *machine learning*). Jest to proces tworzenia wzorców (hipotez, uogólnień) prawdziwych w świecie analizowanych danych. Te wzorce (w formie reguł lub drzew decyzyjnych) z jednej strony opisują istotne związki występujące w danych, a z drugiej mogą być wykorzystane do predykcji związków w nowych danych. Należy jednak zaznaczyć za Karlem Popperem (1977), że wiedza wyindukowana z danych jest jedynie domyślna, tzn. nie można udowodnić prawdziwości uogólnień specyficznych obserwacji lub analogii do znanych faktów, choć można je obalić. Tego ograniczenia nie ma drugi typ wnioskowania w uczeniu maszynowym, zwany dedukcją.

Dla przykładu przeanalizujmy zdanie: „Wszyscy studenci kierunku Informatyka i specjalności ISWD (Inteligentne Systemy Wspomagania Decyzji) są inteligentni”. Dedukcyjnym wnioskiem wyciągniętym z tego zdania może być stwierdzenie, że: „Student JB ze specjalności ISWD jest inteligentny”. Przykładem indukcyjnego wniosku z pierwotnego zdania może być stwierdzenie: „Wszyscy studenci na kierunku Informatyka są inteligentni”. W tym wypadku, nawet jeśli pierwotne zdanie jest prawdziwe, wniosek indukcyjny może okazać się nieprawdziwy (gdy znajdzie się student informatyki, który inteligentny nie jest). Zatem w przeciwieństwie do wnioskowania dedukcyjnego, prawdziwa przesłanka we wnioskowaniu indukcyjnym nie gwarantuje prawdziwości wniosku.

Pomimo powyższego ograniczenia indukcyjna eksploracja danych oddaje cenne usługi w odkrywaniu wzorców z danych dla celów opisu i predykcji. Dla zilustrowania tych korzyści rozpatrzmy niewielki zbiór danych opisujący zachowania ludzi udających się na kolację do restauracji (jest to modyfikacja przykładu podanego przez Russela i Norviga (1995)) (por. także Słowiński, 2003). W każdej z 12 sytuacji przedstawionych w tabeli 1, klienci decydowali się ‘czekać’ albo ‘nie czekać’ na stolik. Chcielibyśmy odkryć z tych danych związki między opisem sytuacji w języku 10 atrybutów warunkowych a decyzją o czekaniu. Naprzeciw tego typu dążeniu wychodzi teoria zbiorów przybliżonych (ang. *rough set theory*) zaproponowana przez Zdzisława Pawlaka (1991) (por. także Słowiński, 1992; Słowiński, 1995; Pawlak i Słowiński, 2004). Przykład „restauracyjny” ukaze nam nieformalnie niektóre aspekty tej teorii. Wszystkie atrybuty są nominalne i założmy początkowo, że żadne z nich nie jest kryterium. Ich znaczenie i dziedziny wartości są następujące:

1. *Alt* – czy istnieje alternatywna restauracja w okolicy? {TAK, NIE},
2. *Bar* – czy jest tam bar, w którym można wygodnie poczekać? {NIE, TAK},
3. *Sob* – czy to jest sobotni wieczór? {NIE, TAK},
4. *Głód* – czy są głodni? {NIE, TAK},
5. *Zaj* – zajętość restauracji na pierwszy rzut oka {NIS, ŚR, WYS},
6. *Cena* – średnia cena dań {NIS, ŚR, WYS},

7. *Pada* – czy pada deszcz? {NIE, TAK},

8. *Rez* – czy klienci dokonali wcześniej rezerwacji? {NIE, TAK},

9. *Typ* – rodzaj restauracji {FR, WŁ, AZJ, McD},

10. *Czas* – czas oczekiwania przewidywany przez obsługę {KR, NDŁ, DŁ, BDŁ}.

Atrybut decyzyjny, który określa wynik każdej sytuacji jest dwuwartościowy: {Nie\_czekać, Czekać}.

Zauważmy najpierw, że zbiór danych „restauracyjnych” jest spójny, tzn. nie występują w nim sytuacje, które miałyby ten sam opis (takie same wartości) w języku 10 atrybutów warunkowych, a wartość atrybutu decyzyjnego byłaby dla nich różna.

Tabela 1. Zbiór danych „restauracyjnych”

Syt.	Atrybuty warunkowe										Atrybut decyzyjny
	<i>Alt</i>	<i>Bar</i>	<i>Sob</i>	<i>Głód</i>	<i>Zaj</i>	<i>Cena</i>	<i>Pada</i>	<i>Rez</i>	<i>Typ</i>	<i>Czas</i>	<i>Decyzja</i>
1	TAK	NIE	NIE	TAK	ŚR	WYS	NIE	TAK	FR	KR	Czekać
2	TAK	NIE	NIE	TAK	WYS	NIS	NIE	NIE	AZJ	DŁ	Nie_czekać
3	NIE	TAK	NIE	NIE	ŚR	NIS	NIE	NIE	McD	KR	Czekać
4	TAK	NIE	TAK	TAK	WYS	NIS	NIE	NIE	AZJ	NDŁ	Czekać
5	TAK	NIE	TAK	NIE	WYS	WYS	NIE	TAK	FR	BDŁ	Nie_czekać
6	NIE	TAK	NIE	TAK	ŚR	ŚR	TAK	TAK	WŁ	KR	Czekać
7	NIE	TAK	NIE	NIE	NIS	NIS	TAK	NIE	McD	KR	Nie_czekać
8	NIE	NIE	NIE	TAK	ŚR	ŚR	TAK	TAK	AZJ	KR	Czekać
9	NIE	TAK	TAK	NIE	WYS	NIS	TAK	NIE	McD	BDŁ	Nie_czekać
10	TAK	TAK	TAK	TAK	WYS	WYS	NIE	TAK	WŁ	NDŁ	Nie_czekać
11	NIE	NIE	NIE	NIE	NIS	NIS	NIE	NIE	AZJ	KR	Nie_czekać
12	TAK	TAK	TAK	TAK	WYS	NIS	NIE	NIE	McD	DŁ	Czekać

Powyższe 12 sytuacji można traktować jak przykłady uczące (dane) dla algorytmu indukcji, który odkrywa związki między decyzją a opisem sytuacji. Związki te są reprezentowane przez wyrażenia logiczne: reguły decyzyjne lub drzewo decyzyjne. Reprezentacje te powinny być minimalne w tym sensie, że liczba związków pokrywających wszystkie przykłady uczące ma być minimalna i liczba warunków elementarnych używanych w każdym związku ma być także minimalna.

Siedem poniższych reguł decyzyjnych spełnia te warunki. Spełnia je także drzewo decyzyjne przedstawione na rycinie 1. W nawiasach przy regułach i przy liściach drzewa podano numery sytuacji pokrywanych przez daną regułę lub gałąź drzewa. Reguły decyzyjne i drzewo decyzyjne reprezentują wiedzę odkrytą z danych.

Reguła 1: Jeżeli $Zaj=NIS$ , to $Decyzja=Nie\_czekać$	(7, 11)
Reguła 2: Jeżeli $Zaj=WYS$ i $Głód=NIE$ , to $Decyzja=Nie\_czekać$	(5, 9)
Reguła 3: Jeżeli $Zaj=WYS$ i $Typ=WŁ$ , to $Decyzja=Nie\_czekać$	(10)
Reguła 4: Jeżeli $Zaj=WYS$ i $Sob=NIE$ , to $Decyzja=Nie\_czekać$	(2)
Reguła 5: Jeżeli $Zaj=ŚR$ , to $Decyzja=Czekać$	(1, 3, 6, 8)
Reguła 6: Jeżeli $Sob=TAK$ i $Typ=AZJ$ , to $Decyzja=Czekać$	(4)
Reguła 7: Jeżeli $Głód=TAK$ i $Typ=McD$ , to $Decyzja=Czekać$	(12)

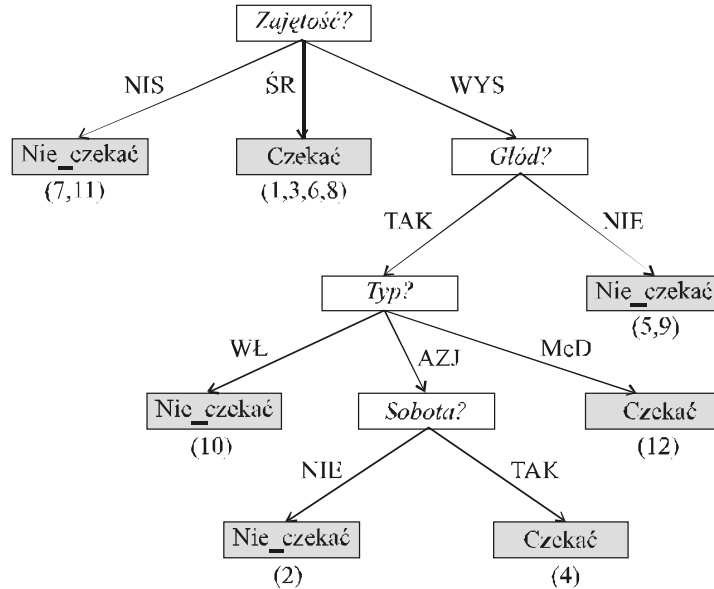
Okazało się, że reguły i drzewo decyzyjne można wyrazić w języku tylko 4 atrybutów za pomocą nie więcej niż 12 warunków elementarnych (tabela 1 zawiera ich aż 120). Nie jest to jedyna możliwa reprezentacja związków występujących w danych „restauracyjnych”. Opracowany przez nas system eksploracji danych o nazwie ROSE (dostępny pod adresem <http://www-idss.cs.put.poznan.pl/rose>), oparty na teorii zbiorów przybliżonych, stwierdza, że istnieje 19 podzbiorów atrybutów warunkowych (cztero- lub trój-elementowych reduktów), którymi można wyrazić wiedzę z tą samą dokładnością. Są one następujące:

- |                                     |                                      |
|-------------------------------------|--------------------------------------|
| 1. { <i>Sob, Pada, Typ</i> }        | 11. { <i>Alt, Sob, Zaj, Rez</i> }    |
| 2. { <i>Pada, Typ, Czas</i> }       | 12. { <i>Sob, Głód, Zaj, Rez</i> }   |
| 3. { <i>Bar, Zaj, Czas</i> }        | 13. { <i>Sob, Zaj, Pada, Rez</i> }   |
| 4. { <i>Zaj, Typ, Czas</i> }        | 14. { <i>Sob, Zaj, Rez, Czas</i> }   |
| 5. { <i>Bar, Sob, Cena, Pada</i> }  | 15. { <i>Alt, Sob, Zaj, Typ</i> }    |
| 6. { <i>Bar, Sob, Pada, Rez</i> }   | 16. { <i>Sob, Głód, Zaj, Typ</i> }   |
| 7. { <i>Alt, Sob, Zaj, Cena</i> }   | 17. { <i>Bar, Głód, Pada, Czas</i> } |
| 8. { <i>Sob, Głód, Zaj, Cena</i> }  | 18. { <i>Bar, Cena, Pada, Czas</i> } |
| 9. { <i>Sob, Zaj, Cena, Pada</i> }  | 19. { <i>Bar, Pada, Rez, Czas</i> }  |
| 10. { <i>Sob, Zaj, Cena, Czas</i> } |                                      |

Zauważmy, że podzbiór 4 atrybutów użytych w powyższych regułach i drzewie decyzyjnym odpowiada reduktowi nr 16.

Przykład „restauracyjny” został dobrany, tak by zwrócić uwagę na pewną sprzeczność, która pojawia się w reprezentacji wiedzy odkrytej klasycznymi metodami indukcji, gdy pominiemy ważną informację semantyczną zawartą w danych.

Otóż reguła 1. mówi, że klienci nie decydują się czekać na stolik, gdy w restauracji jest mało ludzi, a reguła 5. mówi, że klienci decydują się czekać, gdy w restauracji jest więcej ludzi, niezależnie od wartości pozostałych atrybutów. Należy stwierdzić, że reguła 1. jest niespójna z regułą 5., gdyż decyzja o odejściu z mniej zajętego lokalu jest nieracjonalna.



Ryc. 1. Drzewo decyzyjne wyindukowane ze zbioru danych „restauracyjnych”

Zauważona sprzeczność wynika z pominięcia w analizie istotnej informacji semantycznej mówiącej o korelacji preferencji w dziedzinach pewnych atrybutów warunkowych z dziedziną atrybutu decyzyjnego. Zgodźmy się najpierw z tym, że jeśli wybieramy się do restauracji, to decyzja o czekaniu na stolik oznacza, że prędzej czy później zjemy kolację i zrealizujemy swój zamiar, natomiast decyzja o nieczekaniu czyni realizację tego zamiaru niepewną. W świetle tego zamiaru decyzja ‘czekać’ jest zatem lepsza od ‘nie czekać’. Ponadto, w świetle tego samego zamiaru, dziedziną wartości atrybutu *Zaj* (zajętość) jest uporządkowana według preferencji, gdyż niska zajętość jest lepsza od średniej, a średnia jest lepsza od wysokiej. Atrybut *Zaj* jest zatem kryterium.

Tak jak *Zaj*(-), kryteriami są również atrybuty *Alt*(-), *Bar*(-), *Głód*(-), *Cena*(-), *Pada*(-), *Rez*(-) i *Czas*(-). Strzałka obok nazwy kryterium oznacza kierunek preferencji w dziedzinie wartości: „→” oznacza kryterium typu ‘zysk’, a „-” kryterium typu ‘koszt’. Weźmy na przykład kryterium *Pada*(-) – jeśli na zewnątrz pada deszcz, to bardziej racjonalne jest czekanie niż nieczekanie na stolik, czyli między dziedzinami kryterium *Pada* i atrybutu decyzyjnego *Decyzja* istnieje pozytywna korelacja semantyczna. Pozytywna korelacja semantyczna między kryterium typu zysk i decyzją oznacza, że jeśli ocena na tym kryterium staje się wyższa, to decyzja nie powinna się pogorszyć. Analogicznie, negatywna korelacja semantyczna między kryterium typu koszt i decyzją oznacza, że jeśli ocena na tym kryterium staje się niższa, to decyzja nie powinna się pogorszyć.

Atrybuty *Sob* i *Typ* nie mają takiej korelacji z atrybutem decyzyjnym, dlatego pozostają zwykłymi atrybutami.

Powyższe informacje nt. uporządkowania dziedzin niektórych atrybutów według preferencji oraz nt. semantycznej korelacji między tymi atrybutami (kryteriami) a atrybutem decyzyjnym stanowią tzw. wiedzę dziedzinową, której nie należy pomijać przy odkrywaniu wiedzy z danych ze względu na ryzyko otrzymania fałszywych wzorców (por. Słowiński i in., 2002).

Po uwzględnieniu wiedzy dziedzinowej, dotyczącej naszego przykładu, można zauważyć, że zbiór danych „restauracyjnych” nie jest spójny, gdyż między sytuacjami 3 i 7 zachodzi sprzeczność (patrz tabela 2).

Tabela 2. Niespójność w zbiorze danych „restauracyjnych”

Syt.	Kryteria i atrybuty warunkowe										Atrybut decyzyjny
	<i>Alt</i>	<i>Bar</i>	<i>Sob</i>	<i>Głód</i>	<i>Zaj</i>	<i>Cena</i>	<i>Pada</i>	<i>Rez</i>	<i>Typ</i>	<i>Czas</i>	<i>Decyzja</i>
	→	→		→	←	←	→	→		←	→
3	NIE	TAK	NIE	NIE	ŚR	NIS	NIE	NIE	McD	KR	Czekać
7	NIE	TAK	NIE	NIE	NIS	NIS	TAK	NIE	McD	KR	Nie_czekać

Rzeczywiście, ocena sytuacji 3 różni się od oceny sytuacji 7 jedynie na dwóch kryteriach: *Zaj*, na którym sytuacja 7 jest lepsza od 3, i *Pada*, na którym sytuacja 7 jest także lepsza od 3. Tymczasem w sytuacji 7 podjęto decyzję ‘nie czekać’, czyli, w świetle zamiaru zjedzenia kolacji, decyzję gorszą niż ‘czekać’ podjętą w sytuacji 3.

Innymi słowy, sytuacja 7 dominuje sytuację 3 w sensie oceny za pomocą kryteriów i atrybutów, a mimo to decyzja w sytuacji 7 jest gorsza niż w 3. O takich sytuacjach (objektach w bazie danych) mówimy, że są niespójne z zasadą dominacji (zwanej też zasadą Pareto) będącą jedyną obiektywną zasadą obowiązującą przy wielokryterialnym porównywaniu obiektów.

Ważne jest, by taką niespójność wykryć w bazie danych, gdyż wpływa ona na zaufanie do odkrywanej wiedzy. Greco, Matarazzo i Słowiński (1999, 2001) podali metodykę indukowania reguł decyzyjnych o zadanym stopniu spójności, opartą na rozszerzeniu teorii zbiorów przybliżonych na obecność kryteriów w problemach decyzyjnych, w szczególności w problemach klasyfikacji (por. systemy *4eMka* i *JAMM* pod adresem <http://www-idss.cs.put.poznan.pl>). Metodyka ta nosi nazwę podejścia zbiorów przybliżonych opartego na dominacji (ang. *Dominance-based Rough Set Approach – DRSA*) Zastosowanie metodyki DRSA do indukcji reguł decyzyjnych z danych „restauracyjnych” dało następujący wynik:



- Reguła 1': Jeżeli  $Zaj = WYS$  i  $Typ = WŁ$ , to  $Decyzja = Nie\_czekać$  (10) [Zaufanie=1]  
 Reguła 2': Jeżeli  $Głód = NIE$ , to  $Decyzja = Nie\_czekać$  (3, 5, 7, 9, 11) [Zaufanie=0,8]  
 Reguła 3': Jeżeli  $Zaj = WYS$  i  $Sob = NIE$ , to  $Decyzja = Nie\_czekać$  (2) [Zaufanie=1]  
 Reguła 4': Jeżeli  $Zaj \leq \acute{S}R$  i  $Głód = TAK$ , to  $Decyzja = Czekać$  (1, 6, 8) [Zaufanie=1]  
 Reguła 5': Jeżeli  $Sob = TAK$  i  $Typ = AZJ$ , to  $Decyzja = Czekać$  (4) [Zaufanie=1]  
 Reguła 6': Jeżeli  $Głód = TAK$  i  $Typ = McD$ , to  $Decyzja = Czekać$  (12) [Zaufanie=1]

Powyższa reprezentacja wiedzy jest bardziej zwarta niż poprzednia, gdyż liczba reguł jest o jeden mniejsza. Składnia tych reguł jest również ogólniejsza niż poprzednio, gdyż dopuszcza relacje typu „ $\leq$ ”, „ $\geq$ ” w warunkach elementarnych i jest ich w sumie 11. Ponadto, alternatywnych reprezentacji minimalnych jest obecnie 11, gdyż stwierdzono tyleż reduktów złożonych z 3, 4 lub 5 kryteriów i atrybutów. Należy jednak dodać, że jakość wyrażenia decyzji za pomocą tych kryteriów i atrybutów jest obecnie niższa niż 100%, gdyż, po uwzględnieniu wiedzy dziedzinowej, w zbiorze danych stwierdzono obecność dwóch sytuacji niespójnych z zasadą dominacji (3 i 7); jakość ta wynosi teraz 83%. Redukty zapewniające tę jakość są następujące:

- |                                    |  |
|------------------------------------|--|
| 1. { <i>Głód, Typ, Czas</i> }      | 7. { <i>Bar, Głód, Cena, Czas</i> }      |
| 2. { <i>Sob, Głód, Pada, Typ</i> } | 8. { <i>Sob, Głód, Rez, Typ</i> }        |
| 3. { <i>Sob, Głód, Zaj, Typ</i> }  | 9. { <i>Bar, Sob, Cena, Rez, Czas</i> }  |
| 4. { <i>Alt, Sob, Głód, Typ</i> }  | 10. { <i>Bar, Sob, Głód, Zaj, Cena</i> } |
| 5. { <i>Sob, Rez, Typ, Czas</i> }  | 11. { <i>Bar, Sob, Głód, Cena, Rez</i> } |
| 6. { <i>Sob, Pada, Typ, Czas</i> } |  |

Zauważmy, że w powyższych regułach decyzyjnych zostały użyte kryteria i atrybuty z reduktu nr 3.

Do informacji o pokryciu sytuacji przez poszczególne reguły dołączona jest informacja o stopniu zaufania, jaki można do nich żywić, biorąc pod uwagę niespójności występujące w pokrywanych sytuacjach. Reguła 2. ma zaufanie rzędu 0,8, gdyż pokrywa pięć sytuacji z decyzją ‘nie czekać’ i jedną sytuację z decyzją przeciwną (sytuacja 3.).

Dodajmy, że wiedzę reprezentowaną przez reguły decyzyjne 1'-6' można także przedstawić w postaci monotonicznego drzewa decyzyjnego, tak jak to zaproponowano w (Giove i in., 2002).

Z punktu widzenia wspomaganie decyzji, przykłady sytuacji decyzyjnych podane w tabeli 1 można potraktować jako informacje preferencyjne dostarczone przez decydenta (w tym przypadku klienta) w celu skonstruowania modelu jego preferencji. Faktycznie, w wyniku zastosowania podejścia DRSA do tej informacji preferencyjnej uzyskujemy *model preferencji decydenta* w postaci zbioru reguł decyzyjnych opartych na dominacji.

Model ten może być wykorzystany do wyjaśniania preferencji i (lub) predykcji nowych decyzji. Reguły decyzyjne mają specjalną składnię, do której wchodzi częściowy profil ocen na podzbiorze kryteriów i relacja dominacji między tym profilem a oceną obiektu. Jest to bardzo prosta i naturalna w interpretacji reprezentacja preferencji. Jej konkurentami w klasycznej teorii decyzji jest funkcja użyteczności i system relacyjny (np. relacja przewyższania lub relacja rozmyta). Modele te nie są tak naturalne w interpretacji jak reguły decyzyjne, a ponadto ich konstrukcja nastrocza wiele kłopotów, głównie z powodu trudności z akwizycją specyficznych informacji preferencyjnych typu wag kryteriów, współczynników substytucji, progów preferencji, nierozróżnialności i veta. Tymczasem akwizycja informacji preferencyjnych w przypadku podejścia DRSA jest stosunkowo prosta – decydent podaje przykłady klasyfikacji lub jego decyzje są automatycznie rejestrowane *on line*. Model ten spełnia ponadto postulat przejrzystości i zrozumiałości modeli preferencji we wspomaganiu decyzji.

Zaproponowanie „trzeciej drogi” w modelowaniu preferencji wymagało porównania nowego modelu regułowego z modelami klasycznymi. Autorzy podejścia DRSA dokonali wnikliwej charakteryzacji podstaw aksjomatycznych wszystkich trzech typów modeli preferencji (por. Greco i in., 2001a, 2004; Słowiński i in., 2002), z jednej strony w terminach teorii wspólnego pomiaru (ang. *conjoint measurement*), a z drugiej w terminach reguł decyzyjnych. W porównaniu z innymi tego typu charakteryzacjami operatorów agregacji, aksjomaty zaproponowane we wspomnianych pracach nie wymagają żadnych wstępnych założeń odnośnie do skal kryteriów. Ważnym wnioskiem z tych badań jest udowodniony fakt, że model preferencji w postaci reguł decyzyjnych jest najogólniejszym modelem preferencji wśród wszystkich znanych operatorów agregacji. W (Greco i in., 2002; Słowiński i in., 2002) udowodniono, że funkcja użyteczności lub relacja przewyższania istnieje dla danego zbioru spójnych przykładów wtedy i tylko wtedy, gdy istnieje model regułowy. Reguły decyzyjne wyindukowane z przybliżeń klas decyzyjnych, zbudowanych w oparciu o relację dominacji, mogą ponadto reprezentować preferencje decydenta wyrażone przez zbiór niespójnych przykładów, czego nie potrafią klasyczne modele preferencji (Greco i in. 2004). Niespójność informacji preferencyjnej pochodzącej od decydenta jest typowa dla realnych sytuacji decyzyjnych; niespójność ta wynika z niestałości preferencji w czasie, wahań decydenta oraz niepełnego opisu sytuacji decyzyjnej za pomocą kryteriów i atrybutów.

Wyniki zawarte w pracach (Greco i in. 1999, 2001, 2001a, 2002, 2004; Słowiński i in. 2002, 2002a, 2002b) nadają nową perspektywę modelowaniu preferencji. Modelowanie preferencji za pomocą reguł decyzyjnych nastawione jest bardziej na zrozumienie mechanizmów podejmowania decyzji wyrażonych przez związki logiczne niż na „wyliczenie” decyzji na podstawie funkcji agregującej oceny wielokryterialne do pojedynczej liczby, czy też na podstawie relacji przewyższania w zbiorze wariantów decyzyjnych.

Metodyka konstrukcji regułowego modelu preferencji nie wymaga ponadto od decydenta podania specyficznych parametrów modeli funkcyjnych lub relacyjnych takich, jak: wagi kryteriów, progi nierozróżnialności, preferencji i weta na poszczególnych kryteriach, gdyż konstrukcja reguł dokonuje się z przykładów decyzji, czyli z obserwacji. Jest to znacznie łatwiejsze dla decydenta, a nawet umożliwia tworzenie modelu preferencji bez jego udziału, np. decyzje użytkowników zgłaszających się na portale internetowe lub dokonujących transakcji rejestrowanych elektronicznie, mogą posłużyć do indukcji reguł decyzyjnych reprezentujących ich preferencje.

W zakresie modelowania preferencji metodą indukcji reguł decyzyjnych z przybliżeń klas decyzyjnych według podejścia DRSA zaobserwowano także możliwość wykorzystania komplementarności teorii zbiorów przybliżonych i teorii zbiorów rozmytych (ang. *fuzzy set theory*) (por. Słowiński, 1998) – dotyczą one bowiem różnych rodzajów „nie-doskonałości” danych, odpowiednio – niespójności i nieostrości.

Powyższe wyniki nadają również nową perspektywę inteligencji maszynowej, gdyż włączają wiedzę o preferencjach do wiedzy odkrywanej z danych. W rozproszonych systemach sztucznej inteligencji, zwanych systemami wieloagentowymi (autonomiczne programy działające w sieci komputerowej), powyższy sposób odkrywania wiedzy daje możliwość rozpoznawania preferencji przez agentów na podstawie wzajemnej obserwacji zachowań. W ten sam sposób można rozpoznawać preferencje użytkowników współdzielących zasoby w systemach grid w ramach jednej organizacji wirtualnej (Węglarz, 2003).

Podejście DRSA było także dostosowywane do innych problemów decyzyjnych niż klasyfikacja wielokryterialna. Ważną kategorią wielokryterialnych problemów decyzyjnych są problemy wyboru i porządkowania. W tych problemach decyzje wobec obiektów nie są podejmowane na podstawie ich bezwzględnych ocen, tak jak w problemie klasyfikacji wielokryterialnej, lecz na podstawie względnych ocen wynikających z porównania obiektów parami. Z tego powodu punktem wyjścia do analizy za pomocą podejścia DRSA jest nie tablica decyzyjna z przykładami klasyfikacji, lecz tablica porównań obiektów parami. Wiersze tej tablicy odpowiadają parom obiektów, a kolumny kryteriom warunkowym i decyzji w postaci globalnej relacji preferencji; każda para obiektów w takiej tablicy jest opisana przez cząstkowe relacje preferencji na poszczególnych kryteriach i przez globalną relację preferencji. Celem analizy DRSA jest konstrukcja przybliżeń globalnej relacji przewyższania i indukcja reguł decyzyjnych z tych przybliżeń. Reguły decyzyjne zawierają warunki elementarne dotyczące relacji cząstkowych i decyzję określającą relację globalną dla pary obiektów. Wynik zastosowania tych reguł na zbiorze obiektów może być przedstawiony w postaci grafu relacji preferencji – do wypracowania zalecenia (wyboru lub uporządkowania) można zastosować procedurę eksploatacji grafu typu „bilansowania przepływu” (ang. *net flow score*) (por. Greco i in., 1998, 2001; Słowiński i in. 2002b).

Inne adaptacje DRSA do rozwiązywania problemów decyzyjnych dotyczą m.in.:

- podejmowania decyzji w warunkach ryzyka i niepewności (Greco i in. 2001b),
- analizy niekompletnych danych (Greco i in. 2000),
- powiązań DRSA z podejściem zbiorów rozmytych (Greco i in. 2000a, 2002a, 2004a),
- problemów decyzyjnych z hierarchiczną strukturą atrybutów i kryteriów (Dembczyński i in., 2002),
- indukowania reguł asocjacyjnych z danych uporządkowanych według preferencji (Greco i in. 2002b).

### Odkrywanie wiedzy z danych tekstowych

Spróbujmy wyobrazić sobie olbrzymich rozmiarów bibliotekę; wielopiętrową, z ciągle rosnącym księgozbiorem, z szeregami regałów, na których prężą się dumnie tomy zaszyte w słowa wiedzy. Niestety, biblioteka owa posiada jedną wadę – brak w niej katalogu lub indeksu, a zbiory są rozsiane po całym budynku zupełnie losowo. Tak mniej więcej prezentuje się obraz Internetu. W końcu kwietnia 2004 roku, liczba dokumentów istniejących w Internecie, rozumianych jako strony internetowe, pliki PDF, postscript i wszelkie inne zasoby dające się sprowadzić do formy tekstowej, była określana na ponad 4 miliardy.

Jednak to ani wielkość, ani dynamika wzrostu nie jest czynnikiem rewolucyjnym w Internecie – olbrzymie ilości informacji były już od dawna dostępne w bibliotekach. To łatwość dostępu do całej tej wiedzy stanowi historyczny przełom. Nie musimy już biegać między regałami, by, co gorsza, w końcu dowiedzieć się, że jedyny egzemplarz interesującej nas książki został akurat wypożyczony. Przy korzystaniu z Internetu jedynym problemem jest wiedza o tym, gdzie należy szukać.

Atrakcyjność Internetu jako zbioru danych, w połączeniu z nieefektywnością jego bezpośredniej eksploracji, musiała prowokować do prób tworzenia automatów, które by tę czynność ułatwiły. Najbardziej rozpowszechnione owoce tych prób to oczywiście *wyszukiwarki internetowe* (ang. *search engine*), które skupiają w sobie najnowsze osiągnięcia wielu specjalności informatyki: baz danych, przetwarzania rozproszonego, odkrywania wiedzy, analizy grafowej, a nawet lingwistyki. Analiza zasobów Internetu w celu pozyskania z niego użytecznej informacji została nazwana *eksploracją danych internetowych* (ang. *Web mining*).

Wróćmy jeszcze do wyszukiwarek. Składają się one zwykle z wielu małych programów, zwanych agentami, pajakami lub terminem zapożyczonym z języka angielskiego – *crawlerami*. Programy te bezustannie przechodzą między witrynami w Internecie, zapisując jednocześnie informację o treści danej witryny, miejscu, gdzie można ją znaleźć, oraz wszystkich słowach, jakie ona zawiera. Następnie dla ogółu znanych wy-

szukiwarce dokumentów tworzona jest bardzo istotna struktura danych zwana *indeksem odwrotnym* (ang. *inverted index*). Indeks odwrotny to zbiór wszystkich unikalnych słów (ciągów znaków), jakie napotkała wyszukiwarka. Dla każdego słowa w indeksie tworzona jest lista dokumentów, które owe słowo zawierają. Indeks odwrotny umożliwia wyszukiwarce błyskawiczne wyznaczenie zbioru dokumentów, które „pasują” do zapytania użytkownika – jest nim po prostu lista dokumentów powstająca z przecięcia (części wspólnej) poszczególnych list odnalezionych dla wszystkich słów tego zapytania w indeksie odwrotnym.

Od razu widać, że proces wyszukiwania nie zawiera w sobie żadnych semantycznych mechanizmów, a opiera się jedynie na współwystępowaniu słów w dokumencie. Na przykład dla informatyka zapytanie „sztuczne życie” jednoznacznie kojarzy się z dziedzinami informatyki inspirowanymi biologią (ang. *artificial life*). Jednakże dla wyszukiwarki wszystkie inne kombinacje semantyczne tych słów są równie poprawne i także będą umieszczone w wyniku! Innymi słowy, wyszukiwarka nie ma możliwości poznania semantycznego kontekstu zapytania. Co gorsza, jak pokazuje praktyka, człowiek dąży do lapidaryzacji wszelkich form interakcji z komputerem; zapytania użytkowników są więc bardzo często ogólne, niejednoznaczne i dotyczą wielu tematów, które niekoniecznie pokrywają się z oczekiwaniami. Niebagatelnym zadaniem jest więc modelowanie „typowego” użytkownika (ang. *Web surfer*) i posortowanie dokumentów wynikowych według jego najbardziej prawdopodobnych preferencji w ten sposób, by hipotetycznie najlepsze dokumenty znalazły się na początku listy. Problemem tym zajmują się algorytmy rangujące (ang. *ranking algorithms*), z tyle oryginalnym, co efektywnym algorytmem PageRank (por. Brin i Page, 1998) zastosowanym w wyszukiwarce Google.

Ostatnią czynnością, jaką wyszukiwarka musi wykonać po wyznaczeniu zbioru wyników oraz jego posortowaniu, jest jego wyświetlenie na ekranie komputera użytkownika. Najbardziej rozpowszechniona metoda prezentacji to wyświetlanie dla każdego dokumentu tytułu (nadanego przez autora dokumentu), jego adresu URL (ang. *Uniform Resource Locator*) oraz automatycznie utworzonego streszczenia (ang. *snippet*). Streszczenie owo jest zwykle kontekstowe, to znaczy pokazuje fragmenty tekstu w sąsiedztwie słów należących do zapytania (ang. *query in context*), co często pozwala na odgadnięcie tematyki dokumentu z pominięciem wglądu do jego pełnej wersji.

Powyższa technika prezentacji wyników w wyszukiwaniu jest najbardziej rozpowszechniona i spotkać ją można praktycznie we wszystkich dużych komercyjnych serwisach wyszukiwających. Metoda ta sprawdza się znakomicie, gdy użytkownik wie, czego oczekuje i potrafi zmodyfikować zapytanie w razie uzyskania mało interesujących wyników. Jednak w przypadku zapytań ogólnych, gdy użytkownik chce uzyskać ogólne rozeznanie w temacie lub gdy nie wie dokładnie, jakim zapytaniem może dotrzeć do interesującej go informacji, prezentacja zwykłej listy dokumentów wydaje się nie być wystarczająca.

The image shows a screenshot of a Google search interface. At the top, the Google logo is on the left, and navigation links for 'WWW', 'Grafika', 'Grupy dyskusyjne', and 'Katalog' are on the right. A search bar contains the text 'sztuczne życie'. To the right of the search bar is a 'Szukaj' button and a link to 'Zaawansowane szukanie Ustawienia'. Below the search bar, there are radio buttons for 'Szukaj w Internecie' (selected) and 'Szukaj na stronach kategorii: Polski'. A horizontal bar below the search bar displays 'WWW' on the left and 'Wyniki 1 - 10 spośród około 17,600 dla zapytania sztuczne życie. (Znaleziono w 0.38 sek.)' on the right. The search results are listed below, each with a title, a brief description, and a URL with additional links like 'Kopia' and 'Podobne strony'. The first result is from 'Portal Sztucznego Życia: alife.pl' with the title 'Sztuczne życie, sztuczna ...' and a description about artificial intelligence. The second result is from the same portal with the title 'Fascynujący świat ...' and a description about the nature of artificial life. The third result is from 'www.frams.poznan.pl/pindex.html' with the title 'Podobne strony'. The fourth result is from 'Sztuczna Inteligencja :: Wywiady' with the title 'Wywiady' and a description about artificial intelligence. The fifth result is from 'Reporter nr 03 - 1999.03.10. Gość numeru: Maciej Komosiński' with the title 'Maciej Komosiński. Sztuczne życie, sztuczna inteligencja - co nas czeka w przyszłości? ... Sztuczne życie' and a description about the future of artificial intelligence. The sixth result is from 'aracinowski.republika.pl' with the title 'Sztuczne życie' and a description about artificial intelligence.

WWW Grafika Grupy dyskusyjne Katalog

Google sztuczne życie Szukaj Zaawansowane szukanie Ustawienia

Szukaj w Internecie Szukaj na stronach kategorii: Polski

WWW Wyniki 1 - 10 spośród około 17,600 dla zapytania sztuczne życie. (Znaleziono w 0.38 sek.)

Portal Sztucznego Życia: alife.pl - **Sztuczne życie, sztuczna ...**  
**Sztuczne życie**, sztuczna inteligencja. **Sztuczne Życie**, Sztuczna Inteligencja –  
 co nas czeka w przyszłości? ... Co się kryje pod nazwą „**sztuczne życie**”? ...  
 www.alife.pl/portala/articles/p/komosinski.html - 30k - [Kopia](#) - [Podobne strony](#)

Portal Sztucznego Życia: alife.pl - I. Fascynujący świat ...  
 ... Jest to właśnie **sztuczne życie** czyli dosłownie „**Życie** stworzone przez Człowieka,  
 a nie przez Naturę”. ... W książce Ellen Thro „**Sztuczne życie**. ...  
 www.alife.pl/portala/articles/p/piwowarczyk1.html - 19k - [Kopia](#) - [Podobne strony](#)  
 [ [Więcej wyników z www.alife.pl](#) ]

[www.frams.poznan.pl/pindex.html](http://www.frams.poznan.pl/pindex.html)  
[Podobne strony](#)

[Sztuczna Inteligencja :: Wywiady](#)  
 ... **Sztuczne życie**, sztuczna inteligencja - co nas czeka w przyszłości? **Sztuczne**  
**Życie**, Sztuczna Inteligencja - co nas czeka w przyszłości? ...  
 www.bassmedia.art.pl/si/interview.php?id=2 - 26k - [Kopia](#) - [Podobne strony](#)

[Reporter nr 03 - 1999.03.10. Gość numeru: Maciej Komosiński](#)  
 ... Maciej Komosiński. **Sztuczne życie**, sztuczna inteligencja - co nas czeka w przyszłości? ... **Sztuczne**  
**życie**" brzmi interesująco i trochę przerażająco. ...  
 magazyn.reporter.pl/1999/03/r04.html - 24k - [Kopia](#) - [Podobne strony](#)

**Sztuczne życie**  
 aracinowski.republika.pl/  
 www.myster.prv.pl/ - 1k - [Kopia](#) - [Podobne strony](#)

Ryc. 2. Wynik wyszukiwania dla zapytania „sztuczne życie” w serwisie Google

Parę lat temu mogło się jeszcze wydawać, że lukę tę wypełnią *serwisy katalogowe*, czyli manualnie tworzone przez człowieka struktury, które w logiczny sposób organizują informację dostępną w Internecie. Serwisy te są oczywiście nadal dostępne (np. *Open Directory Project*, *Yahoo!*) lecz wyraźnie tracą na znaczeniu w porównaniu z wyszukiwarkami wzbogaconymi o *automatyczne grupowanie wyników wyszukiwania* (ang. *search results clustering*). Automatyczne grupowanie wyników wyszukiwania jest metodą *odkrywania wiedzy w danych tekstowych* i ma na celu automatyczne odkrycie logicznej struktury semantycznych powiązań dokumentów w wynikach wyszukiwania oraz utworzenie dla grup podobnych dokumentów zwięzłego opisu. Użyteczność grupowania wydaje się bezdyskusyjna, co widać po rosnącej popularności komercyjnych systemów grupujących (Vivisimo, Grokker) oraz systemów rozpowszechnianych na zasadach „wolnego oprogramowania”. Niepoślednim przykładem tych ostatnich jest system Carrot<sup>2</sup> współpracujący z wyszukiwarką Google i dostosowany do zapytań zarówno w języku polskim, jak i angielskim. Carrot<sup>2</sup> powstał w Zakładzie Inteligentnych Systemów

mów Wspomagania Decyzji Instytutu Informatyki Politechniki Poznańskiej i jest publicznie dostępny (opis projektu Carrot<sup>2</sup> i sama aplikacja są dostępne pod adresem: <http://www.cs.put.poznan.pl/dweiss/carrot2> oraz <http://carrot.cs.put.poznan.pl>)

The screenshot shows the Carrot2 search interface. At the top, there are navigation links: 'komponenty', 'administracja', 'duże zapytanie', 'demonstracja', and 'czym jest Carrot?'. The search query 'sztuczne życie' is entered in the search bar. Below the search bar, the process is identified as 'Google, LINGO, Dynamic Tree' and the number of results is set to 100. On the left, a 'sub topics' sidebar lists 129 groups, including 'Sztuczne Inteligencja (16)', 'Sztuczne Serce (8)', 'Życie po Życiu (11)', 'Magazyn (7)', 'Algorytmy Genetyczne (6)', 'Nauki Technika Biologia inne HOGA (5)', 'PI (7)', 'Życie Dnia (3)', 'Tworzywa Sztuczne (3)', 'Linki (3)', 'Wirtualny Wszechświat (3)', 'Programowanie (3)', 'Artykuły (5)', 'Życie na Ziemi (2)', 'Nauki KOGNITYWNE Marek Kasperski (3)', 'Maciej Komosiński (2)', 'Sztuczne Zapłodnienie (2)', 'Biotechnologia Sztuczne Życie Sztuka (2)', 'Zwierzętach (2)', 'Sztuczne Życie Nowa (3)', 'Życie Uniwersalne (2)', 'Czym (2)', and '(Other) (32)'. On the right, two search results are displayed: result 63, 'Sztuczne zapłodnienie', and result 75, 'KATECHETA 4/01 - W trosce o godność życia ludzkiego. Sztuczne ...'.

Ryc. 3. Grupy utworzone przez system Carrot<sup>2</sup> dla 100 wyników wyszukiwania zwróconych dla zapytania „sztuczne życie” przez wyszukiwarkę Google

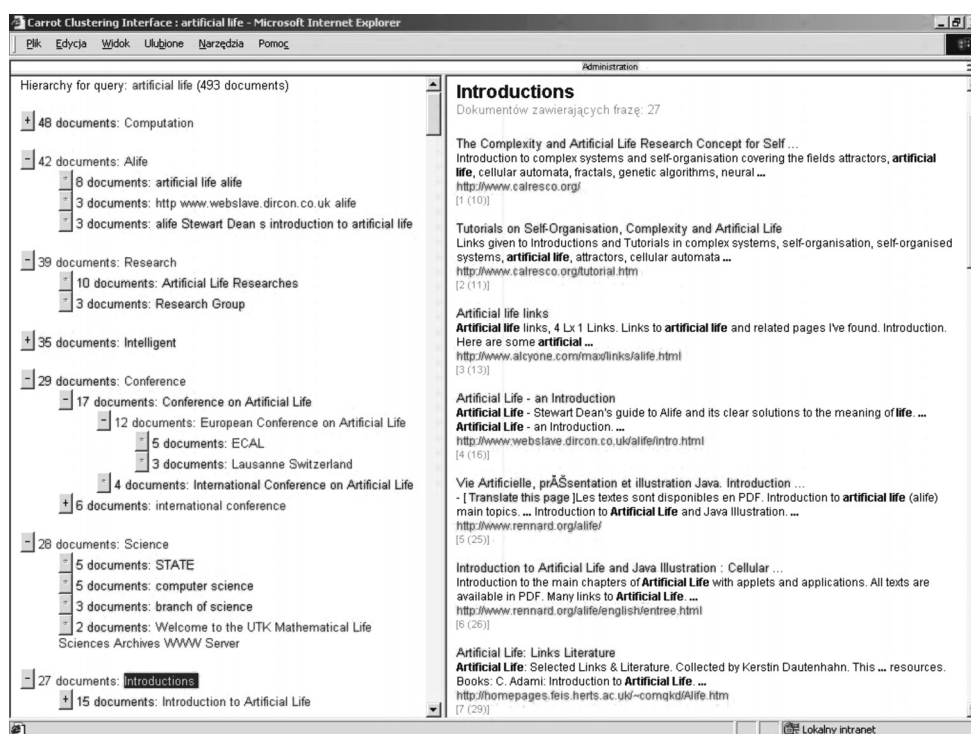
Dla przykładu, porównajmy wyniki wyszukiwania dla zapytania „sztuczne życie” uzyskanych z wyszukiwarki Google (ryc. 2), z grupami, które utworzył system grupujący Carrot<sup>2</sup>. „Surowy” wynik z Google, co widać na rycinie 2, jest zdominowany przez dokumenty dotyczące tematyki sztucznego życia. Grupowanie wyników pozwoliło na dotarcie do innych ciekawych tematów – biotechnologii, nauk kognitywnych, czy też prezentowanych na rycinie 3 dokumentów o naturze filozoficznej i religijnej. Co ciekawe, grupowanie pozwoliło również naświetlić luki w omówionej metodzie działania wyszukiwarek – grupa „tworzywa sztuczne” powstała dlatego, iż wyszukiwarka potraktowała jako pasujące do zapytania dokumenty, zawierające fragment tekstu „weszła w życie ustawa o substancjach [...] i tworzywa sztuczne”.

Analiza dokumentów tekstowych i ich grupowanie nie jest dziedziną nową. Zarówno metody ekstrakcji cech z tekstu, jak i ich zastosowanie do analizy skupień mają swoje korzenie w dziedzinie odkrywania wiedzy i wyszukiwania informacji (ang. *information retrieval*). Okazało się jednak, że w zakresie grupowania wyników z wyszukiwarek





Niestety, okazało się, że w praktyce algorytmy oparte na identyfikacji częstych fraz mają tendencję do preferowania pojedynczych większych tematów i ich podzbiorów, co powoduje powtarzanie się tych samych dokumentów w różnych grupach (redundancja). Pojawił się również problem z sekwencjami słów często występujących w tekście, a stanowiących szum. Próby użycia bardziej złożonych metod analizy skupień, powracających do paradygmatu porównywania dokumentów na podstawie współwystępowaniu pojedynczych słów, pozwoliły uzyskać lepsze wyniki w testach typu *precision-recall*, lecz ich praktyczne użycie jest mocno ograniczone właśnie przez trudności z pozyskaniem czytelnych opisów dla wykrytych grup.



Ryc. 5. Hierarchia grup dla zapytania „artificial life” oraz dokumenty z grupy „introductions”

Wydaje się, że powyższe problemy z redundancją i zaszumieniem oraz z czytelnością struktury grup został zadowalająco rozwiązany przez metodę HSTC (Masłowska i Słowiński, 2003). Metoda ta indukuje strukturę hierarchiczną w redundantnym zbiorze grup dokumentów utworzonym przez algorytm STC. Punktem wyjścia do budowy hierarchii jest graf nieprzechoďnej, zwrotnej i asymetrycznej relacji wzajemnego

zawierania się grup dokumentów. Hierarchia ma strukturę drzewa, w którym kolejna warstwa rozgałęzień jest utworzona z grup stanowiących jądro grafu relacji zawierania – grupy należące do jądra mają tę właściwość, że nie zawierają się nawzajem, a każda grupa spoza jądra jest bezpośrednio zawierana przez przynajmniej jedną grupę z jądra. Powstałe drzewo hierarchii znakomicie ułatwia eksplorację całej przestrzeni grup. Przykładowe hierarchie grup utworzone metodą HSTC dla zapytania „artificial life” przedstawiono na rycinach 4 i 5. W obu przypadkach grupy dokumentów powstały przez grupowanie wyników zwróconych przez wyszukiwarkę Google, z tym że na rycinie 4 wzięto pod uwagę pierwszych 591 dokumentów z rankingu, a na rycinie 5, 493.

Większość z opisanych powyżej algorytmów została zaprojektowana z myślą o języku angielskim, co znajduje odbicie w słabszej jakości uzyskiwanych przez nie wyników dla języków o innej strukturze składniowej zdania (por. Stefanowski i Weiss, 2003). Istniejące algorytmy zwykle koncentrowały się również na tworzeniu struktury skupień dokumentów, pomijając lub kładąc mniejszy nacisk na znalezienie właściwego opisu dla tych skupień. Nowatorskim pomysłem okazało się rozdzielenie procesu analizy skupień i poszukiwania dobrych opisów grup, wprowadzone w algorytmie Lingo (Osiński i Weiss, 2004). Algorytm ów najpierw wyszukuje ukryte zależności między dokumentami (ang. *latent semantic relationships*) przy pomocy metod spektralnej dekompozycji macierzy. Powstałe w ten sposób grupy są dobrze zróżnicowane, czyli, mówiąc językiem specjalistów od dekompozycji, „tematycznie ortogonalne”. Skupienia te nie posiadają jednak żadnego sensownego opisu w języku naturalnym. Dlatego też w drugiej fazie algorytmu następuje dopasowanie grup do (niepospolitych) fraz, często występujących w analizowanych tekstach. Jeśli dla danej grupy nie uda się znaleźć dobrego dopasowania do opisu, to algorytm w ogóle tę grupę ignoruje, wychodząc z założenia, iż nie byłoby z niej pożytku dla użytkownika. Algorytm Lingo jest wykorzystywany przez system Carrot<sup>2</sup> – przykładowy wynik jego działania pokazano na rycinie 2.

### Odkrywanie wiedzy z danych obrazowych

Korzystając na co dzień z percepcji wzrokowej traktujemy ją jako zdolność naturalną i nieskomplikowaną. Przywykliśmy do tego, że zarówno zadania realizowane w sposób świadomy (np. wyłowienie z tłumy twarzy znajomej osoby), jak i podświadomy (np. ocena położenia i rozmiarów omijanej przeszkody) nie sprawiają nam najmniejszych trudności. Choć w naszym ośrodkowym układzie nerwowym przetwarzane są miliony bitów danych na sekundę, nie zdajemy sobie z tego sprawy, ponieważ do naszej świadomości dociera co najwyżej kilka informacji, wyselekcjonowanych i przydatnych w kontekście aktualnie wykonywanych przez nas czynności. Dzięki selekcji informacji i – nie do końca zbadanym – mechanizmom jej przechowywania, łatwo rozpoznajemy na przykład karykaturę znanej postaci, nawet jeśli powstała kilkoma pociągnięciami ołówka.

Natura przetwarzania informacji w naszym układzie nerwowym jest odmienna od obliczeń realizowanych przez komputery. To, co realizowane jest w sposób równoległy przy użyciu złożonej sieci neuronowej w ośrodkowym układzie nerwowym, okazuje się bardzo trudne w realizacji technicznej. W konsekwencji, sposób przeprowadzania analizy obrazu i cechy z niego ekstrahowane przez maszynę różnią się zazwyczaj całkowicie od naszej percepcji wzrokowej.

Większość konstruowanych współcześnie systemów wnioskowania na podstawie danych obrazowych wymaga jawnego (*a priori*) konstruowania bazy wiedzy przez eksperta (projektanta systemu). Baza wiedzy zawiera w tym przypadku pewne 'reguły percepcji' i składa się zazwyczaj z sekwencji procedur przetwarzania, analizy i rozpoznawania obrazu. Takie podejście do konstruowania, choć skuteczne w wielu zastosowaniach, jest czasochłonne, kosztowne i prowadzi do wąsko specjalizowanych, mało uniwersalnych algorytmów. Konsekwencją tego stanu rzeczy jest daleko idąca i silnie powiązana z zastosowaniami praktycznymi specjalizacja poszczególnych nurtów rozpoznawania obrazów (np. do rozpoznawania pisma ręcznego używa się odmiennych metod niż do rozpoznawania pisma maszynowego).

W obliczu rosnących możliwości obliczeniowych, oferowanych przez współczesne komputery, bardziej naturalne wydaje się dążenie do tego, aby system wnioskujący na podstawie informacji obrazowej pozyskiwał wiedzę dotyczącą percepcji obrazu i konkretnego zastosowania w wyniku uczenia, np. uczenia się z przykładów (ang. *learning from examples*), będących w tym kontekście obrazami. Takie podejście odpowiada uzasadnionemu badaniom psychologicznymi rozpoznawaniu aktywnemu, które zakłada, że interpretacja obrazu jest iteracyjnym procesem stawiania i weryfikowania hipotez (por. Aloimonos, 1993). W ten sposób zbliżamy się do rzeczywistego odkrywania wiedzy z informacji obrazowej.

Podobnie jak w odkrywaniu wiedzy z danych ilościowych i jakościowych oraz z danych tekstowych, tak i tu poszukujemy hipotezy wyjaśniającej, dlaczego, na przykład, dany obraz mikroskopowy skrawka histologicznego został zdiagnozowany przez lekarza jako obraz nowotworu X.

Dążąc do syntezy uczenia maszynowego i wnioskowania (klasyfikowania) z informacji obrazowej, ekspert zaopatruje system uczący się w proste operatory selekcji pola widzenia oraz selekcji i konstrukcji cech, a następnie inicjuje proces uczenia polegający na konstrukcji programu wnioskującego na podstawie informacji obrazowej. W tym podejściu sam system uczący się określa sposób ekstrakcji cech analizowanych obrazów. Pętla sprzężenia zwrotnego, która w tradycyjnych algorytmach uczenia się z przykładów steruje przeglądaniem przestrzeni hipotez, tu obejmuje także proces wstępnego przetwarzania danych uczących (ekstrakcję cech). System uczący się otrzymuje w ten sposób szersze możliwości formułowania hipotez dotyczących analizowanych danych.

Zabieg ten nazywany jest w literaturze transformacją reprezentacji (ang. *transformation of representation*), konstrukcją cech (ang. *feature construction*), konstruktywną indukcją cech (ang. *constructive induction of features*) bądź syntezą cech (ang. *feature synthesis*).

Konstrukcja programu wnioskującego polega na budowaniu procedury ekstrakcji złożonych cech drogą ewolucyjnego doboru prostych operatorów. Proces ewolucyjny moderowany jest przez funkcję oceniającą, która przypisuje każdemu osobnikowi ocenę (dopasowanie) na podstawie skuteczności dyskryminacyjnej definiowanych przez niego cech. Skuteczność ta estymowana jest w wyniku uczenia i testowania przeprowadzanego w ramach tzw. zbioru uczącego (zbioru obrazów ze znaną klasyfikacją). Najlepszy osobnik wykształcony w procesie symulowanej ewolucji stanowi podstawę konstrukcji końcowego systemu rozpoznawania, który może być następnie oceniony na tzw. zbiorze testującym (zbiorze obrazów ze znaną klasyfikacją, który nie był wykorzystywany w procesie uczenia).

Powyższy sposób odkrywania wiedzy z danych obrazowych ma jeszcze jedną istotną zaletę. Pozostawiając systemowi swobodę w konstruowaniu programu wnioskującego, unika się tradycyjnego rozróżniania etapów tworzenia opisu i interpretacji obrazu (np. rozpoznawania) na podstawie tegoż opisu. Omija się w ten sposób potrzebę jawnego określenia sposobu reprezentacji obrazu przed procesem uczenia (co jest zazwyczaj trudne z racji objętości i strukturalnego charakteru danych obrazowych).

Przydatność powyższej metodyki potwierdzona została w praktyce, m.in.:

- we wspomaganiu diagnozowania nowotworów ośrodkowego układu nerwowego na podstawie obrazów mikroskopowych skrawków histologicznych (Jelonek i in., 1999),
- w rozpoznawaniu przedmiotów (Krawiec i Bhanu, 2003),
- w identyfikacji osób na podstawie obrazu twarzy (Krawiec, 2001),
- w rozpoznawaniu obrazów radarowych (Krawiec i Bhanu, 2003a).

We wszystkich wyżej wymienionych zastosowaniach uzyskano zachęcające wyniki przy użyciu tej samej metodyki, dostosowując ją jedynie do specyfiki zastosowania przez odpowiedni dobór prostych operatorów wykorzystywanych przez system uczący.

Poza wcześniej wymienionymi zaletami, inną istotną zaletą rozwijanego podejścia jest jawna (czytelna, symboliczna) reprezentacja hipotez (programów) konstruowanych w procesie uczenia. Daje to możliwość wglądu w uzyskane rozwiązania i, na przykład, dalszego, ręcznego udoskonalania ich. Eksperymenty obliczeniowe przeprowadzone z użyciem komputerowej implementacji proponowanej metody wykazały zdecydowanie lepszą czytelność pozyskanej wiedzy w porównaniu z podejściami konwencjonalnymi. Sprawia to, że proponowane podejście jest szczególnie przydatne w odkrywaniu wiedzy na podstawie informacji obrazowej, rozumianym jako proces poszukiwania nowych,

interesujących, potencjalnie użytecznych i zrozumiałych dla użytkownika wzorców informacyjnych w danych obrazowych.

W ostatnich pracach poświęconych temu podejściu (por. Krawiec, 2004) reprezentuje się procedury ekstrakcji cech w sposób inspirowany liniowym programowaniem genetycznym (ang. *linear genetic programming* (por. Nordin i in., 1999)). Przy tej reprezentacji procedura ekstrakcji cech jest sekwencją (parametryzowanych) wywołań elementarnych operatorów.

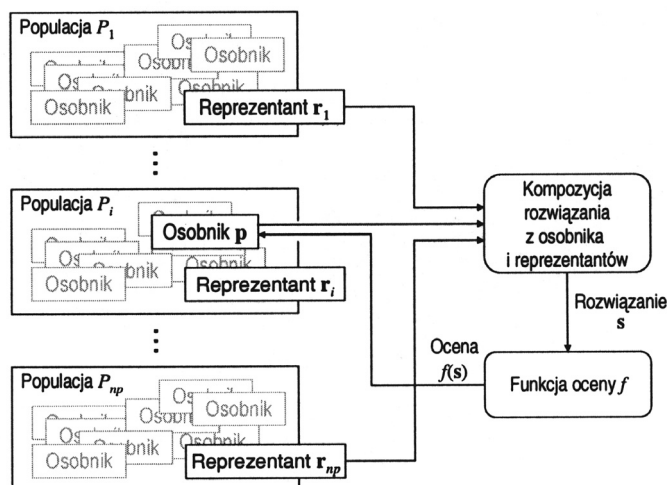
W następnych podpunktach scharakteryzujemy dwa oryginalne aspekty powyższego podejścia do odkrywania wiedzy z danych obrazowych.

#### *Dekompozycja zadania konstrukcji cech przez koewolucję*

Zadanie ekstrakcji, a właściwie konstrukcji cech, sformułowane w poprzednim punkcie jest złożone zarówno pod względem koncepcyjnym, jak i obliczeniowym. Z drugiej strony – jest ono częściowo dekomponowalne. Dekomponowalne, ponieważ dla większości nietrywialnych problemów rozpoznawania potrzebujemy więcej niż jednej cechy do skutecznego dyskryminowania rozpoznawanych klas obiektów. Częściowo, gdyż bezcelowe jest np. stworzenie wzajemnie odseparowanych agentów, z których każdy wypracowuje pojedynczą cechę – taka pełna dekompozycja nie da dobrych rezultatów, ponieważ poszczególne cechy muszą wchodzić w użyteczne (synergiczne) interakcje, aby wspólnie skutecznie dyskryminować rozpoznawane obiekty w przestrzeni cech.

Częściowa dekomponowalność zadania konstrukcji cech czyni je dogodnym polem dla zastosowania koewolucji kooperatywnej (inaczej: algorytmy współewoluujące (por. Michalewicz, 1996), ang. *cooperative coevolution* (por. także Porter i De Jong, 2000)). Jest to wariant obliczeń ewolucyjnych, w którym w miejsce jednej populacji osobników wprowadza się wiele populacji (ryc. 6). Osobniki w każdej populacji nie reprezentują kompletnych rozwiązań badanego problemu, a jedynie ich części (rozwiązania częściowe). W konsekwencji, ocena osobników nie może odbywać się w każdej populacji niezależnie. Osobniki pochodzące z różnych populacji muszą być agregowane w celu utworzenia kompletnych rozwiązań, które można ocenić globalną funkcją oceny  $f$ .

W szczególności, każda populacja utrzymuje tzw. *reprezentanta*  $\mathbf{r}$ , którym jest najlepszy dotąd znaleziony w tej populacji osobnik. Gdy zachodzi potrzeba oceny osobnika  $\mathbf{p}$  z pewnej populacji, z osobnika  $\mathbf{p}$  oraz reprezentantów  $\mathbf{r}_i$  wszystkich pozostałych populacji komponowane jest kompletne rozwiązanie  $\mathbf{s}$  (ryc. 6). Rozwiązanie to podlega ocenie funkcją oceny  $f$ , a otrzymana wartość przypisywana jest jako ocena (dopasowanie, ang. *fitness*) ocenianemu osobnikowi  $\mathbf{p}$ . Poza fazą oceny rozwiązań, pozostałe etapy procesów ewolucyjnych (selekcja, krzyżowanie, mutacja) przebiegają dla każdej populacji niezależnie.



Ryc. 6. Równoległe ewoluowanie cech i ich wspólna ocena dla zadania rozpoznawania

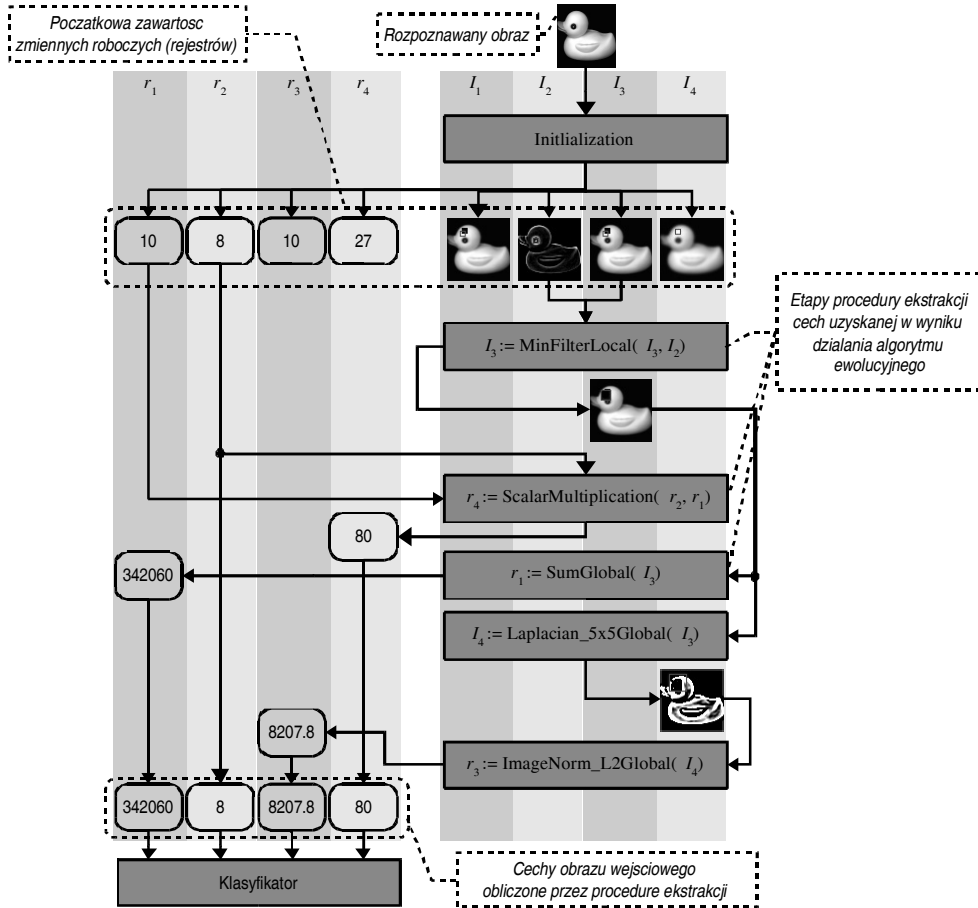
Podstawową zaletą koewolucji kooperatywnej jest zatem możliwość dekompozycji zadania na podzadania (wypracowanie jednej cechy przez każdą populację) bez konieczności specyfikowania osobnych (lokalnych) funkcji celu dla poszczególnych podzadań.

Wyniki eksperymentu obliczeniowego, dotyczącego rozpoznawania przedmiotów codziennego użytku na podstawie monochromatycznych, dwuwymiarowych obrazów pozyskanych w paśmie widzialnym (baza COIL-20, popularny problem testowy w dziedzinie widzenia komputerowego, który można znaleźć pod adresem:

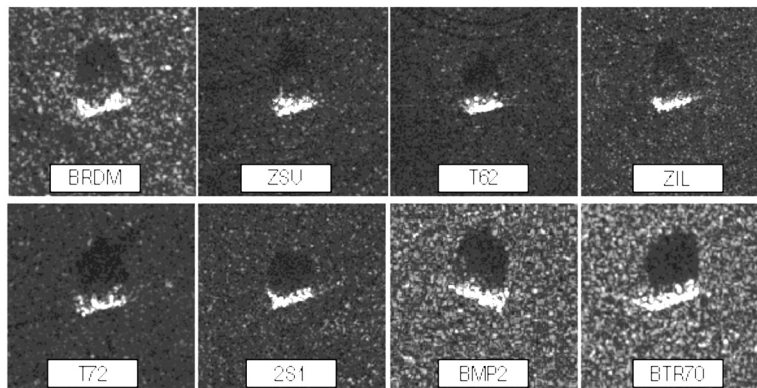
<http://www1.cs.columbia.edu/CAVE/research/softlib/coil-20.html>),

wskazują na wysoką przydatność tej metody (Krawiec i Bhanu, 2003, 2003a).

Rycina 7 przedstawia przykładową procedurę ekstrakcji cech, będącą wynikiem zastosowania koewolucji kooperatywnej do problemu COIL-20. Schemat ten prezentuje zarówno samą procedurę, jak i przebieg przetwarzania przez nią obrazu należącego do wybranej klasy decyzyjnej (*Duck*). Poszczególne bloki (*MinFilterLocal*, *ScalarMultiplication*, itd.) odpowiadają elementarnym operacjom wykonywanym przez procedurę na obrazie (np. *MinFilterLocal*) lub na wielkościach skalarnych obliczonych na podstawie obrazu (np. *ScalarMultiplication*). Operacje działają na rejestrach, czyli zmiennych roboczych; w nich przechowywane są wyniki pośrednie i końcowe (przetworzone obrazy, pośrednie wartości skalarnie, końcowe wartości cech). Przedstawiana tu procedura operuje na czterech rejestrach numerycznych ( $r_1 \dots r_4$ ) oraz czterech rejestrach obrazowych ( $I_1 \dots I_4$ ); ich obecność zilustrowana jest szarymi pasami umieszczonymi w tle rysunku. Kończącym wynikiem działania procedury są wartości czterech cech skalarnych.



Ryc. 7. Przykładowa procedura ekstrakcji cech otrzymana przy pomocy koewolucji kooperatywnej dla testowego zadania rozpoznawania przedmiotów (COIL-20)



Rys. 8. Wybrane obrazy SAR rozpoznawane w przeprowadzonych eksperymentach

Wartości te opisują obraz jako przykład w sensie uczenia maszynowego i reprezentacji atrybut-wartość. Na ich podstawie nauczony klasyfikator dokonuje rozpoznania analizowanego obrazu.

Równie zachęcające wyniki otrzymano dla bardziej skomplikowanego zadania rozpoznawania pojazdów na obrazach radarowych (ang. *synthetic aperture radar*, SAR). W tym trudnym zastosowaniu obrazy charakteryzują się niską rozdzielczością, znacznym zaszumieniem oraz brakiem ‘dosłownych’ cech, jak np. krawędzie, tak charakterystycznych dla obrazowania w paśmie widzialnym. Rycina 8 przedstawia obrazy wybranych klas pojazdów rozpoznawanych w ramach przeprowadzonych eksperymentów. Mimo tych trudnych właściwości, koewolucyjna odmiana omawianego podejścia była w stanie zsyntetyzować systemy rozpoznawania charakteryzujące się bardzo wysokimi zdolnościami predykcyjnymi.

Otrzymane wyniki wskazują na wysoką skuteczność proponowanego podejścia, porównywalną z rezultatami osiąganymi przy użyciu metod wymagających ręcznego doboru sposobu ekstrakcji cech. W szczególności, rezultaty otrzymane z wykorzystaniem koewolucji są znacząco lepsze od przypadku, w którym wykorzystywano pojedynczy proces ewolucyjny, co potwierdza tezę o częściowej dekomponowalności tego problemu.

#### *Współdzielenie wiedzy pomiędzy zadaniami uczenia*

Centralnym problemem nękającym algorytmy uczenia indukcyjnego jest przeuczenie (ang. *overfitting*). Zjawisko to polega na osiągnięciu przez klasyfikator niższej trafności klasyfikowania na zbiorze testującym niż na zbiorze uczącym. Innymi słowy, przeuczony klasyfikator słabo uogólnia wiedzę pozyskaną ze zbioru uczącego na nowe instancje tego samego problemu. Przyczyną przeuczenia jest zazwyczaj zbyt wyspecjalizowany klasyfikator (np. drzewo decyzyjne o zbyt dużej liczbie węzłów, sieć neuronowa o zbyt dużej liczbie neuronów i/lub zbyt dużych wartościach wag).

Algorytm uczenia maszynowego wyposażony w możliwość konstrukcji cech jest szczególnie podatny na przeuczenie, ponieważ zyskuje dodatkowe „stopnie swobody” umożliwiające mu „dopasowanie się” do danych uczących. W omawianym podejściu, wykorzystującym konstrukcję cech do odkrywania wiedzy z danych obrazowych, dla danego zadania istnieje zazwyczaj wiele cech, które zapewniają bardzo dobrą lub nawet perfekcyjną dyskryminację klas decyzyjnych w ramach zbioru uczącego, ale tylko niektóre z nich sprawują się równie dobrze na zbiorze testującym.

Tradycyjnie przeuczeniu zapobiega się, stosując zasadę „brzytwy Ockhama”, co technicznie polega na takim doborze parametrów algorytmu uczenia, aby nie pozwolić na „przespecjalizowanie” budowanego klasyfikatora. O skuteczności doboru parametrów nie można się jednak przekonać inaczej niż poprzez test klasyfikatora na zbiorze

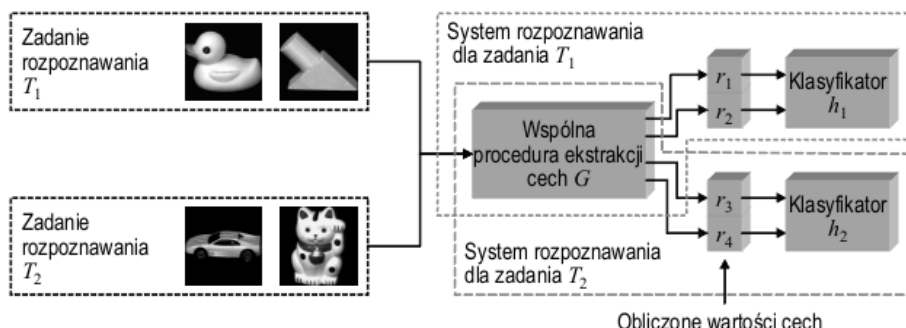


testującym, który to zbiór jest przecież niedostępny w trakcie uczenia. Dlatego też zdolność predykcyjną klasyfikatora (hipotezy) często estymuje się, przeprowadzając eksperyment uczenia i testowania w ramach zbioru uczącego (tzw. *wrapper* (por. Kohavi i John, 1997)).

Punktem wyjścia alternatywnego sposobu zapobiegania przeuczeniu (Krawiec, 2004) jest spostrzeżenie, że większość konwencjonalnych algorytmów uczenia maszynowego nie kumuluje wiedzy, tj. nie modyfikuje podejścia do jednego zadania rozpoznawania (indukcji klasyfikatora) na podstawie doświadczenia z innym zadaniem rozpoznawania. Przejawem tego jest ściśle rozgraniczenie na algorytm indukcji (np. generator drzew decyzyjnych) i klasyfikator (np. drzewo decyzyjne). Wiedza pozyskana w trakcie uczenia zostaje odzwierciedlona jedynie w klasyfikatorze, zaś algorytm indukcji nie podlega żadnym zmianom.

Choć takie rozwiązanie ma pewne zalety (np. powtarzalność eksperymentów), to z praktycznego punktu widzenia brak transferu wiedzy między kolejnymi uruchomieniami tego samego algorytmu uczenia jest marnotrawstwem. Choć każde zadanie uczenia jest inne, istnieją pewne ich wspólne właściwości, które można by potencjalnie wykorzystać, aby zapobiegać przeuczeniu. Wspólne właściwości przydatne w wielu zadaniach uczenia nazywa się *metawiedzą*, a proces pozyskiwania i wykorzystywania metawiedzy – *metauczeniem*. W przypadku uczenia na podstawie informacji obrazowej ilość wiedzy, jaka może być współdzielona pomiędzy zadaniami, jest znacząco większa w porównaniu z uczeniem z przykładów zadanych w postaci wektorów atrybut-wartość. Zarówno badania neurofizjologiczne, jak i praktyka projektowania systemów rozpoznawania obrazów wskazują, iż repertuar podstawowych cech wykorzystywanych w dość odmiennych zadaniach jest bardzo zbliżony. Na przykład pojęcia krawędzi czy obszaru nadają się do opisu różnych scen i obiektów.

Powyższe spostrzeżenia można wykorzystać do przedefiniowania opisanych wcześniej systemów rozpoznawania obrazów z konstrukcją cech. Załóżmy, że zadaniem systemu uczącego się jest wyewoluowanie systemów rozpoznawania dla dwóch zadań:  $T_1$  i  $T_2$  (ryc. 9). Idea polega na uwspólnieniu części procesu ekstrakcji cech. Technicznie można to zrealizować, poddając procesowi ewolucji jedną (wspólną) procedurę ekstrakcji cech, kierując część obliczonych przez nią cech do klasyfikatora przeznaczonego dla zadania  $T_1$ , a część do klasyfikatora przeznaczonego dla zadania  $T_2$ . Wstępne eksperymenty dotyczące rozpoznawania różnych rzutów obiektów, przeprowadzone na bazie obrazów COIL-20, wskazują na skuteczność proponowanego podejścia (por. Krawiec, 2004). Na przykład dla zadań  $T_1$  i  $T_2$  z ryciny 9, zdolność predykcyjna (trafność rozpoznawania na zbiorze testującym) dla czterech zilustrowanych klas decyzyjnych jest lepsza przy zastosowaniu wspólnej procedury ekstrakcji cech niż przy osobnym rozwiązywaniu obu zadań.



Ryc. 9. Ekstrakcja cech ze współdzieleniem wiedzy między dwoma zadaniami rozpoznawania

Opisywane tu podejście jest krokiem w kierunku uczenia ciągłego (ang. *continuous learning*). W coraz częstszych zastosowaniach uczenia w robotyce, systemach kognitywnych (ang. *cognitive systems*) czy innych systemach pracujących w sposób ciągły, możliwości stosowania konwencjonalnych, „jednorazowych” technik uczenia maszynowego są ograniczone. Trudno bowiem wyobrazić sobie, aby np. robot mobilny wyposażony był w osobne podsystemy odpowiedzialne za rozpoznawanie przeszkód-ścian, przeszkód-krzesel i przeszkód-ludzi. Stąd potrzeba nowych metodologii, umożliwiających ciągłą adaptację systemu odkrywającego wiedzę do otaczającego go środowiska, tak aby sprostać pojawiającym się nowym zadaniom uczenia lub zadaniom, których charakterystyka zmienia się w czasie.

### Sztuczne życie, czyli odkrywanie wiedzy przez sztucznych agentów

Jeśli połączyć odkrywanie wiedzy z działaniem sztucznego agenta w środowisku, z którego dane pochodzą, to zamyka się pętla „środowisko-informacja-agent-czynność-środowisko”, która stwarza warunki do ewolucji tzw. *sztucznego życia* (ang. *artificial life*).

Ostatnio coraz powszechniejszy staje się pogląd, że inteligencja maszynowa, rozumiana jako odkrywanie wiedzy, nie jest procesem abstrakcyjnym, lecz wymaga osadzenia w środowisku (ang. *embedding, situatedness*) i „ucieleśnienia” (wcielenia, ang. *embodiment*). Ucieleśniony intelekt staje się agentem rozumianym jako „żywy” organizm, istniejący albo rzeczywiście, jako fizyczny robot, albo wirtualnie, jako robot symulowany w pamięci komputera (Dellaert i Beer, 1996). W wyniku ucieleśnienia powstaje oddziaływanie agent-receptory-środowisko, wymagające efektywnej koordynacji pomiędzy odbieraniem bodźców i działaniem (ryc. 10). Dzięki ucieleśnieniu agent może dokonywać aktywnej percepcji, zmieniając sposób postrzegania obiektów, docierając do dodatkowych danych lub nawet obserwując swoje zachowanie i analizując swoją interakcję ze środowiskiem.

Badaniem tak określonych agentów zajmuje się właśnie dziedzina sztucznego życia. Dziedzina ta zajmuje się w ogólności budowaniem modeli życia istniejącego na Ziemi oraz życia, jakie mogłoby istnieć w innych warunkach. Przedmiotem badań są również procesy rozwoju życia, uczenia, inteligencji, ewolucji, oddziaływań i zachowania systemów złożonych. W tym zakresie mieszczą się także eksperymenty dotyczące realistycznych agentów.

Kiedy agent (program, robot, organizm) działa w pewnym środowisku, jego zachowanie zależy od dynamiki agent-środowisko oraz wzajemnych zależności i interakcji. Do odczuwania środowiska służą zmysły i receptory (w terminologii technicznej – czujniki, sensory), a do wpływania na nie – efekторы (mięśnie, silniki, siłowniki). Inteligencja i system sterowania agenta mogą być przystosowane do sposobu odbierania przez niego sygnałów ze środowiska i do specyfiki jego efektorów. Ma więc miejsce przepływ informacji ze środowiska do agenta oraz przepływ akcji (działań) od agenta do jego otoczenia.

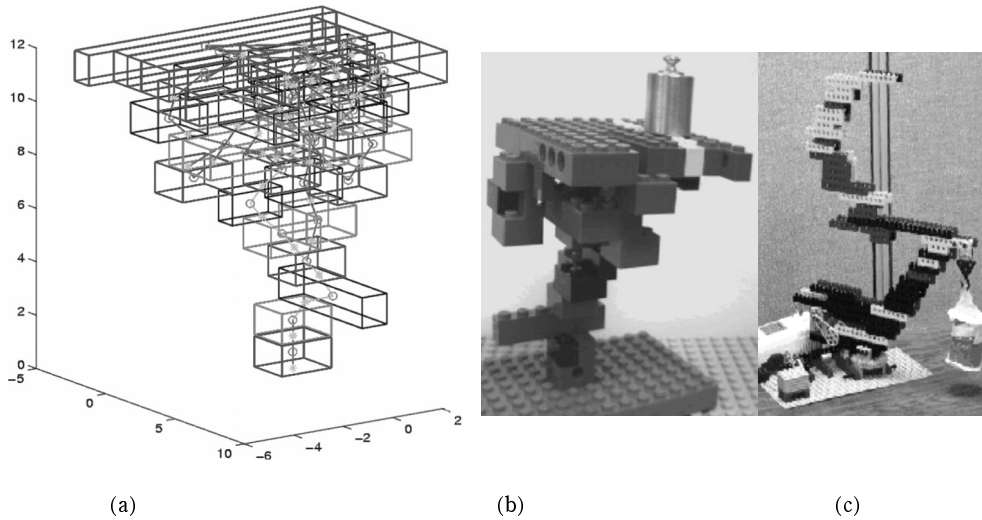
Naturalne jest oczekiwanie, by taki kompletny agent był optymalizowany automatycznie przy użyciu odpowiednich algorytmów (Bentley i Kumar, 1999). Optymalizacja oznacza tu pewną ewolucję agenta, mającą na celu jak najlepsze dopasowanie go do środowiska. Zatem nie człowiek miałby być projektantem agenta, lecz program komputerowy, który dokonywałby jego optymalizacji.

W optymalizacji agentów stosowano najpierw podejścia rozłączne, polegające na optymalizacji jednej tylko części agenta – statycznej budowy fizycznej albo części sterującej. Przykładem takiego podejścia jest optymalizacja budowy mostu, drabiny, stołu czy dźwigu (ryc. 10), a także kół zamachowych i geometrii samolotów (Bentley, 1999). W aspekcie techniki mówi się tu o budowie, wyglądzie, kształcie (ang. *design*), a w biologii – o morfologii lub ciele (ang. *morphology, body*).

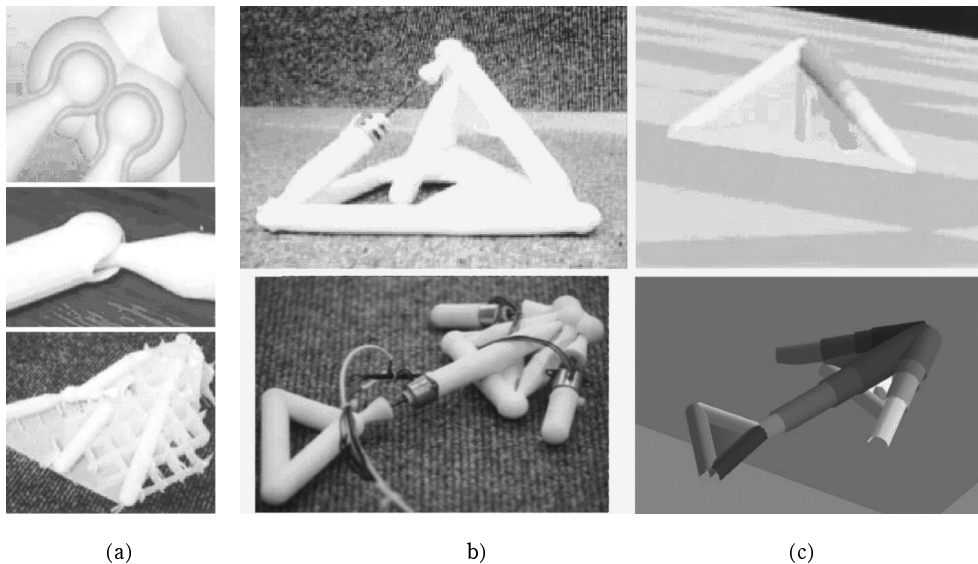
Z kolei optymalizacja samego sterowania polega na jego modyfikowaniu w celu osiągnięcia zamierzonego zachowania (np. trafności rozpoznawania); cel taki realizują na przykład metody odkrywania wiedzy scharakteryzowane w poprzednich paragrafach. Przy optymalizacji złożonych agentów techniki te okazują się mało przydatne, ponieważ stawiane zadania są zbyt złożone, ocena jest bardzo opóźniona, a od systemu oczekuje się dynamicznego, ciągłego sterowania. Najczęściej nie można stwierdzić, które części systemu są odpowiedzialne za sukces lub porażkę agenta, można jedynie ocenić go jako całość.

W przyrodzie dobór dotyczy jednak całych organizmów i całych genotypów. Dopasowanie osobnika do środowiska jest wynikiem jego postępowania i chociaż „decyzje” podejmuje jego układ nerwowy, to za oddziaływanie ze środowiskiem odpowiedzialne jest ciało. Samo istnienie ciała i mózgu nie wystarczy; niezwykle ważne są połączenia pomiędzy nimi – pozwalające na odbieranie bodźców przez mózg oraz na wykonywanie

jego poleceń przez ciało. Dopiero domknięcie pętli „środowisko-informacja-agent-czynność-środowisko” pozwala na aktywne zachowanie się osobnika w jego otoczeniu i wykorzystuje zalety jego ciała i mózgu.



Rys.10. Optymalizacja konstrukcji zbudowanej z klocków LEGO; (a) model symulacyjny, (b) prawdziwa konstrukcja stołu zbudowana na podstawie symulacji, (c) dźwig powstały w wyniku optymalizacji



Ryc. 11. (a) Wytwarzanie połączeń i konstrukcja podtrzymująca po zakończeniu produkcji, (b) realne roboty i (c) ich wzorce w symulacji

W przypadku optymalizacji budowy (czyli „ciała”) agentów aktywnych (wyposażonych w sterowanie, czyli „mózg”), „ciało” oddziałujące ze środowiskiem jest również odpowiedzialne za zachowanie agenta oraz gromadzenie informacji. Ten fakt był często niedoceniany. Tymczasem obecnie uważa się, że również ciało przetwarza informacje, a odpowiednia budowa agenta (kształt i materiał, z którego jest zbudowany) może znacznie uprościć jego sterowanie. Pożądana jest zatem równoczesna optymalizacja zarówno konstrukcji materialnej, jak i systemu sterującego, czyli ciała i mózgu (por. Sims, 1994; Funes i Pollack, 1998).

Całościowe zadanie optymalizacji realistycznych agentów czy to do celów robotyki, czy biologii, jest trudne ze względu na:

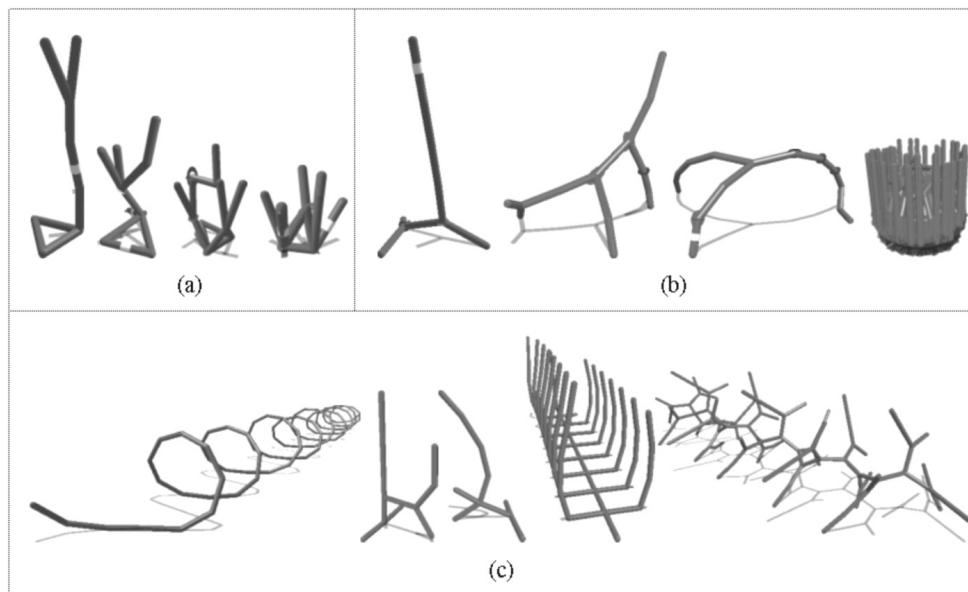
- nieskończoną liczbę rozwiązań,
- wielorakość ograniczeń dyskretno-ciągłych,
- trudną do określenia strukturę sąsiedztwa rozwiązań w przestrzeni rozwiązań,
- trudny wybór reprezentacji rozwiązania,
- bardzo silne zależności i skomplikowane powiązania między częściami rozwiązania,
- dużą liczbę optimów lokalnych funkcji oceny,
- niedeterminizm i złożoność oceny,
- wielokryterialną ocenę rozwiązań, złożoną definicję kryteriów oraz efekt opóźnienia oceny względem działania.

Wymienione trudności powodują, że zadanie optymalizacji realistycznych agentów jest jednym z najtrudniejszych zadań optymalizacji. Aby je badać, potrzebny jest odpowiedni model symulacyjny, który pozwoli na ocenę agentów w wirtualnym środowisku i ich optymalizację. W tym celu zaproponowano model umożliwiający symulację trójwymiarowych agentów zbudowanych z połączonych punktów materialnych i sterowanych sieciami neuronowymi (Komosiński i Ulatowski, 1997; Komosiński i Rotaru-Varga, 2000). Zastosowano dwa rodzaje receptorów: dotyku oraz równowagi, a także efekторы naśladujące pracę mięśni (zginających i skręcających).

Ostatnie osiągnięcia technologiczne pokazały, że model ten jest bliski praktyce: pojawiła się możliwość konstruowania realnych robotów wysoce przypominających te symulowane przez nas w wirtualnym środowisku (Lipson i Pollack, 2000). Roboty takie były automatycznie wytwarzane za pomocą termicznej obróbki plastiku. Udział człowieka ograniczał się do włożenia efektorów (siłowników) oraz wysłania programu sterującego do mikrokontrolera; tymczasowa konstrukcja wspierająca (ryc. 11a na dole) odpadała przy pierwszych ruchach robota.

Trudność zadania optymalizacji realistycznych agentów spowodowana jest przede wszystkim przez wielką liczbę możliwych rozwiązań w wielowymiarowej przestrzeni zmiennych. Aby nadać tej przestrzeni strukturę, potrzebne jest przyjęcie pewnej reprezentacji (opisu) agentów (por. Dellaert i Beer, 1996). Z kolei, by algorytm optymalizacji

mógł „poruszać się” w tej przestrzeni, wymagane są operatory, które będą modyfikowały agentów. Różne reprezentacje i różne operatory powodują, że algorytm optymalizacji zachowuje się w odmienny sposób i odkrywa zróżnicowane rozwiązania (Stanley i Miikkulainen, 2003). Na rycinie 12 ilustrują to konstrukcje otrzymane w wyniku maksymalizacji wysokości środka ciężkości agentów (Komosiński i Rotaru-Varga, 2001).

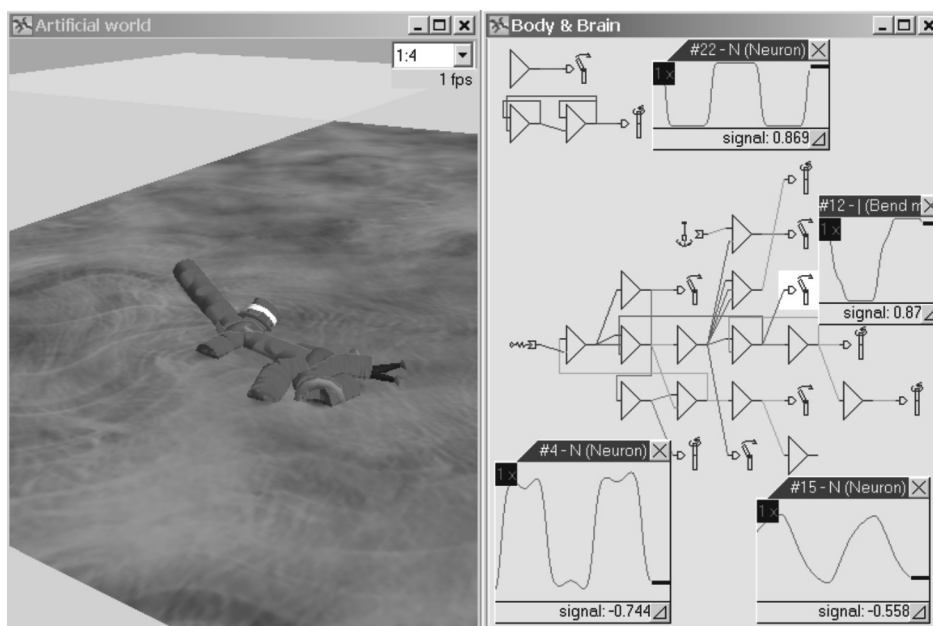


Ryc. 12. Wybrane najlepsze rozwiązania dla zadania maksymalizacji wysokości agentów;  
 (a) reprezentacja bezpośrednia niskiego poziomu, (b) reprezentacja rekurencyjna,  
 (c) reprezentacja kodująca proces rozwoju

Widoczne jest tu zróżnicowanie postaci otrzymanych agentów. W rozwiązaniach przejawia się charakter przyjętej reprezentacji: rozwiązania uzyskane przy użyciu reprezentacji niskiego poziomu nie posiadają struktury ani porządku (ryc. 12a), a reprezentacje ograniczone do struktur drzewiastych ciała dają przejrzystą budowę agenta (ryc. 12b,c). W reprezentacji modelującej rozwój widoczne są segmenty, modularność i symetria (konstrukcje spiralne, łańcuchowe i segmentowe, przypominające szkielet – ryc. 12c).

Do optymalizacji stosuje się proces ewolucyjny wzorowany na ewolucji naturalnej (Komosiński i in. 2001). Proces ten prowadzi często do uzyskania rozwiązań o pewnym stopniu przypadkowości i nadmiarowości. Pomimo to otrzymane rozwiązania były perfekcyjne w tym sensie, że przejawiały bardzo wysoki stopień zintegrowania oraz liczne, nierzadko zaskakujące i niezwykle silne związki pomiędzy elementami składowymi. Często nadmiarowość była pozorna, a próby ręcznego ulepszenia agentów koń-

czyły się pogorszeniem ich jakości; mnogość i siła wzajemnych relacji były nieodróżnialne przy pobieżnej obserwacji struktury i zachowania agenta. Tym samym potwierdzony został związek pomiędzy budową fizyczną agenta i jego systemem sterowania (ryc. 13). Części te współgrają ze sobą i nie można powiedzieć, by którakolwiek z nich pełniła pierwszorzędą rolę.



Ryc.13. Agent działający w wirtualnym środowisku; po lewej: „ciało”; po prawej: „mózg” (sieć neuronowa); pokazano niektóre przebiegi sygnałów wyjściowych neuronów; wyjścia większości neuronów sterują mięśniami zginającymi lub obracającymi; dwa wejścia podłączone są do receptorów: równowagi i dotyku

Otrzymane w wyniku optymalizacji rozwiązania nie są na tyle celowe, by przypominały przemyślane i starannie zaprojektowane twory ludzkie, jednak w sensie wartości funkcji oceny są zadowalające. Próby skonstruowania lepszych rozwiązań przez ludzi-ekspertów są trudne ze względu na wysoką czasochłonność związaną z poznaniem dynamiki konstrukcji, charakterystyki pracy sieci neuronowych i efektorów oraz ręcznym budowaniem agentów. Twory człowieka są raczej interesujące jakościowo: przejawia się w nich przejrzysta celowość, elegancja i prostota (minimum środków), często symetria i modularność. Cechy te są w opozycji do właściwości wytworów ewolucji, gdzie znajdziemy ukrytą celowość, złożoność i trudno zauważalne, silne związki oraz nadmiarowość i przypadkowość.

## Zakończenie

W artykule tym staraliśmy się naświetlić inną rolę technicznych systemów informatyki, niż tę, która wynika z postępu technologii – nie epatuje ona mocą obliczeniową, pojemnością baz danych i przepustowością połączeń sieciowych, lecz koncentruje się na przetwarzaniu rozrastających się zbiorów danych w wiedzę, czyli w prawdziwe, nietrywialne, potencjalnie użyteczne i bezpośrednio zrozumiałe wzorce ukryte w danych. Rola ta nazywa się odkrywaniem wiedzy z danych na drodze uczenia maszynowego, czyli przez indukcję. Odkryte wzorce służą z jednej strony wyjaśnianiu sytuacji decyzyjnych opisanych przez dane, a z drugiej, zalecaniu dobrych decyzji w przyszłych sytuacjach decyzyjnych. W zależności od typu danych, zarówno wzorce, jak i metody ich odkrywania mają swoją specyfikę, dlatego podzieliliśmy ten przegląd na odkrywanie wiedzy z danych ilościowych i jakościowych, tekstowych i obrazowych. Na zakończenie poszerzyliśmy rolę informatycznych systemów odkrywania wiedzy o oddziaływanie w zamkniętej pętli ze środowiskiem, z którego dane pochodzą. Takie systemy mogą służyć do symulacji życia sztucznych agentów.

Odkrywanie wiedzy z danych i symulacja życia należą do aktywnych funkcji technicznych systemów informatyki, które polegają na maszynowym uczeniu się i adaptacji do preferencji użytkownika tych systemów. Funkcja aktywna uzupełnia funkcję bierną polegającą na gromadzeniu danych, organizowaniu dostępu do baz danych i przesyłaniu ich w sieci. Nie ulega wątpliwości, że w informatyce proporcje między funkcją bierną a czynną będą musiały przechylać się bardziej zdecydowanie na korzyść tej drugiej, bowiem odkrywanie wiedzy z danych jest niezbędne dla zrozumienia świata, a także dla ewolucji i adaptacji sztucznych agentów, z których człowiek chce mieć wyrękę.

## Literatura

- Aloimonos, Y. (1993), *Active Perceptron*. Lawrence Erlbaum Associates, Hillsdale.
- Bentley, P. (1999), *Evolutionary design by computers*. Morgan Kaufmann, San Francisco.
- Bentley, P., Kumar, S. (1999), Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem. [w]: Proc. of the *Genetic and Evolutionary Computation Conference (GECCO '99)*, July 14-17, 1999, Orlando, Florida, pp.35-43.
- Brin, S., Page, L. (1998), The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30 107-117.
- Dellaert, F., Beer, R.D. (1996), A developmental model for the evolution of complete autonomous agents. [w]: P. Maes, M. Mataric, J.-A. Meyer, J.B. Pollack, S.W. Wilson (eds.), *From Animals to Animats*. Proc. of the 4<sup>th</sup> Int. Conference on *Simulation of Adaptive Behavior (SAB '96)*. MIT Press, Cambridge, MA, pp. 393-401.
- Dembczyński, K., Greco, S., Matarazzo, B., Słowiński, R. (2002), Methodology of rough-set-based classification and sorting with hierarchical structure of attributes and criteria. *Control & Cybernetics*, 31 (2002) 891-920.



- Denning, P. i in. (1998), Computing as a discipline. *Communications of the ACM*, 32, 9-23.
- Funes, P., Pollack, J.B. (1998), Evolutionary body building: adaptive physical designs for robots, *Artificial Life*, 4 (4) 337-357.
- Giove, S., Greco, S., Matarazzo, B., Słowiński, R. (2002), Variable consistency monotonic decision trees. [w]: J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (eds.), *Rough Sets and Current Trends in Computing*. LNAI 2475, Springer-Verlag, Berlin, pp. 247-254.
- Greco, S., Matarazzo, B., Słowiński, R., Tsoukias, A. (1998), Exploitation of a rough approximation of the outranking relation in multicriteria choice and ranking. [w]: T.J. Stewart, R.C. van den Honert (eds.), *Trends in Multicriteria Decision Making*. LNEMS 465, Springer-Verlag, Berlin, pp. 45-60.
- Greco, S., Matarazzo, B., Słowiński, R. (1999), The use of rough sets and fuzzy sets in MCDM. Chapter 14 [w]: T.Gal, T.Stewart, T.Hanne (eds.), *Advances in Multiple Criteria Decision Making*, Kluwer Academic Publishers, Boston, pp. 14.1-14.59.
- Greco, S., Matarazzo, B., Słowiński, R. (2000), Dealing with missing data in rough set analysis of multi-attribute and multi-criteria decision problems. [w]: S.H. Zanakis, G. Doukidis, C. Zopounidis (eds.), *Decision Making: Recent Developments and Worldwide Applications*. Kluwer Academic Publishers, Boston, 2000, pp. 295-316.
- Greco, S., Matarazzo, B., Słowiński, R. (2000a), Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. [w]: J. Fodor, B. De Baets and P. Perny (eds.), *Preferences and Decisions under Incomplete Knowledge*. Physica-Verlag, Heidelberg, 2000, pp.131-151.
- Greco, S., Matarazzo, B., Słowiński, R. (2001), Rough sets theory for multicriteria decision analysis. *European J. of Operational Research*, 129 (1) 1-47.
- Greco, S., Matarazzo, B., Słowiński, R. (2001a), Conjoint measurement and rough set approach for multicriteria sorting problems in presence of ordinal criteria. [w]: A.Colorni, M.Paruccini, B.Roy (eds.), *A-MCDA: Aide Multi Critère à la Décision – Multiple Criteria Decision Aiding*. European Commission Report EUR 19808 EN, Joint Research Centre, Ispra, pp. 117-144.
- Greco, S., Matarazzo, B., Słowiński, R. (2001b), Rough set approach to decisions under risk. [w]: W.Ziarko, Y.Yao (eds.): *Rough Sets and Current Trends in Computing*. LNAI 2005, Springer-Verlag, Berlin, 2001, pp. 160-169.
- Greco, S., Matarazzo, B., Słowiński, R. (2002), Preference representation by means of conjoint measurement and decision rule model, [w]: D.Bouyssou, E.Jacquet-Lagrèze, P.Perny, R.Słowiński, D.Vanderpooten, Ph.Vincke (eds.), *Aiding Decisions with Multiple Criteria – Essays in Honor of Bernard Roy*, Kluwer Academic Publishers, Boston, pp. 263-313.
- Greco, S., Inuiguchi, M., Słowiński, R. (2002a), Dominance-based rough set approach using possibility and necessity measures. [w]: J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (eds.), *Rough Sets and Current Trends in Computing*. LNAI 2475, Springer-Verlag, Berlin, pp. 85-92.
- Greco, S., Matarazzo, B., Słowiński, R., Stefanowski, J. (2002b), Mining association rules in preference-ordered data. [w]: M.-S. Hacid, Z.W. Raś, D.A. Zighed, Y. Kodratoff (eds.), *Foundations of Intelligent Systems*. LNAI 2366, Springer-Verlag, Berlin, pp. 442-450.
- Greco, S., Matarazzo, B., Słowiński, R. (2004), Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European J. of Operational Research*, 158 (2) 269-290.
- Greco, S., Inuiguchi, M., Słowiński, R. (2004a) A new proposal for fuzzy rough approximations

- and gradual decision rule representation. [w]: D. Dubois, J. Grzymala-Busse, M. Inuiguchi, L. Polkowski (eds.), *Fuzzy Rough Sets: Fuzzy and Rough and Fuzzy along Rough*. LNCS 3135, Springer-Verlag, Berlin, pp. 332-355.
- Hearst, M.A., Pedersen, J.O. (1996), Reexamining the cluster hypothesis: Scatter/gather on retrieval results. [w]: Proc. 19<sup>th</sup> ACM Int. Conference on *Research and Development in Information Retrieval (SIGIR-96)*, Zurich, pp. 76-84.
- Jelonek, J., Krawiec, K., Słowiński, R., Szymaś J. (1999), Intelligent decision support in patomorphology. *Polish Journal of Patology*, 50 (2) 115-118.
- Kohavi, R., John, G.H. (1997), Wrappers for feature subset selection. *Artificial Intelligence*, 2, 273-324.
- Komosiński, M., Koczyk, G., Kubiak, M. (2001), On estimating similarity of artificial and real organisms. *Theory in Biosciences*, 120, 271-286.
- Komosiński, M., Rotaru-Varga, A. (2000), From directed to open-ended evolution in a complex simulation model. [w]: M.A. Bedau, J.S. McCaskill, N.H. Packard, S. Rasmussen (eds.), *Artificial Life VII*. MIT Press, Cambridge, MA, pp. 293-299.
- Komosiński, M., Rotaru-Varga, A. (2001), Comparison of different genotype encodings for simulated 3D agents. *Artificial Life*, 7 (4) 395-418.
- Komosiński, M., Ulatowski, Sz. (1997), *Framsticks* Web Site, <http://www.frams.alife.pl>
- Krawiec, K. (2001), Pairwise comparison of hypotheses in evolutionary learning. [w]: *Proc. 18th Int. Conference on Machine Learning*, Williamstown, MA, Morgan Kaufmann, San Francisco, pp.293-301.
- Krawiec, K. (2004), Cross-task knowledge sharing in evolutionary feature construction applied to visual learning. [w]: Proc. of the 15<sup>th</sup> European Conference on *Machine Learning (ECML-2004)*, LNCS, Springer-Verlag, Berlin (w druku).
- Krawiec, K. (2004): *Evolutionary Feature Programming: Cooperative Learning for Knowledge Discovery and Computer Vision*. Monografia habilitacyjna. Wydawnictwo Politechniki Poznańskiej, Poznań (w druku).
- Krawiec, K., Bhanu, B. (2003), Visual learning by evolutionary feature synthesis. [w]: T. Fawcett, N. Mishra (eds.), Proc. of the 20<sup>th</sup> Int. Conference on *Machine Learning (ICML 2003)*. AAAI Press, Menlo Park, CA, pp. 376-383.
- Krawiec, K., Bhanu, B. (2003a), Coevolution and linear genetic programming for visual learning. [w]: E. Cantú-Paz et al. (eds.), Proc. of the *Genetic and Evolutionary Computation Conference (GECCO 2003)*. LNCS 2723, Springer-Verlag, New York, pp. 332-343.
- Lipson, H., Pollack, J.B. (2000), Automatic design and manufacture of robotic lifeforms. *Nature*, 406 (6799) 974-978.
- Masłowska, I., Słowiński, R. (2003), Hierarchical clustering of text corpora using suffix trees. [w]: M. Kłopotek, S. Wierzchoń, K. Trojanowski (eds.), *Intelligent Information Processing and Web Mining*, Springer-Verlag, Berlin, pp.179-188.
- Michalewicz, Z. (1996): *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. WNT, Warszawa, 1996.
- Nordin, P., Banzhaf, W., Francone, F. (1999), Efficient evolution of machine code for CISC architectures using blocks and homologous crossover. [w]: L. Spector, W. Langdon, U. O'Reilly, P. Angeline (eds.), *Advances in Genetic Programming III*. MIT Press, Cambridge, MA, pp. 275 - 299.
- Osiński, S., Weiss, D. (2004), Lingo – concept-driven algorithm for clustering search results. *IEEE Intelligent Systems* (w druku).

- Pawlak, Z. (1991), *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht.
- Pawlak, Z., Słowiński, R. (2004), Zbiory przybliżone we wspomaganii decyzji. [W]: Hryniewicz, O., Kacprzyk, J., Kulczyki, P. (red.), *Współczesna analiza systemowa*. PWN, Warszawa (w druku).
- Popper, K.R. (1974), *Logika odkrycia naukowego*. PWN, Warszawa 1977.
- Potter, M.A., De Jong, K.A. (2000), Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8 (1) 1-29.
- Russel, S.J, Norvig, P. (1995), *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.
- Sims, K. (1994), Evolving 3D morphology and behavior by competition. [w]: R.A. Brooks, P. Maes (eds.), Proc. of the 4<sup>th</sup> Int. Conference on *Artificial Life*. MIT Press, Boston, MA., pp. 28-39.
- Słowiński, R. (ed.) (1992), *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Dordrecht.
- Słowiński, R. (1995), Rough set approach to decision analysis, *AI Expert Magazine*, 10 (3) 18-25.
- Słowiński, R. (ed.) (1998), *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, The Handbooks of Fuzzy Sets Series, Kluwer Academic Publishers, Boston.
- Słowiński, R. (2003), Od sztucznej inteligencji do sztucznego życia, czyli o aktywnej funkcji Informatyki. *Pro Dialog*, 16, 51-76.
- Słowiński, R., Greco, S., Matarazzo, B. (2002), Axiomatization of utility, outranking and decision-rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle. *Control and Cybernetics*, 31 (4) 1005-1035.
- Słowiński, R., Greco, S., Matarazzo, B. (2002a), Rough set analysis of preference-ordered data. [w]: J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (eds.), *Rough Sets and Current Trends in Computing*. LNAI 2475, Springer-Verlag, Berlin, pp. 44-59.
- Słowiński, R., Greco, S., Matarazzo, B. (2002b), Mining decision-rule preference model from rough approximation of preference relation. [w]: Proc. 26<sup>th</sup> IEEE Annual Int. Conference on *Computer Software & Applications (COMPSAC 2002)*. Oxford, England, pp. 1129-1134.
- Stanley, K., Miikkulainen, R. (2003), A taxonomy for artificial embryogeny. *Artificial Life*, 9 (2) 93-130.
- Stefanowski, J., Weiss, D. (2003), Carrot<sup>2</sup> and language properties in Web search results clustering. [w]: Proc. of the 1<sup>st</sup> Int. Atlantic Conference on *Web Intelligence*. LNAI 2663, Springer-Verlag, Berlin, pp. 240-249.
- Węglarz, J. (1998), Informatyka jako dyscyplina a wizja społeczeństwa informatycznego. *Pro Dialog*, 7, 1-9.
- Węglarz, J. (2003), Zarządzanie zasobami w systemach typu grid. *Pro Dialog*, 16, 77-84.
- Węgrzyn, S. (2003), O informatyce jako dziedzinie nauki o ruchu i przetwarzaniu informacji. *Pro Dialog*, 16, 1-22.
- Wiener, N. (1961), *Cybernetics or Control and Communication in the Animal and the Machine*. Second edition, The MIT Press, Cambridge, MA.
- Zamir, O., Etzioni, O. (1999), Grouper: a dynamic clustering interface to Web search results. *Computer Networks*, 31, 1361-1374.

### **Computing systems as tools for knowledge discovery and simulation of life**

Contemporary computing systems evolve from data gathering to knowledge discovery from still growing data bases. Computing systems that learn and adapt to the needs of the users are called intelligent. While intelligent systems meet, in general, expectation of computing system users, knowledge discovery fails to produce useful results if data semantics is not taken into account. The author underlines this aspect while characterizing knowledge discovery from different kinds of data: quantitative and qualitative, texts, and pictures. The claim that intelligence is not an abstract process but an „embodied” one, is a base of specialty called artificial life. Artificial life concerns machine co-evolution of intelligence („brain”) and morphology („body”) of an agent interacting with environment through receptors and effectors. The author briefly presents a programming system permitting simulation and optimization of 3-dimensional agents build of material points connected by elastic joints and controlled by artificial neural networks. The paper ends with remarks on possible development scenarios of intelligent computing.

**Key words:** computing, knowledge discovery, artificial intelligence, web mining, decision support, artificial life