

JANUSZ KASPERKIEWICZ¹⁾

O MOŻLIWOŚCIACH ZASTOSOWAŃ METOD SZTUCZNEJ INTELIGENCJI W INŻYNIERII LĄDOWEJ²⁾

STRESZCZENIE. Omówiono możliwości stosowania metod obliczeniowych nawiązujących do trzech podstawowych rozwiązań tzw. sztucznej inteligencji. Są to sztuczne sieci neuronowe, metody uczenia się maszyn, i metody algorytmów genetycznych. W tych wszystkich przypadkach użytkownik nie musi obmyślać modelu badanego zjawiska, ponieważ system uczy się na podstawie przedstawionych przykładów, generując następnie odpowiedzi w postaci poszukiwanych wartości atrybutów, w formie prawidłowego klasyfikowania następných przykładów o tej samej, ustalonej strukturze lub w postaci formułowania reguł (uogólnień) opisujących dane zjawisko. Nieco bliżej omówiono rozwiązania sieci Fuzzy ARTMAP oraz programu AQ19.

1. WPROWADZENIE

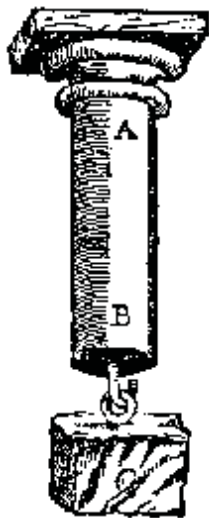
W naukach inżynierskich uogólnianie wyników obserwacji związane jest tradycyjnie z tworzeniem modeli, w szczególności także modeli ośrodków ciągłych. W ten sposób opracowane zostały rozmaite formuły analityczne, nadające się do opisu – w pewnym przybliżeniu – konkretnych materiałów takich jak stal, beton, piasek, itd.

¹⁾ prof. dr hab. inż. – Instytut Podstawowych Problemów Techniki PAN, Warszawa

²⁾ artykuł stanowi rozszerzoną wersję referatu przygotowanego na Seminarium w dn. 26 stycznia 2004, w IPPT PAN, zorganizowanego celem uczczenia 30 rocznicy śmierci ŚP. Prof. Zbigniewa Wasiutyńskiego

Modele pozwalają przewidywać zachowanie się elementów lub konstrukcji pod wpływem oddziaływań zewnętrznych, zmian wewnętrznych oraz upływu czasu. Obejmują mniej lub bardziej złożone zależności między reprezentacjami sił, naprężeń, temperatury, mas, przemieszczeń, etc., definiując w ten sposób aparat pojęciowy poszczególnych działów mechaniki ośrodków ciągłych. W efekcie prac nad modelami powstały, zwłaszcza w okresie ostatnich dwóch stuleci, formalizmy teorii sprężystości, plastyczności, lepko-sprężystości, ośrodków sypkich, cieczy i gazów, etc., [1].

Modele, o jakich mowa dotyczą pewnych bytów wirtualnych, w znaczeniu: „byty nierzeczywiste”. Obserwacje stanowiące podstawę formalnych uogólnień obejmowały sytuacje od bardzo uproszczonych, jak w przypadku Galileusza, który przyjmował, że wytrzymałość pręta jest proporcjonalna do jego przekroju poprzecznego (rys. 1), poprzez założenia Hooke’a, który postulował istnienie liniowej zależności między wartością siły i - mniej już intuicyjnie oczywistą - wielkością odkształcenia, aż po skomplikowane związki różniczkowe nowoczesnych teorii ośrodków ciągłych, np. ciał liniowo sprężystych.



Rys.1. Próba na rozciąganie wg Galileusza
Fig.1. A tensile strength test, according to Galileo

Wszelkie kreowane przez badaczy modele nie są bytami rzeczywistymi, a jedynie pewnymi reprezentacjami rozważanych w filozofii tzw. powszechników, (uniwersaliów), identyfikowanych przez podanie zestawów cech – atrybutów, wspólnych dla pewnych zgodnie przez obserwatorów utożsamianych pojęć. Termin „zgodnie” dotyczy tu intencji utożsamiania, bynajmniej nie zaś zgody, co do konkretnego doboru cech i przypisywanych im znaczeń.

Z chwilą, gdy badacz wyobrazi sobie, co oznacza ośrodek sprężysty, plastyczny, kruchy, czy jakikolwiek inny, opis danego modelu jest zakończony, został on jakby „zamrożony”. Model jest zatem w tym momencie ustalony z dokładnością do danych

liczbowych, którymi są wartości parametrów danego modelu. Owe parametry - charakterystyki materiałowe - mogą mieć rozmaite wartości liczbowe, natomiast sposób globalnego zachowania się danego materiału został już przesądzony.

Założenia modelu stanowią zatem wyraźne ograniczenie opisów danego ciała, zjawiska, konstrukcji, itd. Jest odrębną kwestią na ile owe ograniczenia są uświadamiane przez inżynierów posługujących się danym opisem. Np. często nie uświadamiamy sobie, że w założeniach liniowej teorii sprężystości nie występuje granica sprężystości czy wytrzymałość ośrodka. Upraszczając sytuację do modelu jednowymiarowego można wyobrazić sobie, że związek między naprężeniem i odkształceniem pozostaje liniowy w całym dającym się pomyśleć zakresie zmienności, od naprężeń ujemnych na dowolnym poziomie do dowolnie wysokich naprężeń dodatnich. W tej sytuacji teoria sprężystości opisuje pewien nierealny świat materiałów „nieskończenie wytrzymałych”. Występująca tu sprzeczność z obserwacjami świata rzeczywistego może być w licznych sytuacjach pominięta, co następuje w efekcie przyjęcia ukrytego (a często i nieświadomego!) założenia, że rozważane są tylko „małe odkształcenia”.

Rozwijając mechanikę ośrodków ciągłych badacze tworzyli modele (czyli materiały wirtualne) również takie, których oczywistość dla ludzkiej intuicji bywała przynajmniej dyskusyjna, jak np. w wypadku teorii sprężystości mikropolarnej Cosseratów. Prace analityczne kontynuowano jednak nawet i w tych sytuacjach, a motywacją mogły być np. potrzeby estetyczne matematyka. Później wyniki mogły okazać się zresztą przydatne, chociaż w innym zakresie niż pierwotnie podjęte poszukiwania.

Najczęściej stwierdzano jednak rozbieżności zachowania się kolejnych modeli z rzeczywistością i proponowano nowe modele. To jak model pozwala opisywać rzeczywistość mogło zresztą nie być łatwe do stwierdzenia, ponieważ im bardziej złożone są opisy, tym mniej bezpośrednio są eksperymenty pozwalające określić stałe materiałowe na podstawie doświadczenia.

Bardziej skomplikowane opisy modelowych materiałów związane są z kłopotami rachunkowymi przy przewidywaniu zachowania się całych konstrukcji. Historycznie rzecz biorąc modyfikacje opisów polegały najpierw na upraszczaniu notacji, np. w teorii sprężystości przejście od bardzo niewygodnych oznaczeń naprężeń i odkształceń w podręcznikach Love'a, poprzez indeksową notację Einsteina, aż do nowoczesnej notacji absolutnej, a potem przez usprawnianie obliczeń w postaci rozmaitych metod numerycznych, przede wszystkim metod elementów skończonych.

Nasuwa się pytanie czy posługiwanie się takimi szczegółowo określonymi modelami jest wyłącznym sposobem przewidywania zachowania się konstrukcji czy materiału. Otóż nie jest to rozwiązanie jedyne. Profesor Zbigniew Wasiutyński propagował w latach sześćdziesiątych epistemologiczne hasło: „Jedynym prawdziwym modelem odkształcalności ciała stałego jest to ciało; nie wynajdujcie innych”, [2]. Sens tego wezwania ujawnia się, jeśli termin ciało odnieść nie do indywidualnego przypadku konkretnej próbki czy elementu, ale do odpowiadającego mu pojęcia ogólnego, w pewnej

zbiorowości materiałów, tworzyw, elementów konstrukcyjnych, który człowiek – obserwator skłonny jest klasyfikować do dającej się zidentyfikować kategorii. W szczególności może to być np. rodzina elementów betonowych, wykonanych przy zastosowaniu ściśle określonych składników, przy zadanym reżymie technologicznym, w określonym wieku, itd., eksploatowanych w wybranym stanie naprężenia.

Przez akt zaklasyfikowania obserwator wyróżnia pewną kategorię materiałów (konstrukcji, rozwiązań, etc.), po czym może podejmować wysiłki w kierunku uzyskania praktycznie przydatnych wniosków na jej temat – wniosków ilościowych lub jakościowych. Np. przewidywania wytrzymałości w danym stanie naprężenia, oceny trwałości materiału przy znanych oddziaływaniach, a także wynajdywania reguł do projektowania i optymalizowania wybranej klasy materiałów.

Wszelkie dostępne informacje na temat zachowania się rozumianego w sposób powyżej opisany ciała składają się na bazę wiedzy na jego temat. Wiedza odpowiada rozpoznaniu, jakie reakcje – odpowiedzi - odpowiadają otrzymanywanym bodźcom (sygnałom). Podstawowy zasób ludzkiej wiedzy (nie tylko inżynierskiej) jest właśnie z takim rozpoznawaniem związany.

Wraz z epoką rozwoju komputerów pojawiła się możliwość automatyzacji przetwarzania i eksploatacji wiedzy dostępnej w doświadczeniu. Upowszechnienie maszyn cyfrowych udostępniło narzędzia w sensie urządzeń elektronicznych (*hardware*), a równie ważnym elementem postępu stał się rozwój informatyki, w ramach której powstały całe generacje odpowiedniego oprogramowania (*software*).

Specyficzny typ oprogramowania jest rozwijany na całym świecie jako metody sztucznej inteligencji, (SI). Określenie „sztuczna inteligencja” jest zdaniem piszącego te słowa sformułowaniem niefortunnym i stanowi właściwie sprzeczność, (tzw. *oksymoron* – określenie wewnętrznie sprzeczne, takie jak np.: „sucha woda”). Lepszymi określeniami są, zapewne, znacznie mniej efektowne - „miękkie metody obliczeniowe” lub „metody komputacyjne”. Inteligencja jest cechą jedynie podmiotów żywych, natomiast nie wydaje się, aby mogła stanowić cechy programu, dającego się zapisać na kartkach papieru. Realizacja programu komputerowego może wywoływać w ludzkim obserwatorze złudzenie kontaktu z bytem rozumnym i suwerennym, i może być nawet zaskakująca (skutkiem niedoinformowania obserwatora), jednak program – w odróżnieniu od człowieka - działa „tak jak musi”, czyli nie posiada atrybutu wolności, (wolności wyboru, chociażby błędnego). Dotykamy w ten sposób zagadnień filozoficznych, które nie są dalej rozważane. Zasygnalizowany tu problem semantyczny nie ma zresztą większego znaczenia w zastosowaniach inżynierskich, i przy podanym powyżej zastrzeżeniu wypada uznać termin „sztuczna inteligencja” (SI) za wygodny skrót myślowy.

Postęp w dziedzinie informatyki zaowocował na przestrzeni ostatniego półwiecza nie tylko rozwojem komputerów i języków programowania, ale także rozwojem takich szczególnych dziedzin jak sztuczne sieci neuronowe, metody uczenia maszyn oraz metody algorytmów genetycznych. Wymienionych trzech ujęć dotyczy tekst poniżej. Do tych samych metod można by zaliczyć także niektóre metody statystyki,

metody zbiorów rozmytych i zbiorów przybliżonych (*Fuzzy Sets*, *Rough Sets*) oraz specyficzne metody z zakresu eksploracji danych (*Data Mining*). Zagadnienia te nie są dalej opisywane.

Procedury SI są szczególnie przydatne w zagadnieniach uznawanych za nie-algorytmiczne, tzn. takich, co do których nie tylko rozwiązanie jest nieznane, ale niekiedy nie jest nawet pewne czy odpowiednie rozwiązanie istnieje. W ścisłym sensie w typowych zagadnieniach inżynierskich, (teoria konstrukcji, inżynieria materiałowa), problemy nie-algorytmiczne nie występują. Jako przykłady można natomiast wymienić zagadnienia poza inżynierskie, takie jak problemy gry w szachy, wykrywania nadużyć, rozumienia tekstu. Funkcjonalność umysłu ludzkiego pozwala radzić sobie w wielu podobnych sytuacjach, czego nie są w stanie czynić „sztywne” algorytmy obliczeniowe. Upodobnienie działania automatycznego do rozumowania ludzkiego polega na posługiwaniu się obliczeniowymi metodami określanymi jako „miękkie”, tzn. takie, w których funkcjonowanie systemu jest modyfikowane w zależności od analizowanych przez system danych.

Wzmiankowane tu wnioskowanie związane jest albo z przeszukiwaniem bardzo licznych zbiorów danych albo, przynajmniej, z wielokrotnym analizowaniem tych samych danych (tzw. kolejne epoki nauczania). W efekcie stanowi to pewne przybliżenie sposobów heurystycznego rozumowania człowieka, który zmierza w kierunku uzyskania rozwiązania „ślusznego”, chociaż niekoniecznie optymalnego. Chodzi wówczas o pewne kryterium raczej zadowalające niż optymalne, o badanie rozmaitych możliwości, bez gwarancji sukcesu, z oczywistym ryzykiem pomyłek, etc. Sztywne algorytmy z reguły nie uwzględniają podobnych możliwości.

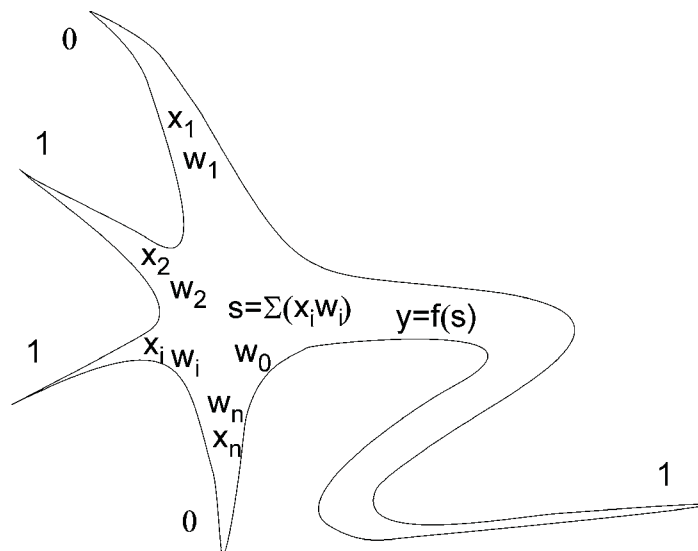
Szczegółami i implementacją metod opisywanych poniżej zajmują się liczne ośrodki badań informatycznych. Celem niniejszego tekstu jest jedynie zwrócenie uwagi środowiska inżynierii lądowej na możliwości, jakie obecnie otwiera rozwój owych metod „Sztucznej Inteligencji”.

2. SZTUCZNE SIECI NEURONOWE

Sztuczne sieci neuronowe (SSNy) są układami służącymi do przekształcania danych liczbowych w sposób, który jest wzorowany na funkcjonowaniu komórek nerwowych w mózgu; stąd zresztą metafora przyjęta w ich nazwie. Otwierają one interesujące możliwości obliczeniowe, a w Polsce inżynierskimi zastosowaniami SSNów najbardziej zajmowano się w zespole Prof. Waszczyszyna (Politechnika Krakowska).

Na poziomie elementarnym sztuczny neuron przekształca wejściową informację w postaci łańcucha binarnego (tzn. złożonego z zer i jedynek) w wyjściową wartość, którą w tym najprostszym przypadku stanowi również albo zero albo jedynka (rys. 2). Jak wiadomo za pomocą ciągów zer i jedynek łatwo można zakodować dowolne liczby, teksty, itd. Przekształcenie danych w neuronie polega na mnożeniu wartości wejścia przez odpowiednie mnożniki, czyli tzw. wagi, zsumowaniu uzyskanych iloczynów i

normalizacji wyniku przy pomocy pewnej funkcji progu (f) tak, aby ostatecznym rezultatem była ponownie liczba ze zbioru $\{0, 1\}$. W bardziej zaawansowanych rozwiązaniach użytkownik operuje i na wejściu i na wyjściu wektorami liczb dziesiętnych.



Rys.2. Idea przetwarzania danych (binarnych) w neuronie sztucznym – elemencie sztucznej sieci neuronowej, wzorowanej na obserwacjach biologicznych, (x_i – wartości sygnałów, w_i – wartości tzw. wag, f – pewna funkcja dyskryminacyjna)
 Fig.2. Transformation of binary data in an artificial neuron, which is a basic element of artificial neural networks. The idea follows general biological observations, (x_i – signals, w_i – weights, f – a certain discrimination function)

Jeśli nauczanie przebiega w trybie pod nadzorem, (*supervised*), wówczas wartościom wejściowym – bodźcom, które wprowadzane są do obliczeń w neuronie, towarzyszą wartości wyjściowe – reakcje. W związku z tym możliwe jest takie ustawienie wartości wag (w_i), aby określonym kombinacjom zer i jedynek na wejściu towarzyszyły wartości bliskie odpowiedziom oczekiwany na wyjściu, tzn. reakjom. Dla pewnej skończonej liczby wzorców system wag da się nastawić tak, że układ neuronu obliczać będzie poprawne odpowiedzi. Liczba ta zależy od stopnia złożoności układu neuronów, tzn. od struktury sieci.

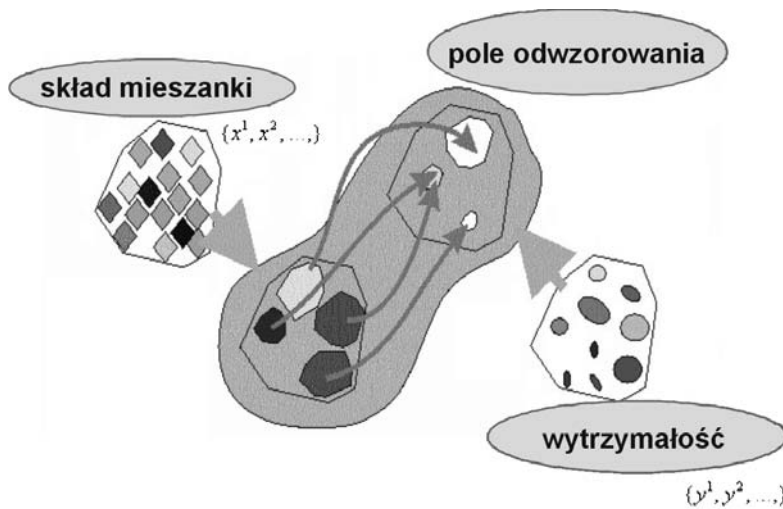
W przedstawionym tu schemacie myślowym użytkownika sztucznych sieci neuronowych ukryte jest istotne założenie, że dane treningowe, tzn. te, których dotyczy uczenie systemu, zawierają pewną „strukturę”, że pewne sytuacje (zdarzenia) powtarzają się w nich, i dane nie są przypadkowym zbiorem liczb, np. o charakterze „białego szumu”, (mieszanka sygnałów o wszelkich możliwych częstotliwościach). Jeśli

dane będą zmieniać się w sposób przypadkowy konstruowanie i eksploatacja sieci pozbawiona jest sensu.

W sztucznej sieci neuronowej dopasowanie wartości wag następuje samoczynnie, np. w formie sprzężenia zwrotnego. Można wówczas mówić, że system nauczył się rozpoznawać zaprezentowane mu wzorce. Szczegóły realizacji rozmaitych rozwiązań SSN omawiane są w licznych monografiach i podręcznikach, np. [3]. Uczenie się realizowane jest na ogół przez kolejne przepuszczanie przez system tych samych danych (tzw. epoki nauczania), dane mogą jednak występować również w formie ciągu czasowego, (np. pochodzącego z obserwacji meteorologicznych), i uczenie przebiegać będzie np. do momentu, gdy stany wewnętrzne (wagi) przestaną ulegać zmianom. Sposób przekazywania informacji o różnicy między wartością wzorca i wynikiem obliczeń (błędu) tak, żeby sieć „dostrajała się” do otrzymywanych danych, jest jednym z podstawowych zadań teorii sztucznych sieci neuronowych, i wraz z opisem topologii sieci (rozmieszczeniem i połączeniami neuronów) określa tzw. architekturę sieci.

W bardziej złożonych systemach sztucznych sieci neuronowych analizowane dane można wyobrazić sobie jako układy liczb dziesiętnych, zarówno po stronie wejścia – np. $x^i = (x_1^i, x_2^i, x_3^i, \dots, x_n^i)$, jak i wyjścia $y^j = (y_1^j, y_2^j, y_3^j, \dots, y_m^j)$ – rysunek 3, (j – identyfikuje tu kolejne obserwacje, czyli kolejne rekordy). Nauczanie się przez sieć rozpoznawania wzorców podawanych na wejściu oznacza w praktyce odtworzenie przez układ sieci odwzorowania przestrzeni liczb rzeczywistych n wymiarowych w przestrzeń m wymiarową. Na przykład może to stanowić odwzorowanie przestrzeni składów mieszanek betonowych w przestrzeń wytrzymałości betonów wykonanych z owych mieszanek. Liczba struktur, jakie może sieć rozpoznać uwarunkowana jest zarówno charakterystykami wprowadzanych danych, (wartości n i m), jak i budową warstw ukrytych. Jako „ukryte” określa się warstwy składające się z neuronów wewnętrznych, do których użytkownik z reguły nie ma dostępu. Po dostosowaniu wartości wag podczas kolejnych epok nauczania sieć może formułować prognozy odnośnie do spodziewanych wartości wektora wyjścia odpowiadających podanej wartości wektora wejścia. Uczenie może dotyczyć również ustawicznie zmieniających się danych dostarczanych z jakiegoś rzeczywistego procesu, który ma charakter dynamiczny.

Sieci mogą uczyć się rozpoznawania kombinacji zestawów liczb pochodzących z pewnego zbioru oraz mogą także nauczyć się rozpoznawania odpowiednio zakodowanych sygnałów akustycznych, obrazów, itp., chociaż zakodowane rekordy mogą wyglądać zupełnie niezrozumiale dla człowieka (por. np. dalej – sygnały na rysunku 5), jednak sieć jest w stanie wykryć w nich obecność pewnych skupień, lub innych charakterystycznych rozkładów. Po wytrenowaniu (nauczeniu) sieci pozwala to na identyfikację np. dźwięków czy form geometrycznych. Zakres rozpoznawania jest jednakże zawsze ograniczony „doświadczeniem” sieci, tzn. zbiorem przykładów, jakich dostarczono sieci w fazie nauczania. Próby ekstrapolacji poza zakres zbiorów treningowych (uczących) są jednakże bardzo ryzykowne.

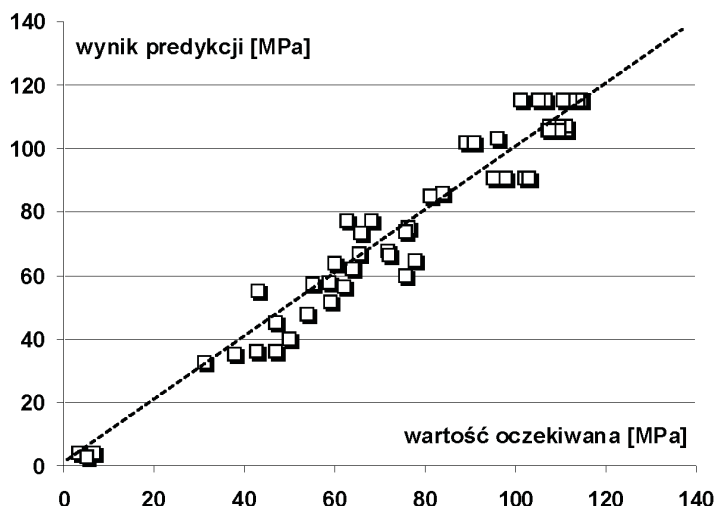


Rys.3. Koncepcja sieci Fuzzy ARTMAP. Wektorom danych wejściowych x odpowiadają wektory danych wyjściowych y
 Fig.3. The Fuzzy ARTMAP network concept. To the input data vectors x correspond output data vectors y

Stosunkowo najbardziej rozpowszechnionym rozwiązaniem sztucznych sieci neuronowych jest sieć ze wsteczną propagacją błędów, BP, (*Back Propagation*). Sieci tego typu uczą się raczej powoli i mają tendencję do poprawnego odwzorowania stosunkowo gładkich odwzorowań, pozbawionych nieciągłości. Natomiast, gdy w danych występują silne niejednorodności sieci te również wtedy generują odwzorowania ciągłe, które mogą być w oczywisty sposób błędne. Inne koncepcje tzw. architektury sieci ukierunkowane są na specyficzne zagadnienia, np. tzw. sieci komórkowe przydatne w zakresie rozpoznawania obrazów.

Korzystne doświadczenia w przypadku silnie niejednorodnych danych z zakresu technologii betonu uzyskano stosując sieć Fuzzy ARTMAP, [4]. W przypadku tego rozwiązania w trybie uczenia pod nadzorem współpracują ze sobą dwa układy oparte na koncepcji ART, (*Adaptive Resonance Theory*). Odmienne niż w wypadku sieci BP poszczególne układy ART są systemami pracującymi bez nadzoru, (*unsupervised*). Rekordy uczące analizowane są równolegle i niezależnie w warstwie atrybutów wejściowych (*input* – impulsy, bodźce, przyczyny) i wyjściowych (*output* – reakcje, skutki), oddzielnie klasyfikując wejściowe i wyjściowe fragmenty rekordów jako „podobne”, (na zasadzie bliskości tych rekordów wg odpowiednio określonej metryki). System Fuzzy ARTMAP rozbudowuje oddzielny blok pamięci (blok odwzorowujący) powiązań pomiędzy kategoriami wejścia i wyjścia, zapamiętując w ten sposób, jakie zestawy danych (np. o składzie betonu) odpowiadają określonym właściwościom materiału (np. określonej wytrzymałości), rysunek 3.

Eksperymenty z zaimplementowaną w środowisku Windows siecią typu Fuzzy ARTMAP obejmowały analizę bazy danych, w której skład betonu opisywany był przez 6 atrybutów (zawartości wody, cementu, kruszywa drobnego, kruszywa grubego, pyłów krzemionkowych i superplastyfikatora), a jedynym atrybutem na wyjściu była wytrzymałość betonu na ściskanie. Baza danych zestawiona została na podstawie publikacji w raportach i czasopismach technicznych. Na rysunku 4 pokazano wyniki przewidywania (predykcji) uzyskane po wytrenowaniu sieci, podczas rozpoznawania rekordów testowych, które nie występowały w zbiorze uczącym. Zade-monstrowany wynik należy uznać za korzystny, ponieważ zawiera dające się zweryfikować pozytywnie uogólnienia, uzyskane w sposób automatyczny, danych zawartych w raportach i publikacjach powstałych w laboratoriach z różnych krajów, bez przestrzegania ustalonych reżimów badań, z nieprecyzyjnymi opisami wyników, itd.



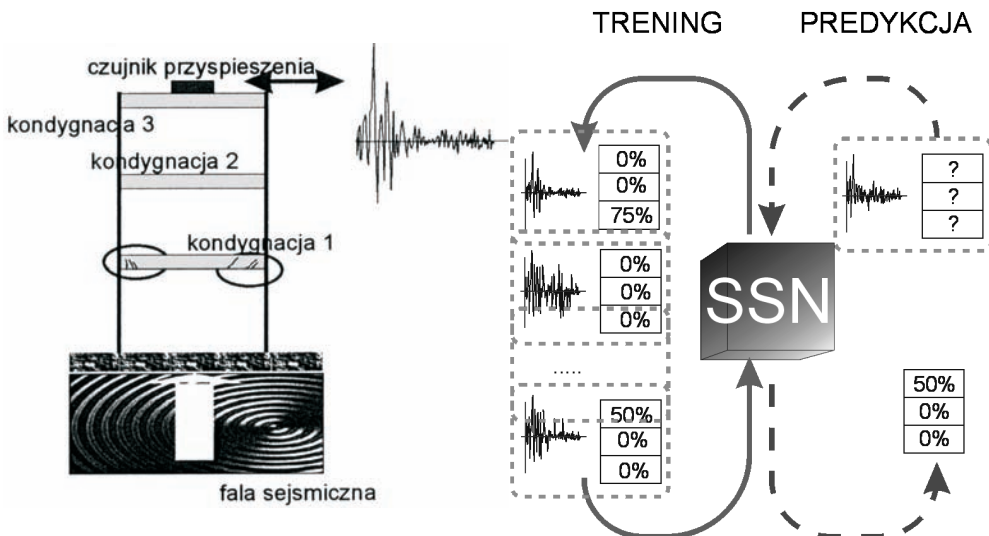
Rys. 4. Wynik przewidywania (predykcji). Program rozpoznał 50 rekordów na 156 pokazanych. Błąd rozpoznania średnio 10%. Kwadrat współczynnika korelacji wyniósł ok. 0,97

Fig.4. Results of a prediction. The program was able to recognize 50 records out of 156 records submitted for testing. The error of recognition was on average 10%. The square of correlation coefficient was about 0.97

W odróżnieniu od sieci rekurencyjnych (np. opartych na koncepcji wstecznej propagacji błędu) Fuzzy ARTMAP może nie rozpoznawać rekordów testowych, jeśli dostatecznie podobne do nich rekordy nie wystąpiły w zbiorze trenującym. Jest to ważne, aby w podobnej sytuacji sieć powstrzymała się właśnie od sugerowania jakiegokolwiek odpowiedzi, ponieważ podawanie jako odpowiedź wartości wynikających z ciągłości funkcji odpowiedzi (interpolacja) nie ma logicznego uzasadnienia. Na pokazanym na rysunku 4 wykresie, uzyskanym z pomocą specjalnie opracowanej w

IPPT PAN implementacji algorytmu Fuzzy ARTMAP, rozpoznanych zostało tylko 50 rekordów (spośród 156 rekordów testowych, które nie występowały w danych uczących), ale współczynnik korelacji między wartościami oczekiwanymi i prognozowanymi uzyskano wysoki ($r=0,984$). Warto dodać, że próby predykcji w zakresie tych samych danych przy pomocy sieci BP trwały długo i nie przyniosły powodzenia.

Przykład dobitnie pokazującym przydatność sztucznych sieci neuronowych w problemach trudnych do analizy przez człowieka został opisany w [5]. Stan uszkodzenia pewnej konstrukcji trójkondygnacyjowej oceniany miał być na podstawie wykresu zanotowanego na umieszczonym na niej rejestratorze drgań, który notował wstrząsy spowodowane falą trzęsienia ziemi, określona poprzez zmiany amplitudy w czasie, rysunku 5. Dla człowieka wszystkie takie sygnały wydają się dość podobne i trudne do odróżnienia jedne od drugich. Sieć (BP) nauczana była zestawem danych złożonych z 200 wartości wynikających z przekształcenia sygnału przy pomocy szybkiej transformacji Fouriera do domeny częstotliwości, którym towarzyszyła informacja, która z trzech kondygnacji została uszkodzona i w jakim stopniu, (możliwe wartości stanu uszkodzenia rozróżniano jedynie jako 0%, 50% lub 75%). Po wytrenowaniu sieć była w stanie dokonywać predykcji stanu konstrukcji na podstawie rozpoznania sygnału z rejestratora.



Rys.5. Ustrój ramownicowy o trzech kondygnacjach uległ uszkodzeniom skutkiem oddziaływań sejsmicznych (powstały zarysowania). Sztuczna sieć neuronowa posłużyła do identyfikacji miejsca i stopnia uszkodzenia, [5]

Fig.5. A three story frame was damaged due to earthquake, (appearance of cracks). Artificial neural network was applied to identify the member damage, [5]

Przykład ten pokazuje, że rozpoznawanie uszkodzenia może odbywać się automatycznie, bez potrzeby przyjmowania przez obserwatora jakichkolwiek założeń odnośnie do przebiegu procesu niszczenia danego ustroju.

SSNy stosowane były w najróżniejszych zagadnieniach, np. – jak powyżej - przy rozpoznawaniu stanu uszkodzenia konstrukcji na podstawie analizy sygnału akustycznego, przy obliczaniu elementów żelbetowych na drodze interpolacji pewnej rodziny rozwiązań wyznaczanych metodami elementów skończonych, przy ocenie stanu konstrukcji na podstawie uogólniania ocen ekspertów, [6].

Sztuczne sieci neuronowe funkcjonują na danych kompletnych, w których występują wszystkie wartości atrybutów, a dane są liczbami rzeczywistymi. W wypadku, gdy wartości niektórych atrybutów nie są określone, muszą być one odgadnięte i uzupełnione, albo też z odpowiednich rekordów trzeba w ogóle zrezygnować.

Gdy w bazie danych występują zmienne nominalne można podejmować próby odpowiedniego kodowania atrybutów przy pomocy liczb naturalnych, jednak nie jest to zabieg obojętny dla poprawnego działania sieci. Alternatywnym rozwiązaniem jest wówczas dodawanie tylu nowych atrybutów wejściowych o charakterze zmiennych boolowskich (wartości „tak” lub „nie”), ile różnych wartości ma dany atrybut nominalny. Jest to rozwiązanie ideowo poprawne, ale może poważnie powiększać rozmiary sieci. Dodatkowo, w wypadku gdyby chodziło o atrybuty wyjściowe, sieć musiałaby być wyposażona w odpowiedni mechanizm eliminacji dwuznaczności, ponieważ wykluczona powinna zostać np. możliwość pojawienia się jednocześnie kilku wartości „tak” (byłby to wówczas wynik sprzeczny), nowych atrybutów stanowiących rozwinięcie pojedynczego, oryginalnego atrybutu nominalnego.

3. METODY UCZENIA SIĘ MASZYN

Sztuczne sieci neuronowe posiadają swoiste ograniczenia. Natomiast ograniczenia, że analizowane bazy danych powinny zawierać wyłącznie atrybuty ilościowe, i nie mogą w nich występować ubytki lub nieokreśloności, nie ma na ogół w algorytmach z zakresu uczenia się maszyn – MUM (ang. *Machine Learning* ML). Istnieje szereg programów typu MUM, zarówno komercyjnych jak i ogólnego dostępu (*freeware* lub *shareware*), określanych często jako procedury przeszukiwania danych (penetracji danych - ang. *Data Mining*), [7]. Znakomita większość publikacji na ten temat nie dotyczy zagadnień inżynierii lądowej, [7 - 10], chyba że w odniesieniu do zagadnień organizacji transportu, [11 - 12].

Podobnie jak w innych metodach SI zasadniczym elementem metod MUM jest zdobywanie wiedzy na podstawie przykładów. Stosowanych jest najczęściej jeden z dwóch sposobów wykrywania struktur w danych: drzewa decyzyjne lub metody indukcyjne. Dotyczyć ich będzie tekst poniżej. Wzmiankowane tutaj struktury danych, składających się z rekordów, mogą być badającemu znane, i wówczas chodzi o ich identyfikację i ocenę, lub też mogą być nieznanne, i wówczas chodzi o wytypowanie

odpowiednich reguł klasyfikacji danych, które można następnie weryfikować, uogólniać lub bezpośrednio stosować.

Drzewa decyzyjne tworzone są przez stopniowy podział danych na podzbiory w zależności od wartości wyszczególnionych parametrów, które potem łączone są w jeden układ – graf drzewa. Śledząc ścieżki grafu można potem sformułować reguły odpowiadające pożądanej przez użytkownika decyzji. Metody indukcyjne rozpoczynają działanie od przypadkowego określenia pierwszej reguły, określającej poszukiwany rezultat. Celem utworzenia następnej reguły rozważane są tylko te rekordy, których ta poprzednia reguła nie obejmuje, i tak dalej, aż do wyczerpania całości danych. W obu wypadkach analiza danych wsparta jest ocenami statystycznymi.

Na końcu postępowania istotnym problemem jest interpretacja wyników i możliwości dodatkowych uproszczeń uzyskanych reguł. Uzyskane rezultaty mogą być analizowane przy pomocy tego samego programu MUM, lub korzystając z łatwo dostępnych arkuszy kalkulacyjnych (np. MS Excel), jednak zawsze sprawdzenia należy dokonywać korzystając z danych, które nie były użyte w fazie nauczania systemu.

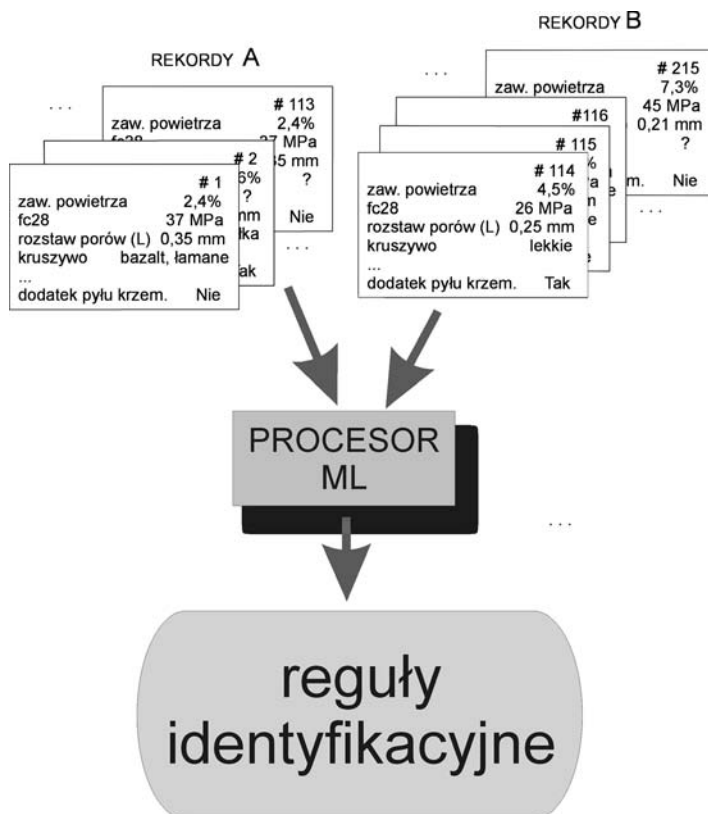
Wszystkie rekordy bazy danych muszą spełniać pewne warunki formalne, polegające na tym, że każdy atrybut może przybierać wartości jedynie ze ściśle określonej domeny. Oznacza to, że w wypadku atrybutów ciągłych (stanowiących odpowiednik liczb rzeczywistych) podać należy zakres zmienności, w wypadku atrybutów strukturalnych – trzeba określić strukturę, a w wypadku atrybutów nominalnych – zestawić listę dopuszczalnych wartości tego atrybutu, (tzw. wartości legalne). Z zachowaniem wspomnianych wymagań baza danych zapisana musi być w postaci odpowiedniego jednego lub kilku pliku tekstowych (skryptów), gdzie poza szczegółami rekordów i definicją struktury bazy sformułowane mogą być rozmaite informacje dodatkowe, takie jak hipotezy sugerowane przez użytkownika, koszty określania poszczególnych atrybutów, wymagania co do stopni ogólności lub szczególności oczekiwanych reguł, itd.

Ogólną ideę funkcjonowania systemu MUM ilustruje rysunek 6: po zaprezentowaniu systemowi dwóch (lub więcej) odmiennych zestawów danych, system wynajduje reguły pozwalające na rozróżnienie, z którego zbioru pochodzi dowolny rekord do tej pory nie analizowany.

Sposób funkcjonowania procedur ML zostanie przedstawiony na przykładzie programu AQ19, [9], [13]. Celem działania tego programu jest wykrycie reguł, (hipotez), tzn. zestawu zdań orzekających o wartościach atrybutów analizowanej bazy danych, odpowiadających podzbiorowi bazy wyspecyfikowanemu jako zbiór treningowy.

Programy z rodziny AQ realizują formułowanie reguł (hipotez), na podstawie przykładów i kontrprzykładów. W zależności od nastawienia parametrów sterujących prac programu rozważane mogą być hipotezy kompletne i zgodne z danymi, dopuszczające lub nie przykłady negatywne, hipotezy nie pokrywające wszystkich przykładów pozytywnych, lub inne kombinacje. Niektóre z nich pozwalają brać pod

uwagę zakłócenia danych (ang. *noise*), inne pozwalają wymóc na systemie, aby przykłady wątpliwe (ang. *ambiguous*, np. rekordy zduplikowane) traktowane były jako pozytywne, etc.



Rys.6. Sposób funkcjonowania procedur MUM
Fig.6. The idea of Machine Learning procedure

Wszystkie reguły formułowane są jako iloczyny logiczne lub sumy logiczne (spójniki – odpowiednio – ORAZ i LUB), orzekających zdań elementarnych, określanych dalej jako selektory. Przykładami selektorów, które są zaznaczone nawiasami kwadratowymi, mogą być wyrażenia:

[x = biały, czarny, czerwony], (gdy x było pewną zmienną nominalną),

[x = 2..13], (co oznacza zakres wartości x od 2 do 13),

[x > 5,71], (co określa dolne ograniczenie wartości x), itp.

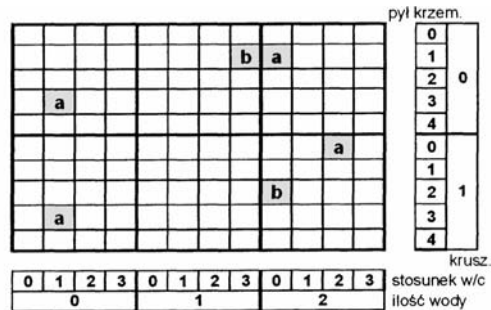
Najczęściej umieszczenie kilku selektorów jeden za drugim oznacza ich koniunkcję, (warunki muszą być spełnione równocześnie), umieszczenie w kolejnych wierszach oznacza ich dysjunkcję, (spełniony musi być którykolwiek z podanych warunków).

Koncepcją poszukiwania reguł wg opracowanych przez R. Michalskiego algorytmów AQ (AQ15, AQ17, AQ19, itd.; zob. np. [13]) opiera się na dopasowywaniu do przedłożonego zbioru danych tzw. „pojęć” (ang. *concept*). Myśl tę ilustruje rysunek 7. Poszczególne komórki wykresu kratkowego na rysunku odpowiadają rekordom wyszczególnionym w załączonej tabelicy. Rekordy opisane są przez poszczególne kombinacje wartości zmiennych (atrybutów): ilość wody, stosunek wodno cementowy, typ kruszywa, zawartość pyłu krzemionkowego. Zmienne te w przykładowej, prostej bazie danych mają od 2 do 5 rozmaitych dopuszczalnych wartości, (np. klasy zawartości pyłu krzemionkowego oznaczono liczbami od „0” do „3”).

Zadaniem programu jest wykrycie reguł pozwalających odróżnić rekordy klasy „a” od rekordów klasy „b”. Jedną z przykładowych hipotez, odpowiadającą sytuacji na rysunku 7 może być sformułowana jako:

Klasa "b" \Leftrightarrow [krusz=0] [pył krz.=1] [w/c=3] [il.wody=1]
ALBO
[krusz=1] [pył krz.=2] [w/c=0] [il.wody=2].

nr	pył krz.	krusz.	w/c	il. wody	klasa
1	3	0	1	0	a
2	3	1	1	0	a
3	1	0	0	2	a
4	0	1	2	2	a
5	1	0	3	1	b
6	2	1	0	2	b



Rys.7. Koncepcja *pojęcia* w uczeniu się maszyn. Rekordy opisane są wartościami atrybutów określającymi kategorie z uwagi na jedną z pięciu możliwych zawartości pyłu krzemionkowego, jeden z dwóch typów kruszyw, jedną z trzech możliwych zawartości wody, itd. Ustalenie przepisu na wyspecyfikowanie poszczególnych wartości atrybutów pozwala wyróżnić klasy zarobów „a” i „b”

Fig.7. The idea of a *concept* in Machine Learning. The records are described by collections of values of the attributes. Each class is defined by a certain combination of the values of the silica content (one of five possible values), type of aggregate (two possible types), water content (three possible values), and water cement value (four possibilities). It is possible to differentiate classes “a” and “b” by specifying values of these four attributes

Jakkolwiek notacja tego typu wydawać się może zbędnym komplikowaniem sprawy umożliwia ona automatyczne i sprawne przeszukiwanie i analizowanie dużych zbiorów danych. Ważne jest, że najnowsze implementacje algorytmu pozwalają na operowanie danymi ciągłymi bez potrzeby dyskretyzacji uwidocznionej w powyższym przykładzie.

Przykład zawartości skryptu danych wejściowych do AQ19, dotyczących oceny mrozoodporności betonu pokazano na rysunku 8.

variables								SF-names							
#	type	size	cost	name				value	name						
1	con	1	1	density,density				0	yes						
2	con	1	1	air,air				1	no						
3	con	1	1	spec_sur,spec_surf				FA_L-names						value name	
4	con	1	1	L,L											
5	con	1	1	fc28,fc28				0	yes						
6	nom	2	1	SF,SF				1	no						
7	nom	2	1	FA_L,FA_L				non_frost_resistant-events						value name	
frost_resistant-events								non_frost_resistant-events							
#	density	air	spec_surf	L	fc28	SF	FA_L	#	density	air	spec_surf	L	fc28	SF	FA_L
1	2380	5.10	?	?	57.8	no	no	1	2420	4.20	4.04	1.15	50.8	no	no
2	1340	6.90	9.76	0.25	36.9	no	no	2	2420	3.40	6.60	0.46	32.3	no	no
3	2380	6.70	?	?	47.9	no	no	3	2380	5.60	13.98	0.18	23.3	no	no
4	2440	2.90	?	?	56.1	no	no	4	2370	5.90	?	?	32.4	no	no
5	2504	7.40	25.00	0.12	65.9	no	no	5	2373	2.60	20.82	0.28	28.9	no	no
6	?	?	28.00	0.22	49.3	no	no	6	2315	2.20	14.02	0.52	38.8	no	no
7	2475	2.21	22.96	0.30	84.6	yes	no	7	2422	1.50	28.67	0.94	49.4	no	no
8	2498	4.05	13.01	0.37	91.7	yes	no	8	?	?	44.00	0.20	53.1	no	no
9	2372	4	20.63	0.24	79.3	yes	yes	9	?	?	30.00	0.21	55.7	no	no
10	2391	3	17.24	0.33	74.1	yes	yes	10	2467	4.46	18.20	0.27	71.5	no	yes
11	2379	4.08	17.17	0.29	86.3	yes	yes	③							
12	2453	5.11	16.47	0.28	76.7	yes	yes								
13	2468	4.92	21.77	0.23	73.9	yes	no								
14	2429	4.35	31.99	0.16	71.7	no	yes								

Rys.8. Fragmenty pliku wejściowego danych do programu AQ19. Na rysunku zgrupowano bloki skryptu definiujące rozpatrywane atrybuty (blok ①), określające legalne wartości atrybutów nominalnych (blok ②) oraz rekordy w dwóch grupach przykładów (bloki ③).

Przykłady po lewej dotyczą mieszanek odpornych na działanie mrozu, te po prawej – mieszanek nieodpornych

Fig.8. Elements of an input script for the AQ19 program. The blocks marked in the picture list the attributes of the records (block ①), the legal values of nominal attributes (block ②), and two groups of examples presented to the system (blocks ③). Examples on the left side concern concrete mixes considered to be frost resistant, those on the right side – mixes sensitive to frost

Po uruchomieniu programu generowane są uproszczone oceny statystyczne reguł, np. liczby rekordów zgodnych z regułą, liczba ocen błędnych, stopień komplikacji reguł itd., (zob. liczby w nawiasach okrągłych na rysunku 9).

Uzyskiwane reguły generowane metodami MUM są zależnościami empirycznymi, tzn. nie stanowią praw fizycznych, a dotyczą tylko zbiorów danych użytych podczas nauki.

W pokazanym na rysunku 9 przykładzie, na podstawie automatycznej analizy danych z bardzo niewielkiego zbioru obserwacji, uzyskaną regułę można sformułować np. jako (reguła #2, dotycząca mieszanek odpornych na działanie mrozu):

mieszanka betonowa należy do klasy „frost_resistant” (tzn. jest odporna na działanie mrozu), jeśli spełnione są jednocześnie dwa warunki:

- 1) wytrzymałość na ściskanie jest powyżej 76,7 MPa,

2) do mieszanki nie dodawano pyłu krzemionkowego.

frost_resistant-outhypo			
#	rule		
1	[density>2330.50][air>3.10][L<0.29][SF=no]	(t:15, u:10, n:3, q:0.609)	
2	[fc28>76.70][SF=no]	(t:11, u:10, n:1, q:0.555)	
3	[alfa=30.35..34.80][L>0.12]	(t:8, u:2, n:2, q:0.433)	
non_frost_resistant-outhypo			
#	rule		
1	[air>1.40] [alfa<26.80] [L>0.21]	(t:70, u:65, n:5, q:0.617)	
2	[air>6.75] [alfa<30.75]	(t:24, u:14, n:0, q:0.440)	
3	[fc28<67.55]	(t:16, u:14, n:1, q:0.303)	

Rys.9. Fragmenty pliku wyjściowego - reguły wykryte przez program AQ19

Fig.9. Elements of an output script generated by the AQ19 program. Presented are the rules discovered by the system

Kolejne reguły (#1 i #3) należy traktować jako alternatywne, tzn. łączyć je trzeba spójnikiem LUB.

4. KONCEPCJA ALGORYTMÓW GENETYCZNYCH

Algorytmów genetyczne (GA) związane są z metaforami pojęć pochodzących z biologii, a dotyczących reprodukcji i dziedziczenia. Metody te są przydatne przy rozwiązywaniu problemów, w których tradycyjne sposoby obliczeń natrafiają na trudności, a proste sprawdzanie wszystkich możliwości jest niesłychanie pracochłonne. Dotyczy to zwłaszcza takich wypadków, gdy wiedza a priori o badanym problemie jest ograniczona.

Te same metafory występują w wypadku innych technik SI, określanych np. jako algorytmy genetyczne, strategie ewolucyjne, programowanie ewolucyjne czy programowanie genetyczne, (ang. *Genetic Algorithms – GA, Evolution Strategies – ES, Evolutionary Programming – EP, Genetic Programming – GP*). Różnice między nimi dotyczą raczej zastosowań, (np. opracowywanie schematów programów komputerowych w wypadku GP), a nie zasadniczej idei poszukiwania rozwiązań poprzez mniej lub bardziej przypadkowe generowanie rozwiązań niedoskonałych, ocenianych później celem wybrania przypadków najwartościowszych, które są następnie użyte do wytworzenia nowej lepszej generacji rozwiązań (rozwiązania lepsze są w pewien sposób „rozmnażane”). Wielokrotne powtarzanie zabiegów oceny, wyboru i reprodukcji ma skutek optymalizowania rezultatów, a analogia z odkryciami w biologii jest oczywista.

Jakkolwiek metody GA w zadaniach optymalizacji nie gwarantują znalezienia optimum globalnego, jednak zapewniają wskazanie rozwiązania wystarczająco dobrego,

i to w dostatecznie krótkim czasie. Uwaga dotycząca ograniczenia czasu („dostatecznie”) jest ważna, ponieważ w niektórych wielu problemach kombinatorycznych rozpatrzenie nawet ułamka procentu wszystkich możliwości nie jest technicznie realne. Metody ewolucyjne nie tworzą modeli. Ich zaletą jest odporność na zakłócenia (w szczególności tzw. odporność na ekstremum lokalne), i możliwość równoległej organizacji obliczeń.

Procedura GA polega na wygenerowaniu w systemie obliczeniowym (w maszynie liczącej) pewnej populacji rozwiązań, złożonej z osobników (rozwiązań), reprezentowanych przez tzw. chromosomy. Chromosomy to kolejna metafora biologiczna – chodzi o pełną liczbową charakterystykę rozwiązań spełniających zadane warunki brzegowe. Chromosomy stanowią odpowiednio zakodowane w formie łańcuchów znaków rozwiązania danego zadania, np. reprezentując punkt w przestrzeni rozwiązań dopuszczalnych. Zależnie od sytuacji mogą występować jako tablice liczb, wektory, ale także reguły i elementy programów.

W najprostszym przypadku łańcuchy mają formę binarną, a każda kombinacja zer i jedynek stanowi jakieś rozwiązanie problemu. Na zbiorze takich chromosomów dokonuje się specyficznych operacji: reprodukcji „Darwinowskiej” (rozmnażają się najlepsi), oraz modyfikacji grup osobników poprzez krzyżowanie i mutacje. Operacja krzyżowania odpowiada ideowo przekazywaniu genów w akcie prokreacji: dwa (ewentualnie więcej niż dwa) łańcuchy chromosomu są rozrywane, po czym następuje wymiana utworzonych w ten sposób członów. Tworzone jest w ten sposób „potomstwo” ostatniej generacji, (rys. 10). Na zbiorze nowych chromosomów przeprowadzana jest ocena ich jakości, (ocena jakości odpowiadających im rozwiązań) i selekcja – wybór osobników lepiej przybliżających rozwiązanie docelowe.

Postępowanie przy kodowaniu rozwiązań do postaci ciągów binarnych odpowiadających poszukiwanym rozwiązaniom nie jest tu omawiane, ale można je znaleźć w monografiach z zakresu metod ewolucyjnych, np. [14]. Operacje – na chromosomach – krzyżowania i mutacji można przedstawić przykładowo jak na rysunku 10.

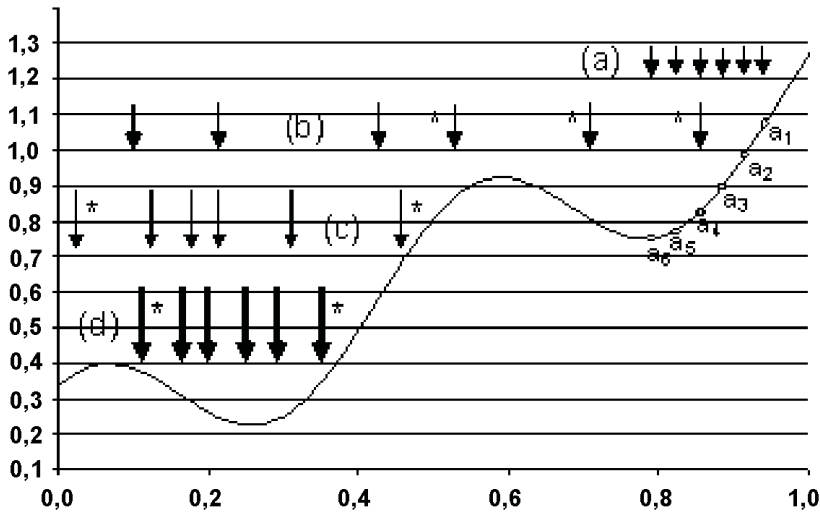
osobnik1	1 0 0 0 1 0 1 1 1 1 0 1 1 1
osobnik2	1 1 1 0 0 1 0 0 1 1 1 0 1 1
potomek1	1 0 0 0 1 0 0 0 1 1 1 0 1 1
potomek2	1 1 1 0 0 1 1 1 1 1 0 1 1 1
mutacja osobnika1	1 0 1 0 1 0 1 1 1 1 0 1 1 1

Rys. 10. Przykład operacji krzyżowania i mutowania chromosomów. „Osobniki”, „potomkowie” i „mutanci”, to zakodowane rozwiązania zadania

Fig. 10. Examples of crossover and mutation operations on chromosomes. The chromosomes (the strings of genes) are coded solutions of problem under consideration

Warunkiem procedury kodowania jest, aby łańcuchy potomków i mutantów były nadal jakimiś rozwiązaniami problemu.

Na rysunku 11 pokazana jest prosta sytuacja poszukiwania ekstremum nieskomplikowanej funkcji: $y = 0,05 + x + 0,2 \cdot \sin(12,0 \cdot x + 1,2)$, rozważanej na odcinku $x \in (0,1)$. Pomijając naturalną oczywistość znalezienia minimum tej funkcji metodami analitycznymi, można na przytoczonym przykładzie porównać sposoby rozwiązywania w tradycyjnym ujęciu numerycznym z postępowaniem ewolucyjnym. Poszukiwanie według gradientu funkcji prowadzi do obliczenia wartości funkcji w ciągu punktów zaczynających się od przypadkowo wybranego punktu – np. punktu a_1 na rysunku 11 – przypadek (a), który ewentualnie zakończy się w lokalnym ekstremum funkcji, w punkcie (a_6), ponieważ dalej wartości funkcji zaczęłyby znowu rosnąć.



Rys.11. Poszukiwanie ekstremum prostej funkcji metodą gradientową - ciąg przybliżeń (a) oraz metodą ewolucyjną - ciąg populacji rozwiązań (b)-(c)-(d)

Fig.11. Searching for the extreme of a simple function by gradient method – a series of approximations (a), and by evolutionary process – a series of populations (b)-(c)-(d).

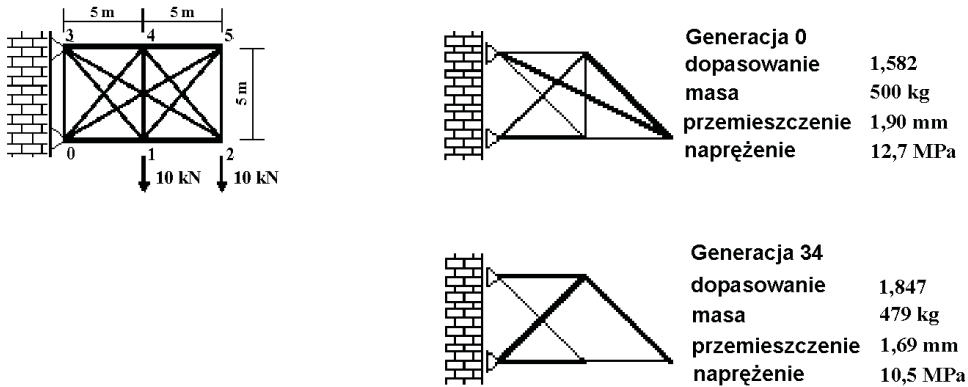
Alternatywą jest rozważanie kolejnych generacji rozwiązań. Na wykresie grupy strzałek symbolizują po kilka osobników w trzech rodzinach rozwiązań GA: (b), (c) i (d), które coraz lepiej przybliżają minimum. Jak można się domyślić rozwiązania (c) stanowią „potomstwo” „lepszyc” rozwiązań (b), a potomstwem tych są z kolei rozwiązania (d). W kolejnych populacjach rozwiązania „gorsze” zaznaczone gwiazdkami (*) są pomijane przy krzyżowaniu i mutowaniu, co w efekcie prowadzi do pożądanego wyspecjalizowania się generowanych osobników.

Postępowanie ewolucyjne jest przerywane z chwilą spełnienia pewnego kryterium zatrzymania procedury. Następuje to albo na podstawie osiągniętej wartości funkcji przystosowania, albo na podstawie innych kryteriów, (np. kryterium maksimum kosztu obliczeń, kryterium minimalnej szybkości poprawy wyniku, kryterium czasu obliczeń), [15]. Wynikiem jest wówczas wygenerowanie - w postaci zakodowanej w strukturze chromosomów - rozwiązań bliskich poszukiwanemu.

W monografiach z zakresu metod ewolucyjnych wymieniane są typowe problemy obliczeniowe skutecznie rozwiązywane metodami algorytmów genetycznych. Jednym z takich zadań jest problem komiwojażera. Chodzi o to, aby przy ustalonej siatce połączeń między miastami i przy znanych cenach przejazdów ustalić optymalną trasę komiwojażera. Ma on odwiedzić każde miasto dokładnie jeden raz, minimalizując jednocześnie koszty podróży. Podobnym przykładem jest zadanie maksymalnego upakowania plecaka o zadanej objętości, korzystając ze zbioru elementów o znanej objętości, kształcie i ciężarze. Przemawiające do wyobraźni są zadania planowania produkcji na hali produkcyjnej (ang. *job-shop scheduling*), wyposażonej w maszyny o określonych możliwościach przerobowych, wytwarzających określone asortymenty, z możliwościami rozmaitej realizacji procesu produkcji poszczególnych elementów na poszczególnych maszynach, przy założeniu zachowania kolejności operacji i wymaganiu ich ukończenia na kolejnych stanowiskach. Przy nieco tylko wyższej liczbie stanowisk (*jobs*) zadanie robi się tak pracochłonne, że – wg [16] – przy 15 operacjach przeszukiwanie kombinacji na komputerze dokonującym 1000 prób na sekundę zajęłoby ponad 41 lat, a przy 50, gdyby rozpocząć próby wraz z początkiem wszechświata, to do czasów współczesnych nie wyszłoby się poza znikomy ułamek procentu wszystkich możliwości - rzędu $1/10^{39}$ (!).

W zakresie inżynierii lądowej istnieje wiele publikacji dotyczących konkretnych zastosowań metod ewolucyjnych. Szereg opracowań dotyczy organizacji ruchu czy planowania i optymalizacji połączeń drogowych, programowania napraw, rehabilitacji, modernizacji nawierzchni, itp., [17]. GA stosowano także celem rozplanowania przekrojów stali zbrojeniowej w konstrukcji żelbetowej, [18].

Jedno z pierwszych inżynierskich zastosowań GA miało miejsce prawdopodobnie przy okazji projektowania kratownic - Goldberg i Samtani 1986, [19]. Wydaje się, że są to nadal najbardziej typowe zastosowania inżynierskie GA. Dotyczą one nie tylko projektowania optymalnej grubości przekrojów prętów kratownic statycznie niewyznaczalnych, ale nawet optymalizacji topologii kratownicy. Na rysunku 12 pokazano przykład takiej zamiany topologii w wyniku procesów adaptacyjnych, wg [20].



Rys.12. Przykład zadania optymalizacji topologii i pól przekrojów wspornikowej kratownicy 15-prętowej. Pokazana jest konfiguracja wyjściowa i rozmieszczenie prętów w dwóch stanach pośrednich, (wg [20])

Fig.12. An example of a problem of the truss topology optimization. Observed can be the initial configuration of the truss layout and two intermediate states, (according to [20])

5. PODSUMOWANIE

Omówione tu, wybrane metody SI pozwalają na efektywne poszukiwanie rozwiązań bez- lub z bardzo ograniczonym wykorzystaniem modeli stanowiących wynik przemyśleń inżyniera.

W zagadnieniach tradycyjnych, w których znane są zadowalające sposoby prowadzenia obliczeń korzystanie z metod SI może nie być uzasadnione. Jednak przy gwałtownie komplikującej się różnorodności nowoczesnych rozwiązań materiałowych i konstrukcyjnych metody sztucznej inteligencji otwierają nowe, praktyczne możliwości uzyskiwania wartościowych rozwiązań inżynierskich.

Metody SSN pozwalają przewidywać wektory odpowiedzi (reakcje) towarzyszące zadany wartościom wejściowym wektorów przyczyn (impulsów), podczas gdy metody MUM pozwalają w sposób automatyczny odgadywać zasady opisujące badane procesy. Zarówno metody SSN, MUM jak i GA mogą być łączone w formie rozwiązań hybrydowych, o interesujących, jakkolwiek na ogół jeszcze nie rozpoznanych możliwościach.

BIBLIOGRAFIA

- [1] *Timoshenko S.P.*: Historia wytrzymałości materiałów, 476, Arkady, Warszawa 1966
 [2] *Wasiutyński Z.*: Pisma. Tom I, 344, PWN, Warszawa 1977

-
- [3] *Hertz J., Krogh A., Palmer R. G.*: Wstęp do teorii obliczeń neuronowych. 368, WNT, Warszawa 1995
- [4] *Kasperkiewicz J., Racz J., Dubrawski A.*: HPC strength prediction using artificial neural network. *Journal of Computing in Civil Engineering*, **9**, No. 4, 279-284, October 1995
- [5] *Wu X., Ghaboussi J., Garrett J.H. Jr.*: Use of neural networks in detection of structural damage. *Computers & Structures*, **42**, No.4, 649-659, 1992
- [6] *Kasperkiewicz J.*: The applications of ANNs in certain materials-analysis problems. *Journal of Materials Processing Technology*, 106 (2000), 74-79
- [7] *King M.A., Elder IV J.F., et al.*: Evaluation of fourteen desktop data mining tools. 1998 IEEE International Conference on Systems, Man, and Cybernetics, 6, San Diego, CA, 12-14 October 1998
(zob. także: http://www.datamininglab.com/pubs/smc98_king_elder.pdf)
- [8] *Cichosz P.*: Systemy uczące się. WNT, 894, Warszawa 2000
- [9] *Michalski R.S., Carbonell J.G., Mitchell T.M.* (Ed-s): Machine learning. An artificial intelligence approach. **II**, 738, Morgan Kaufmann Publishers, Palo Alto 1986
- [10] *Mitchell T.M.*: Machine learning. WCB/MacGraw-Hill, 414, Boston 1997
- [11] *Jha M.K., McCall C., Schonfeld P.*: Using GIS, Genetic Algorithms, and visualization in highway development. *Computer-Aided Civil and Infrastructure Engineering*, 16(6), 399-414, 2001
(zob. także Integrating Genetic Algorithms and GIS to Optimize Highway Alignments; Authors: Jha, M.K. and P. Schonfeld; Transportation Research Board; 79 th Annual Meeting, Washington, D.C., 9-13 January 2000)
- [12] *Nunoo C., Jolibois S.C., Agyei E.*: Optimization of pavement preservation programming using Shuffled Complex Evolution Algorithm. A paper prepared for presentation at the 81st Annual Meeting of Transportation Research Board Washington, D.C. January 2002; TRB 2002; 81-st Annual Meeting CD-ROM
- [13] *Michalski R.S., Kaufman K.A.*: The AQ19 system for machine learning and pattern discovery: a general description and user's guide. George Mason University, 39, MLI 01-2, March 2001
- [14] *Michalewicz Z.*: Algorytmy genetyczne+struktury danych=programy ewolucyjne. WNT, 430, Warszawa 1996
- [15] *Arabas J.*: Wykłady z algorytmów ewolucyjnych. WNT, 303, Warszawa 2002
- [16] *Flood I.*: Solving constructional operational problems using artificial neural networks and simulated evolution. *Proceedings of CIB 90*, (Congress Sydney 14-21 Mar. 1990), 197-208, CSIRO 1990
- [17] *Raja A. Shekharan*: Solution of pavement deterioration equations by genetic algorithms. Transportation Research Board, 79 th Annual Meeting January 9-13, Washington, D.C., 2000

- [18] *Koumoussis V.K., Arsenis S.J.*: Genetic algorithms in optimal detailed design of reinforced concrete members. *Computer-Aided Civil and Infrastructure Engineering*, **13**, iss. 1, 43-52, 1998
- [19] *Goldberg, D.E. and Samtani, M.P.*: Engineering optimization via genetic algorithms. *Proceedings of the Ninth Conference on Electronic Computation*, 471-482, ASCE, New York 1986
- [20] *Kawamura H., Ohmori H.*: Computational morphogenesis of discrete structures via genetic algorithms. *Memoirs of the School of Engineering*, **53**, No.1/2 (2001), 28-55, Nagoya University

ON POSSIBILITIES OF APPLICATION OF ARTIFICIAL INTELLIGENCE METHODS IN CIVIL ENGINEERING

Abstract

Described are possibilities of Civil Engineering applications of computing methods originating in so called Artificial Intelligence. The three basic concepts are Artificial Neural Networks, Machine Learning and Genetic Algorithms. In case of all these solutions the user is not obliged to bother about the model of the process or phenomenon, because the system itself gains the knowledge from examples. It can generate thereupon answers in form of the unknown values of the attributes, classification of new examples of the same format or formulation of rules (hypotheses, generalisations) concerning the process under consideration. More details are given in relation to the applied solutions of Fuzzy ARTMAP and ML program AQ19.