

Politechnika Śląska  
Wydział Automatyki, Elektroniki i Informatyki  
Instytut Informatyki

Rozprawa doktorska pt.:  
SYSTEMY EWOLUCYJNO-ROZMYTE  
Z PARAMETRYCZNYMI KONKLUZJAMI  
W REGUŁACH **IF-THEN**.

mgr inż. Piotr Czekalski

Promotor:  
prof. dr hab. inż. Jacek Łęski

Gliwice, 2004



## **Podziękowania**

Pragnę serdecznie podziękować promotorowi **prof. dr hab. inż. Jackowi Łęskiemu** za okazaną pomoc podczas przygotowywania niniejszej rozprawy doktorskiej. W szczególności dziękuję za czas poświęcony na konsultacje, za pełne profesjonalizmu podejście oraz wyjątkową wyrozumiałość.

Dziękuję również **dr inż. Henrykowi Małysiakowi** za opiekę w trakcie trwania studiów doktoranckich oraz za wsparcie i motywację w trakcie pracy nad rozprawą.

*Niniejszą rozprawę pragnę zadedykować  
mojemu dziesięciomiesięcznemu synowi  
Erykowi, który zdecydowanie wygrywa  
wyścig po wiedzę, nawet z najlepszymi  
algorytmami i najszybszymi komputerami.*



# Spis treści

<b>Wykaz oznaczeń i skrótów .....</b>	<b>vii</b>
<b>1 Wprowadzenie .....</b>	<b>1</b>
1.1 Tematyka rozprawy .....	1
1.2 Teza rozprawy .....	5
<b>2 Systemy rozmyte .....</b>	<b>7</b>
2.1 Wstęp do teorii zbiorów rozmytych .....	7
2.1.1 Relacja rozmyta .....	9
2.1.2 Zmienna lingwistyczna .....	9
2.2 Interpretacja reguł if-then .....	10
2.2.1 Wnioskowanie z wykorzystaniem rozmytej bazy reguł .....	14
2.3 Modele systemów rozmytych .....	23
2.3.1 System rozmyty z parametryzowanymi konkluzjami .....	27
2.4 Podsumowanie .....	34
<b>3 Metody ewolucyjne .....</b>	<b>35</b>
3.1 Techniki naśladowania ewolucji biologicznej .....	35
3.2 Algorytmy genetyczne .....	37
3.2.1 Podstawy analityczne .....	38
3.2.2 Kodowanie osobników .....	41
3.2.3 Reprodukacja .....	42
3.2.4 Operatory genetyczne .....	44
3.2.5 Warunki zatrzymania .....	47
3.3 Strategie ewolucyjne .....	47
3.3.1 Modele strategii ewolucyjnych .....	48
3.3.2 Kodowanie osobników .....	48
3.3.3 Podstawy analityczne .....	49
3.3.4 Operatory ewolucyjne .....	49
3.4 Programowanie ewolucyjne .....	53
3.5 Podsumowanie .....	53
<b>4 Systemy ewolucyjno-rozmyte .....</b>	<b>55</b>
4.1 Strojanie bazy danych systemu rozmytego .....	58
4.2 Uczenie bazy reguł .....	59

4.3	Uczenie bazy wiedzy.....	60
4.4	Inne techniki.....	61
4.5	Przykłady systemów ewolucyjno-rozmytych.....	62
4.5.1	System ewolucyjno-rozmyty Mamdaniego-Assiliana oparty na trójetapowej metodzie uczenia.....	62
4.5.2	System ewolucyjno-rozmyty TSK oparty na dwuetapowej metodzie uczenia...	66
4.6	Podsumowanie .....	68
<b>5</b>	<b>Zastosowanie metod ewolucyjnych do projektowania systemów rozmytych z parametryzowanymi konkluzjami .....</b>	<b>69</b>
5.1	Pokrywanie danych rozmytymi regułami if-then.....	71
5.1.1	Algorytm pokrywania zbioru uczącego .....	71
5.1.2	Algorytm odnajdywania najlepszej reguły .....	73
5.2	Redukcja bazy reguł.....	86
5.3	Strojenie bazy danych systemu rozmytego .....	93
<b>6</b>	<b>Wyniki przeprowadzonych badań .....</b>	<b>103</b>
6.1	Implementacja środowiska testowego.....	103
6.1.1	Moduł systemu rozmytego z parametryzowanymi konkluzjami .....	103
6.1.2	Implementacja modułu obliczeń ewolucyjnych.....	106
6.1.3	Implementacja - podsumowanie .....	108
6.2	Przeprowadzone badania.....	109
6.2.1	Identyfikacja pieca gazowego (Box-Jenkins) .....	109
6.2.2	Przewidywanie liczby plam na słońcu (Sunspots).....	123
6.2.3	Predykcja chaotycznych szeregów czasowych (Mackey–Glass).....	128
6.2.4	Wycena wartości nieruchomości (Boston Housing).....	135
6.3	Czasy obliczeń i platforma sprzętowa.....	139
<b>7</b>	<b>Podsumowanie i wnioski końcowe .....</b>	<b>143</b>
	<b>Literatura .....</b>	<b>147</b>
	<b>Dodatek A – parametry funkcji przynależności oraz parametry konkluzji.....</b>	<b>159</b>

## Wykaz oznaczeń i skrótów

$x, y$	element zbioru
$A, B, C$	zbiór
$\emptyset$	zbiór pusty
$A_{II}$	zbiór rozmyty typu II
$\mathbb{X}, \mathbb{Y}, \mathbb{Z}$	przestrzeń rozważań, przestrzeń fenotypu
$[a, b]$	przedział domknięty pomiędzy liczbami $a$ i $b$
$\mu_A(\bullet)$	funkcja przynależności do zbioru rozmytego $A$
$\mu_{AII}(\bullet, \bullet)$	funkcja przynależności do zbioru rozmytego $A$ , typu II
$\{\bullet, \bullet, \dots, \bullet\}$	opis zbioru za pomocą listy elementów
$\{\bullet   a\}$	opis zbioru za pomocą cechy $a$
$\in$	przynależność do zbioru
$\subseteq$	zawieranie się zbiorów
$\cap$	iloczyn zbiorów
$\cup$	suma zbiorów
$\forall$	kwantyfikator ogólny
$\exists$	kwantyfikator szczegółowy
$(\bullet \oplus \bullet)$	operacja XOR na argumentach
$\text{card}(\bullet)$	moc zbioru
$\text{dim}(\bullet)$	długość wektora

$T(\bullet, \bullet), \star_T$	<i>t-norma</i>
$S(\bullet, \bullet), \star_S$	<i>s-norma (t-konorma)</i>
$\star_{Ti}$	<i>t-norma</i> dla wielowejściowych przesłanek
$\star_{Tc}$	<i>t-norma</i> stosowana przy koniunkcyjnej interpretacji reguł <b>if-then</b>
$\star_{Tz}$	<i>t-norma</i> złożenia
$A \times B$	iloczyn kartezjański zbiorów $A$ i $B$
min	operator minimum
max	operator maksimum
round( $\bullet$ )	operator zaokrąglenia do najbliższej wartości całkowitej
mod( $\bullet$ )	modulo
abs( $\bullet$ )	wartość bezwzględna
ones( $\bullet$ )	operacja utworzenia wektora jednostkowego o zadanej parametrem długości
$Ce(A)$	rozszerzenie cylindryczne zbioru $A$
$Proj_{\mathbb{X}}(B)$	projekcja zbioru $B$ na przestrzeń $\mathbb{X}$
sup	kres górny zbioru
$\mu_R(\bullet, \bullet)$	funkcja przynależności relacji rozmytej $R$
$f: \mathbb{X} \rightarrow \mathbb{Y}$	odwzorowanie $f$ przestrzeni $\mathbb{X}$ na przestrzeń $\mathbb{Y}$
$X$	zmienna lingwistyczna
$\mathcal{N}$	nazwa zmiennej lingwistycznej
$\mathcal{T}(\mathcal{G})$	zbiór terminów lingwistycznych
$\mathcal{G}$	zbiór reguł syntaktycznych
$\mathcal{S}$	zbiór reguł semantycznych



$X_i$	$i$ -ta zmienna lingwistyczna w przesłance reguły
$Y_i$	$i$ -ta zmienna lingwistyczna w konkluzji reguły
$A_i$	$i$ -ty zbiór rozmyty w przesłance reguły
$B_i$	$i$ -ty zbiór rozmyty w konkluzji reguły
$R_{MIMO}$	reguła rozmyta o wielu wejściach i wielu wyjściach
$R_{MISO}$	reguła rozmyta o wielu wejściach i jednym wyjściu
$N$	liczba wejść systemu rozmytego
$n$	numer wejścia systemu rozmytego
$P$	liczba wyjść systemu rozmytego
$\mathbb{X}$	wielowymiarowa przestrzeń rozważań
$\underline{X}$	wektor wejściowych zmiennych lingwistycznych
$\underline{x}$	wektor wartości wejściowych
$\underline{x}_0(n_T)$	$n_T$ -ty wektor danych wejściowych (uczących)
$\underline{x}^T$	wektor transponowany
$\underline{x}'_0$	rozszerzony wektor danych wejściowych
$\underline{x}_0^{\leftarrow}, \underline{x}_0^{\rightarrow}$	wektor zawierający odpowiednio lewe i prawe ograniczenia przedziału zmienności wartości wejściowych ze zbioru uczącego
$x_i^{\leftarrow}, x_i^{\rightarrow}$	wektor zawierający odpowiednio lewe i prawe ograniczenia przedziału zmienności $i$ -tego argumentu
$\underline{\tilde{x}}_0^d$	wektor zakresów zmienności wszystkich parametrów bazy wiedzy
$A \Rightarrow B$	$A$ implikuje $B$
$\neg$	zaprzeczenie logiczne
$\wedge$	iloczyn logiczny

$\vee$	suma logiczna
$I(\bullet, \bullet)$	implikacja rozmyta
$\mathcal{R}$	baza reguł
$\mathcal{R}_{II}$	baza reguł zoptymalizowana
$L, I$	liczba reguł
$I_{II}$	liczba reguł w bazie wiedzy po drugim etapie trójetapowej metody uczenia
$I_{\tilde{x}}$	liczba reguł w bazie reprezentowanej przez osobnika $\tilde{x}$
$R^{(l)}$	$l$ -ta reguła rozmyta lub odpowiadająca jej relacja rozmyta
$\circ$	złożenie relacji
$\mathcal{R}$	zbiór liczb rzeczywistych
$\mathcal{R}^N$	iloczyn kartezjański $N$ zbiorów liczb rzeczywistych
$\oplus$	agregacja
$\Psi_{(\alpha)}$	operator uogólnionej średniej
$\alpha_i$	$i$ -ty parametr metody MICOG
$a_n, b_n, c_n$	parametry $n$ -tej, trójkątnej funkcji przynależności (lewy, środek, prawy)
$s_n / c_n, \sigma_n$	parametry $n$ -tej, gaussowskiej funkcji przynależności (środek, dyspersja)
$\mathcal{F}(\mathbb{Y})$	rodzina zbiorów rozmytych, określonych na przestrzeni $\mathbb{Y}$
$y^*$	wyostrzony element wyjściowego zbioru rozmytego
$y_0$	numeryczna wartość wyjściowa systemu rozmytego
$\underline{y}_0$	wektor numerycznych wartości wyjściowych systemu rozmytego
$\tau_i$	wysokość $i$ -tego zbioru wyjściowego
$\underline{D}$	macierz aktywacji

$\underline{D}_1$	macierz rozszerzonych wektorów wejściowych
$G^{(i)}$	macierz wag dla $i$ -tej reguły rozmytej
$E_{NT}$	zbiór uczący
$\underline{T}$	wektor wartości wyjściowych ze zbioru uczącego
$N_T$	liczebność zbioru uczącego
$\underline{x}_0(n_T)$	$n_T$ -ty wektor danych wejściowych (uczących)
$\underline{t}_0(n_T)$	$n_T$ -ta wartość wyjściowa (ucząca)
$t_0^{\leftarrow}, t_0^{\rightarrow}$	lewa i prawa granica zmienności wartości wyjściowych ze zbioru uczącego
$\underline{c}_0^{\leftarrow}, \underline{c}_0^{\rightarrow}$	wektor zawierający odpowiednio lewe i prawe ograniczenia przedziału zmienności środków funkcji gaussowskich w funkcjach przynależności przesłanek reguł
$\underline{\sigma}_0^{\leftarrow}, \underline{\sigma}_0^{\rightarrow}$	wektor zawierający odpowiednio lewe i prawe ograniczenia przedziału zmienności dyspersji funkcji gaussowskich w funkcjach przynależności przesłanek reguł
$\underline{w}_0^{\leftarrow}, \underline{w}_0^{\rightarrow}$	wektor zawierający odpowiednio lewe i prawe ograniczenia przedziału zmienności nośnika funkcji w trójkątnych w funkcjach przynależności konkluzji reguł
$\mathcal{P}(\bullet)$	pole powierzchni
$F^{(i)}$	stopień aktywacji $i$ -tej reguły rozmytej
$\theta$	parametry konkluzji systemu rozmytego TSK
$\underline{p}^{(i)}$	wektor parametrów dla $i$ -tej reguły
$\underline{P}$	wektor parametrów konkluzji systemu rozmytego z parametryzowanymi konkluzjami
$w^{(i)}$	długość podstawy trójkąta w funkcji przynależności konkluzji $i$ -tej reguły
$\mu_B^{(i)*}(\bullet)$	funkcja przynależności konkluzji $i$ -tej reguły, po usunięciu części nieinformatywnej
$f$	funkcja przystosowania

$f'$	zmodyfikowana funkcja przystosowania
$\underline{f}$	wektor z ocenami osobników
$\underline{f}_{\mathcal{J}}, \underline{f}_{\mathcal{J}_c}, \underline{f}_O$	wektory z ocenami pokoleń, odpowiednio $\mathcal{J}^{(k)}, \mathcal{J}_c^{(k)}, O^{(k)}$
$f_J(\bullet, \bullet)$	funkcja kary
$J_{RMSE}(\bullet, \bullet)$	błąd średniokwadratowy, wyliczony dla zadanego systemu i wskazanych danych
$J_R$	wskaźnik kompletności pokrycia
$\mathbf{v}_s$	wektor sumy przyrastającej wartości funkcji przystosowania
$\mathbf{v}_L$	wektor markujący osobniki niewykorzystane w procesie wyboru rodziców
$\mathcal{M}$	przestrzeń dozwolonych rozwiązań
$\rho(\bullet, \bullet)$	miara pseudometryczna w przestrzeni rozwiązań
$\underline{x}^{\uparrow L}$	optimum lokalne
$g_j$	$j$ -te ograniczenie nierównościowe
$h_j$	$j$ -te ograniczenie równościowe
$Q^{(k)}$	$k$ -te pokolenie bazowe
$\mathcal{J}^{(k)}$	$k$ -te pokolenie tymczasowe
$\mathcal{J}_4$	pokolenie pomocnicze, wykorzystywane podczas krzyżowania max-min-arytmetycznego
$\mathcal{J}_c^{(k)}$	$k$ -te pokolenie pośrednie
$O^{(k)}$	$k$ -te pokolenie potomne
$q_B$	najlepszy osobnik (w modelu elitarnym)

$i_{P1}, i_{P2}$	indeksy osobników rodzicielskich
$j_{P1}, j_{P2}$	osobniki rodzicielskie wybrane do krzyżowania
$\underline{C}_{P1}, \underline{C}_{P2}$	chromosomy osobników rodzicielskich
$\underline{C}_{Ci}$	chromosom $i$ -tego osobnika potomnego
$\underline{C}_{\rho,i}$	podciąg chromosomu, reprezentujący $i$ -tą cechę osobnika o chromosomie złożonym
$\underline{C}_M$	chromosom osobnika mutowanego
$\underline{C}_M^{(i)}$	$i$ -ty gen w chromosomie (podczas operacji mutacji)
$k$	indeks pokolenia algorytmu genetycznego / strategii ewolucyjnej
$K$	ustalona, maksymalna liczba pokoleń algorytmu genetycznego / strategii ewolucyjnej
$l$	pozycja w ciągu kodowym chromosomu
$\mathcal{L}$	długość ciągu kodowego chromosomu
$\zeta$	punkt rozcięcia w operacji krzyżowania
$o(S)$	rząd schematu $S$
$\alpha(S)$	długość definiująca schematu $S$
$\xi$	liczebność schematu
$P^S$	wzorzec prawdopodobieństwa schematu $S$
$p_r$	prawdopodobieństwo
$p_{rM}$	prawdopodobieństwo mutacji
$p_{rC}$	prawdopodobieństwo krzyżowania
$p_{rCS}(S)$	prawdopodobieństwo przeżycia schematu $S$ w wyniku krzyżowania

$p_{rMS}(S)$	prawdopodobieństwo przeżycia schematu S w wyniku mutacji
$P(\bullet)$	wartość dystrybucyjności
$E(\bullet)$	wartość oczekiwana
$M$	liczebność pokolenia
$\tilde{\mathbf{X}}$	przestrzeń genotypu
$\tilde{x}$	osobnik w przestrzeni genotypu (rozróżniane, jeśli genotyp nie jest tożsamy z fenotypem)
$\tilde{\underline{x}}$	osobnik w przestrzeni genotypu, gdy istotna jest jego postać wektorowa
$\tilde{\underline{x}}_\rho$	osobnik o genotypie złożonym
$\tilde{\underline{x}}_\alpha, \tilde{\underline{x}}_\beta$	osobniki rodzicielskie
$\tilde{\underline{x}}_1, \tilde{\underline{x}}_2$	osobniki potomne
$b_l$	pozycja ciągu kodowego
$r, r_b, r_i', r_l$	realizacja zmiennej losowej
$\underline{M}_m$	macierz mutująca, macierz realizacji zmiennej losowej
$\underline{c}_x, c_y$	wektor parametrów położenia środków krzywych gaussowskich w przesłankach oraz położenie środka ciężkości trójkąta w konkluzji
$\mu, \kappa, \lambda$	parametry definiujące strategię ewolucyjną
$\mathcal{K}$	funkcja konwersji fenotypu do genotypu
$\mathcal{D}$	funkcja konwersji genotypu do fenotypu
$\lceil \bullet \rceil$	sufit
$\lfloor \bullet \rfloor$	podłoga
$\ \bullet\ _g$	norma w przestrzeni genotypu

$\ \bullet\ _f$	norma w przestrzeni fenotypu
$\mathcal{V}(k)$	współczynnik sukcesów w $k$ poprzednich pokoleniach
$n_d, n_b, \tau_1, \tau_2$	parametry mutacji
$a$	parametr krzyżowania uśredniającego
$b$	parametr periodycznej mutacji nierównomiernej
$G_L$	parametr określający brzegową wartość krzywej gaussowskiej
$\kappa$	parametr określający stosunek rekordów pokrywanych do niepokrywanych
$\varepsilon$	minimalny stopień pokrycia danych uczących
$E_C$	podzbiór zbioru uczącego, zawierający niepokryte dane uczące
$E_G$	podzbiór zbioru uczącego, używany do generowania pokolenia bazowego
$C_i(\bullet)$	wartość pokrycia dla wskazanej danej przez $i$ -tą regułę
$R_c^{(i)}(\bullet)$	stopień kompatybilności $i$ -tej reguły z bazy reguł, ze wskazanym rekordem danych uczących
$R_c(\bullet, \bullet)$	stopień kompatybilności wskazanej reguły ze wskazanym rekordem danych uczących
$\overline{\Psi}_{NE}(\bullet, \bullet)$	średnia kompatybilność reguły ze zbiorem danych
$C_p(\bullet, \bullet)$	średnie pokrycie danych przez regułę
$C_n(\bullet, \bullet)$	stopień dopasowania konkluzji reguły
$\mathcal{G}(\bullet, \bullet)$	funkcja oceniająca poziom szerokości dyspersji przesłanek i nośnika konkluzji reguły
$\Xi(\bullet, \bullet)$	funkcja oceniająca podobieństwo reguły do innych reguł w bazie wiedzy
$C_V(\bullet)$	wartość pokrycia wskazanego zbioru danych przez bieżącą bazę wiedzy

$\delta(\bullet, \dots, \bullet)$	realizacja zmiennej losowej o rozkładzie równomiernym, na podanym przedziale otwartym
$\delta[\bullet, \dots, \bullet]$	realizacja zmiennej losowej o rozkładzie równomiernym, na podanym przedziale zamkniętym
$\delta\{\bullet, \dots, \bullet\}$	realizacja zmiennej losowej o rozkładzie równomiernym, na podanym zbiorze wartości dyskretnych
FITA	model <i>najpierw wnioskowanie, potem agregacja</i>
FATI	model <i>najpierw agregacja, potem wnioskowanie</i>
COG	metoda środka ciężkości
MICOG <sub><math>\alpha</math></sub>	metoda modyfikowanego, indeksowanego środka ciężkości
HM	metoda wysokości
CAD	metoda uśrednionego centrum
TSK	system rozmyty <i>Takagi-Sugeno-Kanga</i>
GA	algorytm genetyczny
SGA	prosty algorytm genetyczny
RCGA	algorytm genetyczny o zmiennoprzecinkowym kodowaniu osobników
NP	klasa problemów NP-zupełnych
FS	poziom aktywacji
RBFN	sztuczne sieci neuronowe o radialnych funkcjach bazowych
GFS	system genetyczno-rozmyty
EFS	system ewolucyjno-rozmyty
GFRBS	system genetyczno-rozmyty oparty na bazie wiedzy
MIMO	system o wielu wejściach i wielu wyjściach
SIMO	system o jednym wejściu i wielu wyjściach



---

MISO	system o wielu wejściach i jednym wyjściu
DNF	normalna postać dusjunkcyjna
FCM	metoda rozmytych $c$ -średnich
RMSE	błąd średniokwadratowy
AVE	błąd średni



# 1

## Wprowadzenie

### 1.1 Tematyka rozprawy

Ludzie od początku dziejów zmagają się z problemami życia codziennego takimi jak rozpoznawanie twarzy, polowanie, ocena ryzyka, a wraz z rozwojem cywilizacji coraz bardziej skomplikowanymi problemami jak np. pilotaż samolotu czy parkowanie samochodu. Poprawne rozwiązania powstają w warunkach niepełnej, nieprecyzyjnej i często częściowo zafałszowanej informacji na temat obserwowanego zjawiska.

Zmuszenie maszyn cyfrowych do „inteligentnej” pracy było marzeniem wielu pokoleń naukowców, jednakże dopiero ostatnie dekady XX w. przyniosły rozwój technik informatycznych, pozwalających na opracowanie systemów wnioskowania przybliżonego w sposób analogiczny do tego, w jaki (jak przypuszczamy) wnioskuje człowiek, a więc z wykorzystaniem reguł warunkowych. Stworzenie systemu informatycznego zastępującego kompetencje człowieka-eksperta stało się faktem. Jednocześnie rozpoczął się trend ku doskonaleniu nowych technologii tak, aby były precyzyjniejsze i jednocześnie potrafiły lepiej uogólniać posiadaną wiedzę.

Do początku lat 60-tych ubiegłego wieku stosowano klasyczne metody sterowania i modelowania, które kładły zbyt duży nacisk na dokładność odzwierciedlania zjawiska, powodując tym samym ogromne kłopoty z opisaniem i modelowaniem skomplikowanej rzeczywistości otaczającego nas świata. Problemy te doprowadziły L. A. Zadeha w 1965 r. do opracowania i przedstawienia teorii zbiorów rozmytych [185], [186]. Wprowadzony aparat matematyczny pozwolił na stworzenie systemów rozmytych. Działanie tych systemów opiera się na bazie wiedzy w postaci rozmytych reguł warunkowych **if-then**. Przesłanki i konkluzje reguł są najczęściej określone za pomocą zbiorów rozmytych. Początkowo tworzenie bazy wiedzy opierało się na określaniu poziomu wiarygodności reguł rozmytych, które były budowane na podstawie wzorców dostarczonych przez eksperta-człowieka [187]. Kolejnym rozwinięciem były metody szablonowe, nad którymi pracowali m.in. B. Kosko [101], R. R. Yager i D. P. Filev [184]. W. Pedrycz [135] stosował teorię równań relacyjnych do znalezienia bazy wiedzy. Podejście to nakładało jednak ograniczenia w stosunku do zbiorów danych wejście-wyjście, które musiały spełniać równanie relacji, co automatycznie zawężało zakres zastosowań. W 1975 r. E. H. Mamdani i S. Assilian [116] opracowali i zastosowali w praktyce model lingwistycznego systemu wnioskowania rozmytego, zawierającego bazę reguł o przesłankach i konkluzjach w postaci zbiorów rozmytych. System stanowił podstawę badań [7], [8], [9], [31] i doczekał się wielu modyfikacji, w szczególności zmniejszających spory

nakład obliczeniowy wymagany do uzyskania rezultatu końcowego. Najpopularniejsze z nich to systemy *Larsena* oraz *Sugeno-Yakusawy*. Alternatywne rozwiązanie zaproponowali T. Takagi i M. Sugeno [169] oraz M. Sugeno i G. T. Kang [163] (system *TSK*). Ich system rozmyty, w przeciwieństwie do systemu rozmytego *Mamdaniego*, nie zawierał w konkluzjach reguł zbiorów rozmytych, co znacznie upraszcza jego implementację i zmniejsza nakład obliczeniowy niezbędny do uzyskania rezultatu na wyjściu systemu. W 1996 r. E. Czogała i J. Łęski [46], [48] zaprezentowali system rozmyty z parametryzowanymi konkluzjami, łączący cechy systemu *Mamdaniego* i *TSK*.

Równolegle z rozwojem metod wnioskowania rozmytego następował rozwój technik ewolucyjnych, opartych na mechanizmach naśladujących procesy zachodzące w naturalnych populacjach gatunków roślinnych i zwierzęcych. Pierwsze prace, wskazujące przydatność technik ewolucyjnych do rozwiązywania zadań optymalizacji pojawiły się już w późnych latach 50. ubiegłego wieku. W 1975 r. J. H. Holland opublikował monografię [85], w której opisał przydatność algorytmów genetycznych do wyszukiwania optimum globalnego w wielowymiarowych przestrzeniach. Równolegle z rozwojem algorytmów genetycznych, nad zastosowaniem technik ewolucyjnych do optymalizacji parametrycznej pracowali m.in. I. Rechenberg [148] i H.P. Schwefel [160]. Rozwój wariantów dla strategii ewolucyjnych oraz zastosowanie w algorytmach genetycznych reprezentacji zmiennoprzecinkowej spowodowało, że granice pomiędzy dotychczasowym podziałem na algorytmy genetyczne i strategie ewolucyjne zaczął się zacierać. Warty zaznaczenia jest podejście R. J. Kozy [102], [103], [104], [105] proponujące wykorzystanie algorytmów genetycznych do generowania programów. Dzięki takiemu podejściu stworzono system umożliwiający przeszukiwanie przestrzeni rozwiązań problemów. W ostatnich latach pojawiło się sporo pozycji poświęconych zarówno poszczególnym typom metod ewolucyjnych jak również generalizujące zebraną wiedzę [5], [70], [119]. Jakkolwiek nie istnieje zbyt wiele dowodów, które potwierdziłyby analitycznie przydatność metod ewolucyjnych i prawdziwość hipotez leżących u podstaw działania metod ewolucyjnych, obserwacja reguł rządzących światem naturalnym oraz funkcjonujące, praktyczne zastosowania wskazują na ich wysoką przydatność do rozwiązywania skomplikowanych problemów [131]. W szczególności dotyczy to klasy problemów NP-trudnych (ang. *Nondeterministic Polynomial*) [77]. Należy jednakże pamiętać, że metody ewolucyjne zastosowane w praktyce, gdzie np. wielkość populacji czy liczba pokoleń muszą być ograniczone ze względów implementacyjnych, dają rozwiązania suboptymalne, które najczęściej bliskie są doskonałości. Istotnym zagadnieniem wydaje się również złożoność obliczeniowa metod ewolucyjnych, a ich rozwój związany jest z coraz szerszym udostępnianiem i popularyzowaniem technologii obliczeń równoległych i rozproszonych. Niewątpliwą zaletą właściwie wykorzystanych technik ewolucyjnych jest szybkie dążenie w kierunku optimum globalnego i odporność na pozostawanie w minimach lokalnych, jednakże precyzyjne dobranie parametrów sterujących ich wykonaniem bywa żmudne.

Prezentując metody naśladowania natury nie sposób pominąć sztucznych sieci neuronowych, których konstrukcja jest wynikiem wielu lat badań nad procesami zachodzącymi w centralnym układzie nerwowym człowieka. W 1943 r. W. S. McCulloch i W. Pitts opublikowali matematyczny model neuronu [118]. Od tego czasu powstało sporo prac z dziedziny sztucznych sieci neuronowych. Informacje na ten temat można znaleźć m.in. w publikacjach R. Tadeusiewicza [166], [168]. Sztuczne sieci neuronowe, a w szczególności ich połączenia

z innymi technikami, takimi jak systemy rozmyte i obliczenia ewolucyjne, są szczególnie często wykorzystywane w zagadnieniach rozpoznawania obrazów [167].

Systemy rozmyte, metody ewolucyjne oraz sztuczne sieci neuronowe zalicza się do klasy tzw. obliczeń miękkich (ang. *Soft Computing*). Dostarczają one w różnych formach uniwersalnego aparatu do przetwarzania wiedzy, tolerancyjnego wobec braku precyzji, niekompletności oraz niepewności danych i stanowią dobre narzędzie do budowy systemów modelujących złożone procesy zachodzące w rzeczywistym świecie.

Systemy rozmyte udowodniły swoją przydatność w praktyce w zagadnieniach klasyfikacji, modelowania i sterowania [28], [56], [133], [134]. Tworzenie bazy wiedzy systemu jest intuicyjnie proste i nie wymaga skomplikowanego aparatu matematycznego. W pierwotnym podejściu, baza wiedzy systemu była ustalana przez eksperta-człowieka, na podstawie doświadczenia i wiedzy o modelowanym zjawisku. Pomimo dużej skuteczności takiego podejścia zauważono, że wygenerowana w ten sposób baza wiedzy często nosi znamiona sprzeczności, niespójności i subiektywności. Częstokroć też trudno jest znaleźć dobrego eksperta w rozpatrywanej dziedzinie, lub jest to wręcz niemożliwe. Problemy te doprowadziły w latach 90 ubiegłego wieku do prób zautomatyzowanego generowania bazy wiedzy na podstawie danych numerycznych opisujących zjawisko. Wynikiem był szereg prac, w których opisano hybrydowe połączenia systemów rozmytych, sieci neuronowych oraz metod ewolucyjnych. Celem nadrzędnym stało się stworzenie uniwersalnego systemu, który byłby zdolny do uczenia się na podstawie dostarczonych przykładów oraz do uogólniania nabytej wiedzy. Jedną z obiecujących gałęzi rozwoju stały się systemy neuronowe [166], [168] i neuronowo-rozmyte [48], [94], [111]. Inne podejście doprowadziło do utworzenia klasy systemów ewolucyjno-rozmytych [4], [35], [81], [133], [157]. W przeciwieństwie do typowych struktur sztucznych sieci neuronowych, metody ewolucyjne są znanymi optymalizatorami globalnymi i dlatego doskonale nadają się jako element budowy systemu hybrydowego, ewolucyjno-rozmytego. Kompletnie, przeglądowe kompendium wiedzy na temat bieżącego stanu dziedziny systemów ewolucyjno-rozmytych opublikowali w 2004 r. O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, i L. Magdalena [32]. Wskazują oni, iż większa część systemów ewolucyjno-rozmytych opiera się na zastosowaniu obliczeń ewolucyjnych do utworzenia bazy wiedzy, zarówno w zakresie automatycznego wyodrębniania reguł, jak i w lokalnym dostrajaniu parametrów tych reguł. Znane i stosowane warianty metod ewolucyjnych do zautomatyzowanego uzyskiwania bazy wiedzy na podstawie danych numerycznych to podejście Pittsburgh [162], Michigan [86] oraz uczenie iteracyjne [175]. Najbardziej obiecującym wydaje się wieloetapowe uczenie reguł zakodowanych w chromosomach, o kodowaniu zgodnym z podejściem Michigan, m.in. ze względu na znacznie mniejsze wymagania obliczeniowe w stosunku do podejścia Pittsburgh, oraz ze względu na elastyczność, pozwalającą na łatwe wprowadzanie reguł wygenerowanych innymi metodami lub podanych przez eksperta-człowieka. Od kilku lat systemy ewolucyjno-rozmyte znajdują swoje zastosowanie w praktycznych zagadnieniach sterowania. Przykładowo, P. Bonissone i wsp. [17] zaprezentowali system genetyczno-rozmyty zastosowany do zagadnienia sterowania prędkością pociągu towarowego, celem zminimalizowania naprężeń działających na złączeniach wagonów. Zagadnienie znalezienia optymalnej ścieżki prędkości względem czasu i energochłonności dla szybkich kolei rozważał w swojej pracy H. S. Hwang [88]. L. G. Damousis i wsp. [51] opracowali system genetyczno-rozmyty, prognozujący siłę wiatru i produkcję prądu dla elektrowni wiatrowych. Znane są również rozwiązania posługujące się algorytmami

genetycznymi lub strategiami ewolucyjnymi wyłącznie w celu poprawiania jakości (ang. *Tuning*) działania istniejących sterowników rozmytych, danych a priori przez eksperta lub uzyskany inną metodą. Przykładowe rozwiązanie tak postawionego problemu przedstawili m.in. F. Herrera i wsp. [80]. Praktyczne zastosowanie algorytmu genetycznego do poprawy jakości działania sterownika rozmytego, sterującego ogrzewaniem, klimatyzacją i wentylacją przedstawili R. Alcalá i wsp. [2].

Stale rosnąca liczba aplikacji ewolucyjno-rozmytych, jak również możliwości wynikające z hybrydowego połączenia systemów ewolucyjno-rozmytych oraz sztucznych sieci neuronowych pozwalają przypuszczać, że rozwój metod obliczeń miękkich nastąpi w kierunku tworzenia coraz bardziej wydajnych i precyzyjniejszych systemów wykazujących się zdolnościami do samodzielnego uczenia się i uogólniania posiadanej wiedzy. Jedną z tych metod z pewnością będą systemy ewolucyjno-rozmyte, rozumiane jako połączenie różnych modeli systemów rozmytych wraz ze strategiami ewolucyjnymi, programowaniem ewolucyjnym i algorytmami genetycznymi.

W niniejszej rozprawie przedstawiono hybrydowe połączenie strategii ewolucyjnych, algorytmów genetycznych i systemów rozmytych z parametryzowanymi konkluzjami. Metody ewolucyjne zastosowano do automatycznego wyodrębniania reguł **if-then** na podstawie numerycznych danych uczących a następnie do ich optymalizacji. Do wygenerowania bazy wiedzy systemu rozmytego, a następnie dostrojenia parametrów funkcji przynależności w przesłankach i konkluzjach reguł zastosowano strategię ewolucyjną oraz algorytm pokrywania. Do uproszczenia bazy wiedzy wykorzystano algorytm genetyczny z kodowaniem binarnym, a dla przypadków o niskiej liczbie reguł, proste wyszukiwanie wyczerpujące [77], [182]. Dla uzyskania parametrów konkluzji reguł tworzących system rozmyty wykorzystywano metodę najmniejszych kwadratów. Techniki ewolucyjne zostały zastosowane w celu uzyskania systemu rozmytego o jak najwyższej jakości.

Jakość systemu rozmytego oznacza precyzję odwzorowania wartości wejściowych w wyjście zarówno dla danych uczących jak i testujących. W niniejszej rozprawie jako miarę jakości systemu rozmytego przyjęto błąd średniokwadratowy zawarty pomiędzy odpowiedzią generowaną przez system rozmyty dla zadanych argumentów wejściowych i wzorcową wartością wyjściową, zawartą w zbiorach uczących i testowych. Z punktu widzenia użyteczności systemu rozmytego, istotne jest uzyskanie niskiej wartości błędu dla danych testowych, oznaczające wysoką zdolność systemu do uogólniania posiadanej wiedzy.

W ostatnich latach w literaturze pojawiło się wiele prac opisujących połączenia systemów ewolucyjnych i systemów rozmytych, jednakże w żadnej z nich nie wykorzystano systemu rozmytego z parametryzowanymi konkluzjami.

## 1.2 Teza rozprawy

Sformułowano następującą tezę niniejszej rozprawy:

**Możliwe jest poprawienie jakości systemu rozmytego z regułami if-then o parametryzowanych konkluzjach, poprzez wyodrębnianie ich na podstawie danych numerycznych, za pomocą wieloetapowej metody uczenia, z wykorzystaniem technik ewolucyjnych.**

W celu wykazania słuszności powyższej tezy rozwiązano opisane poniżej cele cząstkowe:

1. Opracowanie strategii ewolucyjnej wraz z operatorami genetycznymi, zastosowanej do iteracyjnego wyznaczania reguł systemu rozmytego z parametryzowanymi konkluzjami, metodą pokrywania zbioru uczącego.
2. Opracowanie algorytmu genetycznego wraz z operatorami do uproszczenia zbioru reguł uzyskanego w wyniku działania algorytmu pokrywania.
3. Opracowanie strategii ewolucyjnej wraz z operatorami genetycznymi do dostrojenia parametrów reguł uzyskanych w wyniku działania algorytmu genetycznego, z zastosowaniem podejścia Pittsburgh.
4. Przeprowadzenie doświadczeń z wykorzystaniem publicznie dostępnych baz danych z zakresu identyfikacji systemów oraz predykcji chaotycznych szeregów czasowych, celem porównania uzyskanych wyników z wynikami dostępnymi w literaturze.





## 2

# Systemy rozmyte

## 2.1 Wstęp do teorii zbiorów rozmytych

Klasyczne pojęcie zbiorów, w których każdy element przynależy lub nie do rozpatrywanego zbioru, jest wysoce nieelastyczne w zastosowaniach modelowania świata rzeczywistego. W szczególności problemem staje się zapisanie nieprecyzyjnej wiedzy wyrażanej w sposób werbalny. Przykładowo, jeżeli zdefiniujemy zbiór samochodów jadących szybko jako zbiór pojazdów, których prędkość jest większa lub równa 100km/h, samochody jadące z prędkością 99,9km/h nie będą to tego zbioru przynależeć. Można zauważyć, że jest to spore utrudnienie, gdyż różnica w prędkości jest praktycznie niedostrzegalna. Intuicyjne poszukiwanie sposobu rozwiązania tego i podobnych problemów doprowadziło do powstania teorii zbiorów rozmytych [185], w których zamiast wyraźnej granicy przynależności elementu do zbioru wprowadzono stopniowanie, określone za pomocą funkcji przynależności. Funkcja ta przyjmuje wartości od 0 (zupełnie nie należy do zbioru) do 1 (w pełni należy do zbioru). Wartości pomiędzy 0 i 1 określają stopień w jakim element przynależy do zbioru rozmytego. Tak określona definicja przynależności pozwala na stwierdzenie, iż zbiory klasyczne stanowią szczególny przypadek zbiorów rozmytych, w których funkcja przynależności jest zdefiniowana zerojedynkowo.

Zbiór rozmyty  $A$  określony na przestrzeni  $\mathbb{X}$  stanowi zbiór uporządkowanych par:

$$A = \{(x, \mu_A(x)) \mid x \in \mathbb{X}, \mu_A \in [0,1]\}, \quad (2.1)$$

gdzie  $\mu_A(x)$  oznacza funkcję przynależności. Istnieją różne sposoby reprezentacji zbiorów rozmytych, odmienne dla przestrzeni ciągłych i dyskretnych. Szerokie kompendium wiedzy na ten temat przedstawiono w [138] oraz [48]. Najczęściej wykorzystywana jest notacja zaproponowana w 1973 r. przez L.A. Zadeha [188]:

- Dyskretny zbiór rozmyty  $A$ :

$$A = \sum_{x \in \mathbb{X}} \frac{\mu_A(x)}{x}. \quad (2.2)$$

- Ciągły zbiór rozmyty  $A$ :

$$A = \int_{\mathbb{X}} \frac{\mu_A(x)}{x}. \quad (2.3)$$

Funkcję przynależności  $\mu_A(x)$  można interpretować jako stopień podobieństwa elementu  $x$  do wzorca reprezentowanego przez zbiór  $A$ , stopień preferencji, czy stopień niepewności [110]. Z szerokiego wachlarza funkcji przynależności opisywanych w literaturze np. [138] w niniejszej pracy wykorzystano *gaussowskie* i *trójkątne* funkcje przynależności [48], szczegółowo opisane w podrozdziale 2.2.1. W dalszej części niniejszej rozprawy zbiór rozmyty będzie utożsamiany z jego funkcją przynależności.

Zbiory rozmyte podlegają znacznej części operacji stanowiących naturalne rozszerzenie definicji znanych z teorii zbiorów klasycznych [145], jak również operacjom zdefiniowanym specjalnie dla nich.

Wygodnym aparatem matematycznym algebry zbiorów rozmytych, pozwalającym na definiowanie zarówno podstawowych, jak i złożonych działań są operatory *t-normy* i *t-konormy* (*s-normy*). Podlegają one szeregowi warunków i praw [48]. Poniżej zestawiono najczęściej wykorzystywane *t-normy* i odpowiadające im *s-normy*:

- Zadeha [185] (iloczyn i suma rozmyta):

$$\begin{aligned} x \star_T y &= \min(x, y) \\ x \star_S y &= \max(x, y) \end{aligned} \quad (2.4)$$

- Algebraiczna (iloczyn i suma algebraiczna):

$$\begin{aligned} x \star_T y &= x \cdot y \\ x \star_S y &= x + y - x \cdot y \end{aligned} \quad (2.5)$$

- Łukasiewicza:

$$\begin{aligned} x \star_T y &= \min(x + y - 1, 0) \\ x \star_S y &= \max(x + y, 1) \end{aligned} \quad (2.6)$$

- Drastyczna:

$$\begin{aligned} x \star_T y &= \begin{cases} \min(x, y) & \text{gdy } \max(x, y) = 1, \\ 0 & \text{w przeciwnym wypadku} \end{cases} \\ x \star_S y &= \begin{cases} \max(x, y) & \text{gdy } \min(x, y) = 1, \\ 1 & \text{w przeciwnym wypadku} \end{cases} \end{aligned} \quad (2.7)$$

W praktyce często wykorzystuje się operatory *Zadeha* i *Algebraiczne*, ze względu na mały nakład obliczeniowy i prostotę implementacji. Inne operatory wraz z ich klasyfikacją można znaleźć w pracach [10], [48], [50], [138].

Pozostałe definicje związane ze zbiorami rozmytymi, w szczególności *nośnik*, *jądro*, *singleton*,  $\alpha$ -*przekrój* oraz definicje podstawowych operacji na zbiorach rozmytych można znaleźć w pracy [138]. W dalszych rozważaniach przydatna będzie znajomość operatorów rozszerzenia cylindrycznego oraz projekcji.

Rozszerzenie cylindryczne zbioru  $A$ , zdefiniowanego na przestrzeni ciągłej  $\mathbb{X}$  na przestrzeń  $\mathbb{X} \times \mathbb{Y}$  daje w wyniku zbiór rozmyty  $Ce(A)$ , zdefiniowany następująco [48]:

$$Ce(A) = \int_{\mathbb{X} \times \mathbb{Y}} \frac{\mu_A(x)}{(x, y)}, \quad (2.8)$$

czyli zbiór wszystkich par  $(x, y) \in \mathbb{X} \times \mathbb{Y}$ , dla których funkcja przynależności przyjmuje wartości  $\mu_A(x)$ . Analogiczna definicja obowiązuje również dla przestrzeni wielowymiarowej, co ma miejsce w przypadku większości systemów rozmytych, jak również dotyczy zbiorów dyskretnych.

Projekcja (rzutowanie) zbioru ciągłego  $B$  określonego na przestrzeni  $\mathbb{X} \times \mathbb{Y}$  na przestrzeń  $\mathbb{X}$  jest zbiór rozmyty  $Proj_{\mathbb{X}}(B)$ , zdefiniowany następująco [48]:

$$Proj_{\mathbb{X}}(B) = \int_{\mathbb{X}} \frac{\sup_y \mu_B(x, y)}{x}, \quad (2.9)$$

czyli zbiór wartości  $x$ , dla których funkcja przynależności przyjmuje największą z wartości względem każdego wymiaru. Analogicznie jak to ma miejsce w przypadku rozszerzenia cylindrycznego, podobna definicja obowiązuje dla przestrzeni wielowymiarowej oraz dla zbiorów dyskretnych.

### 2.1.1 Relacja rozmyta

Relacja rozmyta  $R$  pomiędzy przestrzeniami  $\mathbb{X}$  i  $\mathbb{Y}$  jest definiowana jako podzbiór rozmyty, określony na iloczynie kartezjańskim  $\mathbb{X} \times \mathbb{Y}$  i stanowi zbiór uporządkowanych par:

$$R \subseteq \mathbb{X} \times \mathbb{Y} = \{(x, y) \mid x \in \mathbb{X}, y \in \mathbb{Y}\}, \quad (2.10)$$

$$\bigvee_{x \in \mathbb{X}} \bigvee_{y \in \mathbb{Y}} R = \{(x, y, \mu_R(x, y)) \mid x \in \mathbb{X}, y \in \mathbb{Y}, \mu_R \in [0, 1]\}, \quad (2.11)$$

gdzie  $\mu_R: \mathbb{X} \times \mathbb{Y} \rightarrow [0, 1]$  jest funkcją przynależności relacji  $R$ , która każdej parze  $(x, y)$  przyporządkowuje wartość  $\mu_R(x, y)$  reprezentującą miarę powiązania pomiędzy elementami  $x$  i  $y$ . Tak zdefiniowaną relację dwuargumentową można uogólnić na  $n$ -argumentową relację rozmytą [48].

### 2.1.2 Zmienna lingwistyczna

Zmienna lingwistyczna jest jednym z podstawowych pojęć teorii zbiorów rozmytych i stanowi wielkość reprezentowaną przez wartości będące słowami języka naturalnego (wartości lingwistyczne), zdefiniowanymi za pomocą zbiorów rozmytych. Przykładem takiej zmiennej może być prędkość samochodu, która może przyjmować następujące wartości (terminy): „mała”, „średnia”, „duża”. Wnioskowanie rozmyte niejednokrotnie wymusza stosowanie modyfikatorów istniejących charakterystyk zmiennej lingwistycznej, polegających na zastosowaniu operacji na istniejących funkcjach przynależności zmiennej. Przykładowo

operator „bardzo” (prędkość „bardzo duża”) może odpowiadać zastosowaniu funkcji kwadratowej w stosunku do funkcji przynależności określającej termin „duża”. Formalnie zmienną lingwistyczną można zdefiniować jako piątkę [189], [190], [191]:

$$X = (\mathcal{N}, \mathcal{T}(\mathcal{G}), \mathbb{X}, \mathcal{G}, \mathcal{S}), \quad (2.12)$$

gdzie  $\mathcal{N}$ -nazwa zmiennej,  $\mathcal{T}(\mathcal{G})$ -zbiór terminów lingwistycznych,  $\mathcal{G}$ -zbiór reguł syntaktycznych,  $\mathcal{S}$ -zbiór reguł semantycznych definiujących znaczenie każdego z terminów.

W literaturze można spotkać definicję zbiorów rozmytych typu II [138], których elementy nie są określone przez pojedynczą funkcję przynależności, lecz stanowią zbiór par  $(\mu_{AII}(x), x)$ , gdzie  $\mu_{AII}(x) \in \mathcal{T}(\mathcal{G})$  oznacza stopień przynależności elementu  $x$ , stanowiący w istocie jeden z terminów lingwistycznych zdefiniowanych dla zbioru  $A$ :

$$A_{II} = \sum_{x \in \mathbb{X}, l \in \mathcal{T}(\mathcal{G})} \frac{l}{x}, \quad (2.13)$$

gdzie  $A_{II}$  – zbiór rozmyty typu II,  $l$ -termin lingwistyczny spośród terminów  $\mathcal{T}(\mathcal{G})$  zdefiniowanych dla zbioru rozmytego  $A$ .

## 2.2 Interpretacja reguł if-then

Elementarne rozmyte wyrażenie warunkowe w formie implikacji (regułę warunkową) można przedstawić w postaci:

$$\text{jeżeli } (X \text{ jest } A) \text{ to } (Y \text{ jest } B). \quad (2.14)$$

Wyrażenie  $(X \text{ jest } A)$  określa przesłankę reguły,  $(Y \text{ jest } B)$  jej konkluzję. Zmienne lingwistyczne  $X$  i  $Y$  przyjmują wartości ze zbioru etykiet zdefiniowanych na przestrzeni  $\mathbb{X}$ .  $A$  i  $B$  są lingwistycznymi terminami reprezentującymi zbiory rozmyte, dane funkcjami przynależności odpowiednio  $\mu_A(x)$  i  $\mu_B(x)$ . Rozmyte wyrażenie warunkowe pozwala na określenie zależności pomiędzy zmiennymi lingwistycznymi. Z prostych wyrażeń warunkowych, takich jak np. powyższa przesłanka można tworzyć złożone wyrażenia warunkowe, które mogą być przedstawione zarówno w formie koniunkcyjnej:

$$(X_1 \text{ jest } A_1) \text{ i } (X_2 \text{ jest } A_2), \quad (2.15)$$

jak i dysjunkcyjnej:

$$(X_1 \text{ jest } A_1) \text{ lub } (X_2 \text{ jest } A_2), \quad (2.16)$$

gdzie  $X_1, X_2$  - zmienne lingwistyczne o wartościach zdefiniowanych za pomocą odpowiednio  $A_1, A_2$  na przestrzeniach  $\mathbb{X}_1$  i  $\mathbb{X}_2$ .

Wartość logiczną złożonego wyrażenia rozmytego można wyznaczyć stosując odpowiednio dla formy koniunkcyjnej:

$$\mu_{A_1}(x_1) \star_T \mu_{A_2}(x_2), \quad (2.17)$$

oraz dla formy dysjunkcyjnej:

$$\mu_{A_1}(x_1) \star_s \mu_{A_2}(x_2). \quad (2.18)$$

Poniżej przedstawiono ogólną postać rozmytego wyrażenia warunkowego, zdefiniowaną jako implikacja złożonych wyrażeń rozmytych. Tworzy ona regułę  $R_{MIMO}$  (ang. *Multiple-Input-Multiple-Output*) o wielu wejściach i wielu wyjściach, którą można zapisać w następującej postaci koniunkcyjnej:

$$R_{MIMO} : \begin{array}{l} \text{jeżeli } (X_1 \text{ jest } A_1) \text{ i } (X_2 \text{ jest } A_2) \text{ i } \dots \text{ i } (X_N \text{ jest } A_N) \\ \text{to } (Y_1 \text{ jest } B_1), (Y_2 \text{ jest } B_2), \dots, (Y_P \text{ jest } B_P) \end{array}, \quad (2.19)$$

i dysjunkcyjnej:

$$R_{MIMO} : \begin{array}{l} \text{jeżeli } (X_1 \text{ jest } A_1) \text{ lub } (X_2 \text{ jest } A_2) \text{ lub } \dots \text{ lub } (X_N \text{ jest } A_N) \\ \text{to } (Y_1 \text{ jest } B_1), (Y_2 \text{ jest } B_2), \dots, (Y_P \text{ jest } B_P) \end{array}, \quad (2.20)$$

gdzie  $X_1, X_2, \dots, X_N$  – zmienne lingwistyczne wejściowe,  $Y_1, Y_2, \dots, Y_P$  – zmienne lingwistyczne wyjściowe,  $A_1, A_2, \dots, A_N$  – wartości lingwistyczne wejściowe,  $Y_1, Y_2, \dots, Y_N$  – wartości lingwistyczne wyjściowe, zdefiniowane odpowiednio na  $\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_N$  i  $\mathbb{Y}_1, \mathbb{Y}_2, \dots, \mathbb{Y}_P$ .

W dalszej części pracy stosowane będą wyłącznie reguły o postaci koniunkcyjnej.

Każdą regułę typu MIMO, o postaci koniunkcyjnej można dekomponować do postaci zbioru reguł o wielu wejściach i jednym wyjściu (ang. *Multiple-Input-Single-Output*):

$$\begin{aligned} R_{MISO(Y_1)} &: \text{jeżeli } \bigwedge_{n=1}^N (X_n \text{ jest } A_n) \text{ to } (Y_1 \text{ jest } B_1) \\ R_{MISO(Y_2)} &: \text{jeżeli } \bigwedge_{n=1}^N (X_n \text{ jest } A_n) \text{ to } (Y_2 \text{ jest } B_2) \\ &\dots \\ R_{MISO(Y_P)} &: \text{jeżeli } \bigwedge_{n=1}^N (X_n \text{ jest } A_n) \text{ to } (Y_P \text{ jest } B_P) \end{aligned} \quad (2.21)$$

W dalszej części niniejszej rozprawy przyjęto zastosowanie reguł typu MISO. W związku z tym w konkluzjach reguł zostanie pominięty indeks  $p$  określający numer wyjścia:

$$R = R_{MISO} : \text{jeżeli } \bigwedge_{n=1}^N (X_n \text{ jest } A_n) \text{ to } (Y \text{ jest } B). \quad (2.22)$$

Powyższa reguła definiuje tzw. kanoniczną postać reguły rozmytej [48]. Odpowiada jej zapis pseudowektorowy:

$$R : \text{jeżeli } (\underline{X} \text{ jest } \underline{A}) \text{ to } (Y \text{ jest } B). \quad (2.23)$$

Rozmytą relację, zdefiniowaną na iloczynie kartezjańskim  $\mathbb{X}_1 \times \mathbb{X}_2 \times \dots \times \mathbb{X}_N \times \mathbb{Y}$  (wektorowo  $\underline{\mathbb{X}} \times \mathbb{Y}$ ), odpowiadającą powyższej formule można przedstawić następująco:

$$R = ((A_1 \times A_2 \times \dots \times A_N) \Rightarrow B). \quad (2.24)$$

Funkcję przynależności takiej relacji można zdefiniować następująco:

$$\mu_R(x_1, x_2, \dots, x_N, y) = \mu_{A_1 \times A_2 \times \dots \times A_N}(x_1, x_2, \dots, x_N) \Rightarrow \mu_B(y), \quad (2.25)$$

lub w zapisie pseudowektorowym:

$$R = (\underline{A} \Rightarrow B), \quad (2.26)$$

gdzie  $\underline{A} = A_1 \times A_2 \times \dots \times A_N$ . Funkcję przynależności powyższej relacji możemy również wyrazić poprzez zastosowanie zapisu wektorowego:

$$\mu_R(\underline{x}, y) = (\mu_{\underline{A}}(\underline{x}) \Rightarrow \mu_B(y)), \quad (2.27)$$

gdzie  $\underline{x} = [x_1, x_2, \dots, x_N]^T$ .

Znane w logice rozmytej metody wyrażania reguł **if-then** to metoda koniunkcyjna (tzw. metoda konstruktywna) i metoda logiczna (tzw. metoda destruktywna) [47], [48], [56], [184].

W przypadku metody koniunkcyjnej, reguły zdefiniowane za pomocą rozmytych relacji należy interpretować jako iloczyn zbiorów rozmytych  $\underline{A}$  i  $B$ . Funkcja przynależności relacji dana jest wzorem:

$$\mu_R(\underline{x}, y) = \mu_{\underline{A}}(\underline{x}) \star_{Tc} \mu_B(y). \quad (2.28)$$

W przypadku wielowartościowych reguł, łącznik  $\mathbf{i}$  modelowany jest również za pomocą operatora  $t$ -normy:

$$\mu_R(\underline{x}, y) = \mu_{\underline{A}}(\underline{x}) \star_T \mu_B(y) = \mu_{A_1}(x_1) \star_{T_1} \mu_{A_2}(x_2) \star_{T_1} \dots \star_{T_1} \mu_{A_N}(x_N) \star_{Tc} \mu_B(y). \quad (2.29)$$

Do najczęściej stosowanych  $t$ -norm należą operator minimum (wnioskowanie Mamdaniego [48]) oraz iloczyn algebraiczny (wnioskowanie Larsena [48]). Wyrażenie  $\mu_{\underline{A}}(\underline{x})$  zdefiniowane jako:

$$\mu_{\underline{A}}(\underline{x}) = \mu_{A_1}(x_1) \star_{T_1} \mu_{A_2}(x_2) \star_{T_1} \dots \star_{T_1} \mu_{A_N}(x_N), \quad (2.30)$$

określa poziom aktywacji reguły ( $FS$ , ang. *Firing Strength*) [48].

W przypadku interpretacji logicznej reguł rozmytych wykorzystuje się implikacje rozmyte, analogicznie jak to ma miejsce w przypadku logiki dwuwartościowej [145]. Należy jednak zauważyć, iż równoważne postacie zapisu implikacji **jeżeli  $A$  to  $B$ :  $(A \Rightarrow B)$** , czyli **(nie( $A$ ) lub ( $B$ ))** oraz **(( $A$  i  $B$ ) lub nie( $A$ ))**, obowiązujące w przypadku logiki dwuwartościowej, ze względu na odmienną operacji sumy, iloczynu i dopełnienia zbiorów rozmytych, w przypadku implikacji rozmytej nie musi obowiązywać [43].

Tabela 2.1: Tabela prawdy dla implikacji klasycznej  $A \Rightarrow B$ 

$\mu_A$	$\mu_B$	$\mu_{A \Rightarrow B}$
0	0	1
0	1	1
1	0	0
1	1	1

Pierwowzorem implikacji rozmytych były implikacje stworzone dla potrzeb logiki wielowartościowej przez J. Łukasiewicza [114]. W określeniu cech implikacji rozmytej pomocna jest jej definicja aksjomatyczna wprowadzona przez J. C. Fodora [59], [60], [61]. Implikacja rozważana jest jako funkcja  $I : [0,1]^2 \rightarrow [0,1]$  posiadająca pewne cechy, które musi spełnić oraz cechy, które może spełniać [58]. Na tej podstawie implikacje zostały sklasyfikowane jako przynależące do jednej z grup: S-implikacje, R-implikacje, QL-implikacje.

Najczęściej spotykane implikacje zostały zestawione poniżej:

- Implikacja Fodora:

$$I(\mu_A(x), \mu_B(y)) = \begin{cases} 1, & \text{jeżeli } \mu_A(x) \leq \mu_B(y) \\ \max(1 - \mu_A(x), \mu_B(y)) & \text{w przeciwnym wypadku} \end{cases} \quad (2.31)$$

- Implikacja Gödela:

$$I(\mu_A(x), \mu_B(y)) = \begin{cases} 1, & \text{jeżeli } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y) & \text{w przeciwnym wypadku} \end{cases} \quad (2.32)$$

- Implikacja Goguena:

$$I(\mu_A(x), \mu_B(y)) = \min\left(\frac{\mu_A(x)}{\mu_B(y)}, 1\right). \quad (2.33)$$

- Implikacja Kleenego-Dienes:

$$I(\mu_A(x), \mu_B(y)) = \max(1 - \mu_A(x), \mu_B(y)). \quad (2.34)$$

- Implikacja Łukasiewicza:

$$I(\mu_A(x), \mu_B(y)) = \min(1 - \mu_A(x) + \mu_B(y), 1). \quad (2.35)$$

- Implikacja Reichenbacha:

$$I(\mu_A(x), \mu_B(y)) = 1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y). \quad (2.36)$$

- Implikacja Reschera:

$$I(\mu_A(x), \mu_B(y)) = \begin{cases} 1, & \text{jeżeli } \mu_A(x) \leq \mu_B(y) \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad (2.37)$$

- Implikacja Zadeha:

$$I(\mu_A(x), \mu_B(y)) = \max(1 - \mu_A(x), \min(\mu_A(x), \mu_B(y))). \quad (2.38)$$

Graficzną prezentację powyższych implikacji oraz ocenę pseudometrycznej odległości pomiędzy operatorami wybranych implikacji można znaleźć w [48].

### 2.2.1 Wnioskowanie z wykorzystaniem rozmytej bazy reguł

Podstawowym elementem budowy systemu rozmytego jest baza wiedzy. Niniejsza rozprawa wykorzystuje wyłącznie systemy rozmyte, w których baza wiedzy zapisana jest w postaci rozmytych reguł **if-then**. Pojedyncza reguła stanowi swoistego eksperta, który wpływa w sposób określony poprzez parametry rozmytych zbiorów w konkluzji na zachowanie systemu w odpowiadającym mu zakresie wiedzy, zdefiniowanym poprzez zbiór rozmyty w przesłankach. Należy również pamiętać, iż istotne znaczenie ma wybór metody wnioskowania i co jest z tym związane, zastosowanie różnych operatorów. Szczegółowy opis znanych i stosowanych modeli systemów rozmytych znajduje się w rozdziale 3.

Baza reguł  $\mathcal{R}$  systemu rozmytego, składa się z  $L$  reguł typu MIMO:

$$\mathcal{R} = \{R_{MIMO}^{(l)}\}_{l=1}^L = \left\{ \text{jeżeli } \bigwedge_{n=1}^N (X_n \text{ jest } A_n^{(l)}) \text{ to } \{Y_p \text{ jest } B_p^{(l)}\}_{p=1}^P \right\}_{l=1}^L. \quad (2.39)$$

Jak wykazano wcześniej, każdą regułę typu MIMO można przedstawić jako zbiór reguł MISO, zatem bazę reguł można przedstawić w postaci:

$$\mathcal{R} = \left\{ \{R_{MISO}^{(l)(p)}\}_{p=1}^P \right\}_{l=1}^L = \{R_{MISO}^{(i)}\}_{i=1}^I. \quad (2.40)$$

gdzie  $i=1, \dots, L \cdot P$ .

Baza reguł powinna być kompletna oraz spójna [43] lub wg [56] zupełna, zgodna, ciągła oraz podlegająca interakcji. Wymagania wobec bazy można określić intuicyjnie: kompletność jest cechą oznaczającą, iż system potrafi zareagować (odpowiedzieć) na każdą kombinację danych wejściowych, spójność oznacza bazę reguł sformułowaną w sposób jednoznaczny, bez redundancji i sprzeczności (reguł o tych samych przesłankach, prowadzących do różnych konkluzji). Reguły znajdujące się w bazie wiedzy można interpretować jako zbiór zapisów związków przyczynowo-skutkowych takich, jakimi posługuje się człowiek. Prawidłowo wyznaczona baza wiedzy oraz dobrze dobrany mechanizm wnioskowania stanowi klucz do opracowania precyzyjnego systemu rozmytego, zdolnego do uogólniania wiedzy.

Wnioskowanie rozmyte jest analogiczne do wnioskowania w logice klasycznej, w której wykorzystuje się tautologię. Najczęściej stosowane tautologie, to [48]:



1. Modus Ponens (sposób potwierdzający potwierdzeniem): „Jeżeli implikacja  $A \Rightarrow B$  jest prawdziwa i prawdziwa jest przesłanka  $A$ , to w wyniku prawdziwy jest wniosek  $B$ ”. Symbolicznie prawo to można zapisać następująco:

$$(A \wedge (A \Rightarrow B)) \Rightarrow B, \quad (2.41)$$

wraz z odpowiadającym mu schematem wnioskowania:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad A \\ \text{Przesłanka 2 (reguła):} \quad A \Rightarrow B \\ \hline \text{Wniosek:} \quad B \end{array}$$

2. Modus Tollens (sposób zaprzeczający zaprzeczeniem): „Jeżeli implikacja  $A \Rightarrow B$  jest prawdziwa i nieprawdziwy jest wniosek  $B$ , to nieprawdziwa jest przesłanka  $A$ ”. Symbolicznie prawo to można zapisać następująco:

$$(\neg B \wedge (A \Rightarrow B)) \Rightarrow \neg A, \quad (2.42)$$

wraz z odpowiadającym mu schematem wnioskowania:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad \neg B \\ \text{Przesłanka 2 (reguła):} \quad A \Rightarrow B \\ \hline \text{Wniosek:} \quad \neg A \end{array}$$

3. Warunkowy sylogizm (przechodniość implikacji): „Jeżeli implikacja  $A \Rightarrow B$  jest prawdziwa i implikacja  $B \Rightarrow C$  jest prawdziwa, to prawdziwa jest również implikacja  $A \Rightarrow C$ ”. Symbolicznie prawo to można zapisać następująco:

$$((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C), \quad (2.43)$$

wraz z odpowiadającym mu schematem wnioskowania:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad A \Rightarrow B \\ \text{Przesłanka 2 (reguła):} \quad B \Rightarrow C \\ \hline \text{Wniosek:} \quad A \Rightarrow C \end{array}$$

W logice klasycznej, dwuwartościowej, występuje zawsze całkowita zgodność lub niezgodność pomiędzy faktem i przesłanką implikacji. Takie założenie nie jest prawdą w przypadku logiki rozmytej, gdzie możemy jedynie stwierdzić, że fakt jest prawdziwy i zbliżony do przesłanki implikacji rozmytej. Prowadzi to do definicji uogólnionych reguł wnioskowania [48]:

1. Uogólniony Modus Ponens: „Jeżeli implikacja  $A \Rightarrow B$  jest prawdziwa i prawdziwa jest przesłanka  $A'$  zbliżona, choć niekoniecznie równa  $A$ , to w wyniku prawdziwy jest wniosek  $B'$  zbliżony, choć niekoniecznie równy  $B$ ”. Odpowiada temu następujący schemat wnioskowania:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad A' \\ \text{Przesłanka 2 (reguła):} \quad A \Rightarrow B \\ \hline \text{Wniosek:} \quad B' \end{array}$$

Rozszerzając powyższy schemat na przypadek reguły rozmytej otrzymujemy:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad (X \text{ jest } A') \\ \text{Przesłanka 2 (reguła):} \quad \text{jeżeli } (X \text{ jest } A) \text{ to } (Y \text{ jest } B) \\ \hline \text{Wniosek:} \quad (Y \text{ jest } B') \end{array}$$

gdzie  $A, A', B, B'$  – lingwistyczne wartości definiowane za pomocą zbiorów rozmytych na przestrzeniach odpowiednio  $\mathbb{X}$  i  $\mathbb{Y}$ ,  $X$  i  $Y$  – zmienne lingwistyczne.

2. Uogólniony Modus Tollens: „Jeżeli implikacja  $A \Rightarrow B$  jest prawdziwa i prawie nieprawdziwy jest wniosek  $B'$  (bliski, choć niekoniecznie równy  $\neg B$ ), to nieprawdziwa jest przesłanka  $A'$  (bliska, choć niekoniecznie równa  $\neg A$ )”. Odpowiada temu następujący schemat wnioskowania:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad B' \\ \text{Przesłanka 2 (reguła):} \quad A \Rightarrow B \\ \hline \text{Wniosek:} \quad A' \end{array}$$

Rozszerzając powyższy schemat na przypadek reguły rozmytej otrzymujemy:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad (Y \text{ jest } B') \\ \text{Przesłanka 2 (reguła):} \quad \text{jeżeli } (X \text{ jest } A) \text{ to } (Y \text{ jest } B) \\ \hline \text{Wniosek:} \quad (X \text{ jest } A') \end{array}$$

gdzie  $A, A', B, B'$  – lingwistyczne wartości definiowane za pomocą zbiorów rozmytych na przestrzeniach odpowiednio  $\mathbb{X}$  i  $\mathbb{Y}$ ,  $X$  i  $Y$  – zmienne lingwistyczne.

3. Uogólniony warunkowy sylogizm: „Jeżeli implikacja  $A \Rightarrow B$  jest prawdziwa i implikacja  $B' \Rightarrow C$  jest prawdziwa, oraz  $B'$  jest bliskie  $B$ , to prawdziwa jest również

implikacja  $A \Rightarrow C'$ , gdzie  $C'$  jest bliskie  $C''$ . Odpowiada temu następujący schemat wnioskowania:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad \neg B \\ \text{Przesłanka 2 (reguła):} \quad A \Rightarrow B \\ \hline \text{Wniosek:} \quad \neg A \end{array}$$

Rozszerzając powyższy schemat na przypadek reguły rozmytej, otrzymujemy:

$$\begin{array}{l} \text{Przesłanka 1 (fakt):} \quad \textit{jeżeli} (X \text{ jest } A) \text{ to } (Y \text{ jest } B) \\ \text{Przesłanka 2 (reguła):} \quad \textit{jeżeli} (Y \text{ jest } B') \text{ to } (Z \text{ jest } C) \\ \hline \text{Wniosek:} \quad \textit{jeżeli} (X \text{ jest } A) \text{ to } (Z \text{ jest } C') \end{array}$$

gdzie  $A, A', B, B', C, C'$  – lingwistyczne wartości definiowane za pomocą zbiorów rozmytych na przestrzeniach odpowiednio  $\mathbb{X}$ ,  $\mathbb{Y}$  i  $\mathbb{Z}$ , a  $X, Y, Z$  – zmienne lingwistyczne.

Pomocną operacją, niezbędną do zdefiniowania *złożeniowej reguły wnioskowania* jest operator złożenia *sup-T-norma* relacji  $R_1$  i  $R_2$  [48]:

$$R_1 \circ R_2 = \left\{ ((x, z), \sup_{y \in \mathbb{Y}} (\mu_{R_1}(x, y) \star_T \mu_{R_2}(y, z))) \mid x \in \mathbb{X}, y \in \mathbb{Y}, z \in \mathbb{Z} \right\}, \quad (2.44)$$

gdzie  $R_1, R_2$  – relacje rozmyte, określone odpowiednio na  $\mathbb{X} \times \mathbb{Y}$  oraz  $\mathbb{Y} \times \mathbb{Z}$ .

Celem pośrednim każdego wnioskowania rozmytego jest znalezienie funkcji przynależności określającej zbiór w konkluzji reguły R. W tym celu wejściowy zbiór  $A'(x)$  przesłanki, zdefiniowany na przestrzeni  $\mathbb{X}$ , należy poddać rozszerzeniu cylindrycznemu:

$$\mu_{Ce(A')}(x, y) = \mu_A(x), \quad (2.45)$$

a następnie wykonać operację przecięcia zbiorów  $Ce(A')$  i  $R(x, y)$  z wykorzystaniem operatora *t-normy*:

$$\mu_{Ce(A') \cap R}(x, y) = \mu_{Ce(A')}(x, y) \star_T \mu_R(x, y) = \mu_A(x, y) \star_T \mu_R(x, y). \quad (2.46)$$

Ostatecznie, wynikową funkcję przynależności zbioru konkluzji  $\mu_{B'}(y)$  można obliczyć, jako:

$$\mu_{B'}(y) = \sup_{x \in \mathbb{X}} (\mu_{A'}(x) \star_T \mu_R(x, y)). \quad (2.47)$$

Powyższe równania można uogólnić na wielowymiarową przestrzeń rozważań.

W przypadku, gdy wykorzystywana jest kanoniczna postać reguły rozmytej **if-then**, uogólniony, rozmyty schemat wnioskowania Modus Ponens, dla  $R^{(i)}$ -tej relacji rozmytej,

reprezentującej  $i$ -tą regułę bazy wiedzy, przy założeniu wielowymiarowej przestrzeni wejściowej  $\underline{A}' \in \underline{\mathbb{X}}$  przyjmuje postać [48]:

$$B^{(i)'} = \underline{A}' \circ R^{(i)}, \quad (2.48)$$

o funkcji przynależności:

$$\mu_{B'}^{(i)}(y) = \sup_{x \in \mathbb{X}} (\mu_{\underline{A}'}^{(i)}(x) \star_T \mu_{R^{(i)}}(x, y)), \quad (2.49)$$

gdzie  $\underline{A}' = A_1' \times A_2' \times \dots \times A_N'$ . Analogicznie można przedstawić uogólniony, rozmyty schemat wnioskowania Modus Tollens:

$$\underline{A}' = R^{(i)} \circ B^{(i)'}, \quad (2.50)$$

o funkcji przynależności:

$$\mu_{\underline{A}'}^{(i)}(x) = \sup_{y \in \mathbb{Y}} (\mu_{R^{(i)}}(x, y) \star_T \mu_{B'}^{(i)}(y)), \quad (2.51)$$

oraz rozmyty, warunkowy sylogizm:

$$R_3 = R_1 \circ R_2, \quad (2.52)$$

gdzie  $R_1, R_2, R_3$  - rozmyte relacje, określone odpowiednio na  $\mathbb{X} \times \mathbb{Y}, \mathbb{Y} \times \mathbb{Z}, \mathbb{X} \times \mathbb{Z}$ ,

o funkcji przynależności (dla  $i$ -tej reguły):

$$\mu_{R_3}^{(i)}(x, z) = \sup_{y \in \mathbb{Y}} (\mu_{R_1}^{(i)}(x, y) \star_T \mu_{R_2}^{(i)}(y, z)). \quad (2.53)$$

Należy zaznaczyć, że generalnym warunkiem stosowania uogólnionej tautologii jest posiadanie przez mechanizm wnioskujący właściwości ekstrapolacyjnych, co ma miejsce w opisywanym przypadku, ze względu na charakter stosowanych operatorów.

Wynikowa funkcja przynależności powstaje na podstawie bazy wiedzy systemu, z użyciem jednej z dwóch wykorzystywanych metod [48]:

- Złożenie FATI (ang. *First Agregate Then Inference*), gdzie najpierw łączy się wszystkie reguły bazy wiedzy, a następnie aktywuje je wejściem rozmytym.
- Złożenie FITA (ang. *First Inference Then Aggregate*), gdzie najpierw każda reguła jest aktywowana wejściem rozmytym, a następnie poszczególne wyniki są łączone, celem stworzenia wynikowego zbioru przynależności końcowej konkluzji.

W zależności od zastosowanego procesu, łączenie reguł (dla FATI) lub łączenie konkluzji poszczególnych reguł (dla FITA) zwane jest agregacją:

$$R = \bigoplus_{i=1}^I R^{(i)} = \bigoplus (R^{(1)}, R^{(2)}, \dots, R^{(I)}), \quad (2.54)$$

gdzie  $\bigoplus: [0,1]^I \rightarrow [0,1]$  określa operator agregacji, w wyniku którego otrzymujemy pojedynczą relację  $R$ . W praktyce stosowane są różne operatory agregacji, m.in. iloczyn zbiorów

rozmytych (*t-norma*), suma zbiorów rozmytych (*s-norma*), operatory średniej, w szczególności uogólniona postać tego operatora [19]:

$$\bigoplus_{i=1}^I (\alpha) x_i = \left( \frac{1}{I} \sum_{i=1}^I x_i^\alpha \right)^{\frac{1}{\alpha}}, \quad (2.55)$$

gdzie  $\alpha$ -parametr określający rodzaj średniej. W zależności od przyjętej wartości tego parametru otrzymujemy następujące średnie [23]:

- Dla  $\alpha \rightarrow -\infty$  – średnią koniunkcyjną:

$$\lim_{\alpha \rightarrow -\infty} \left( \bigoplus_{i=1}^I (\alpha) x_i \right) = \min_{i=1} x_i. \quad (2.56)$$

- Dla  $\alpha \rightarrow -1$  – średnią harmoniczną:

$$\bigoplus_{i=1}^I (-1) x_i = \left( \frac{1}{I} \sum_{i=1}^I \frac{1}{x_i} \right)^{-1}. \quad (2.57)$$

- Dla  $\alpha \rightarrow 0$  – średnią geometryczną:

$$\bigoplus_{i=1}^I (0) x_i = \left( \prod_{i=1}^I x_i \right)^{\frac{1}{I}}. \quad (2.58)$$

- Dla  $\alpha \rightarrow 1$  – średnią arytmetyczną:

$$\bigoplus_{i=1}^I (1) x_i = \frac{1}{I} \sum_{i=1}^I x_i. \quad (2.59)$$

- Dla  $\alpha \rightarrow 2$  – średnią kwadratową:

$$\bigoplus_{i=1}^I (2) x_i = \sqrt{\frac{1}{I} \sum_{i=1}^I x_i^2}. \quad (2.60)$$

- Dla  $\alpha \rightarrow \infty$  – średnią dysjunkcyjną:

$$\lim_{\alpha \rightarrow \infty} \left( \bigoplus_{i=1}^I (\alpha) x_i \right) = \max_{i=1} x_i. \quad (2.61)$$

System rozmyty typu MISO odwzorowuje rozmyte zbiory wejściowe  $\underline{A}$  w rozmyty zbiór wyjściowy  $B$ . Wykorzystując rozmyty, uogólniony schemat Modus Ponens, dla złożenia FATI otrzymujemy:

$$B'_{FATI} = \underline{A}' \circ R = \underline{A}' \circ \left[ \bigoplus_{i=1}^I R^{(i)} \right], \quad (2.62)$$

o funkcji przynależności:

$$\mu_{B'_{FATI}}(y) = \sup_{x \in \mathbb{X}} (\mu_{A'}(x) \star_{T_z} \mu_R(x, y)) = \sup_{x \in \mathbb{X}} (\mu_{A'}(x) \star_{T_z} \bigotimes_{i=1}^I \mu_{R^{(i)}}(x, y)), \quad (2.63)$$

oraz dla FITA:

$$B'_{FATI} = \bigoplus_{i=1}^I [A' \circ R^{(i)}], \quad (2.64)$$

o funkcji przynależności:

$$\mu_{B'_{FITA}}(y) = \bigoplus_{i=1}^I \sup_{x \in \mathbb{X}} (\mu_{A'}(x) \star_{T_z} \mu_{R^{(i)}}(x, y)). \quad (2.65)$$

W praktyce najczęściej stosowana jest metoda FITA, ze względu na prostszą implementację oraz bardziej intuicyjną formę obiektową. Implementacja operatora pozwalającego na złożenie reguł jest kłopotliwa. Zakłada się przy tym, że wyniki dla systemu FATI są zbliżone do tych, które są uzyskiwane metodą FITA. Wyniki uzyskiwane metodą FITA są bardziej precyzyjne niż metodą FATI [48]:

$$B'_{FATI} \subseteq B'_{FITA}, \quad (2.66)$$

czyli dla funkcji przynależności:

$$\mu_{B'_{FATI}} \leq \mu_{B'_{FITA}}. \quad (2.67)$$

Całości rozważań na temat podstaw matematycznych działania systemów rozmytych dopełniają operacje rozmywania (ang. *Fuzzyfication*) oraz wyostrażania (ang. *Defuzzyfication*).

Dane wejściowe dla systemu rozmytego najczęściej są w postaci numerycznej, a więc zachodzi problem odwzorowania wartości ostrej  $\underline{x}_0 \in \mathbb{X} \subset \mathcal{R}^N$  na wartość rozmytą określoną przez zbiór rozmyty  $A'$ . Częstym odwzorowaniem jest interpretacja wartości numerycznej jako rozmytego singletonu. Stosuje się również funkcje przynależności: trójkątne, gaussowskie, sigmoidalne, trapezoidalne i inne. W najczęściej wykorzystywanym przypadku, gdy wejściowy wektor  $\underline{x} \in \mathbb{X}$  jest wektorem singletonów, metody FATI i FITA prowadzą do takich samych wyników i są sobie równoważne [48].

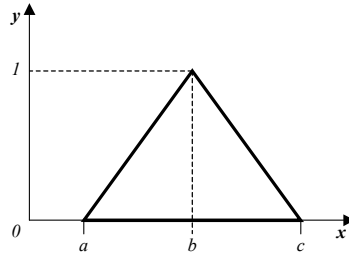
W pracy [49] przedstawiono dowód równoważności (przy określonych założeniach) pomiędzy systemami rozmytymi z koniunkcyjną i logiczną interpretacją reguł **if-then**. Szczegóły dotyczące budowy wybranych systemów rozmytych znajdują się w podrozdziale 2.3.

W niniejszej rozprawie stosowane są dwa typy funkcji przynależności przesłanek i konkluzji:

- Trójkątna:

$$\mu_{A_n^{(i)}}(x) = \begin{cases} 0 & \text{dla } (x \leq a_n^{(i)}) \vee (x \geq c_n^{(i)}) \\ \frac{x - a_n^{(i)}}{b_n^{(i)} - a_n^{(i)}} & \text{dla } x \in (a_n^{(i)}, b_n^{(i)}] \\ \frac{c_n^{(i)} - x}{c_n^{(i)} - b_n^{(i)}} & \text{dla } x \in (b_n^{(i)}, c_n^{(i)}) \end{cases}, \quad (2.68)$$

gdzie  $a_n^{(i)}$ ,  $b_n^{(i)}$ ,  $c_n^{(i)}$  opisują parametry trójkątnej funkcji przynależności:

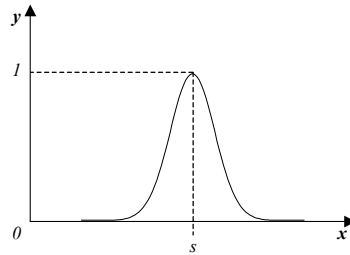


Rysunek 2.1: Trójkątna funkcja przynależności.

- Gaussa:

$$\mu_{A_n^{(i)}} = \exp\left(\frac{(x - s_n^{(i)})^2}{2(\sigma_n^{(i)})^2}\right), \quad (2.69)$$

gdzie  $s_n^{(i)}$ -centrum funkcji gaussowskiej,  $\sigma_n^{(i)}$ -dyspersja:



Rysunek 2.2: Gaussowska funkcja przynależności.

Wyjście systemu rozmytego stanowi zbiór rozmyty  $B'$ , uzyskany w wyniku zastosowania operatora agregacji. W praktyce najczęściej oczekuje się, że system „odpowie” dyskretną wartością numeryczną, reprezentującą rozmytą wartość uzyskaną w wyniku agregacji. Operator wyostrażania można zdefiniować następująco:

$$\{\mu_{B'}(y) \mid B' \in \mathcal{F}(\mathbb{Y}), y \in \mathbb{Y}\} \rightarrow \mathbb{Y}, \quad (2.70)$$

gdzie  $\mathcal{F}(\mathbb{Y})$  oznacza rodzinę zbiorów określoną na przestrzeni  $\mathbb{Y}$  zbiorów możliwych do uzyskania w wyniku operacji agregacji. W wyniku działania operatora wyostrzania otrzymujemy wyjściową wartość numeryczną  $y^*$ . Właściwe dobranie funkcji wyostrzającej bywa kłopotliwe, a sama operacja stanowi znaczne obciążenie obliczeniowe, uzależnione od przyjętego modelu systemu rozmytego i zastosowanego operatora. Przegląd najczęściej stosowanych operatorów znajduje się poniżej [48], [49]:

1. Metoda środka ciężkości (COG, ang. *Center of Gravity*):

$$y^* = \frac{\int_{\mathbb{Y}} y \cdot \mu_{B'}(y) dy}{\int_{\mathbb{Y}} \mu_{B'}(y) dy}. \quad (2.71)$$

2. Metoda zmodyfikowanego, indeksowanego środka ciężkości (MICOG, ang. *Modified Indexed Center of Gravity*), która pozwala na usunięcie z wyniku części nieinformatywnej (poziom jest ustalany za pomocą stałej  $\alpha$ ):

$$y^* = \frac{\int_{\mathbb{Y}_\alpha} y \cdot (\mu_{B'}(y) - \alpha) dy}{\int_{\mathbb{Y}_\alpha} (\mu_{B'}(y) - \alpha) dy}. \quad (2.72)$$

3. Metoda środka sum (COS, ang. *Center of Sum*), sumująca funkcje przynależności wyjściowych zbiorów rozmytych  $B^{(i)}$ :

$$y^* = \frac{\sum_{i=1}^I \int_{\mathbb{Y}} y \cdot \mu_{B^{(i)}}(y) dy}{\sum_{i=1}^I \int_{\mathbb{Y}} \mu_{B^{(i)}}(y) dy}. \quad (2.73)$$

4. Metoda maksimum (MD, ang. *Maximum Defuzzifier*), wybierający  $y^*$  jako punkt z przestrzeni  $\mathbb{Y}$  o największej wartości spośród wszystkich  $\mu_{B^{(i)}}(y)$ . Ponieważ może istnieć kilka punktów o tej samej (maksymalnej) wartości, stosowane są różne wersje tej metody, znane pod nazwami FOM – pierwsza maksymalna (ang. *First of Maximum*), LOM – ostatnia maksymalna (ang. *Last of Maximum*), lub MOM – środek maksimum (ang. *Middle of Maximum*) wyliczone jako średnia arytmetyczna  $y_{FOM}^*$  i  $y_{LOM}^*$ .

5. Metoda wysokości (HM, ang. *Height Method*) zwana również metodą uśrednionego centrum (CAD, ang. *Center Average Defuzzifier*):

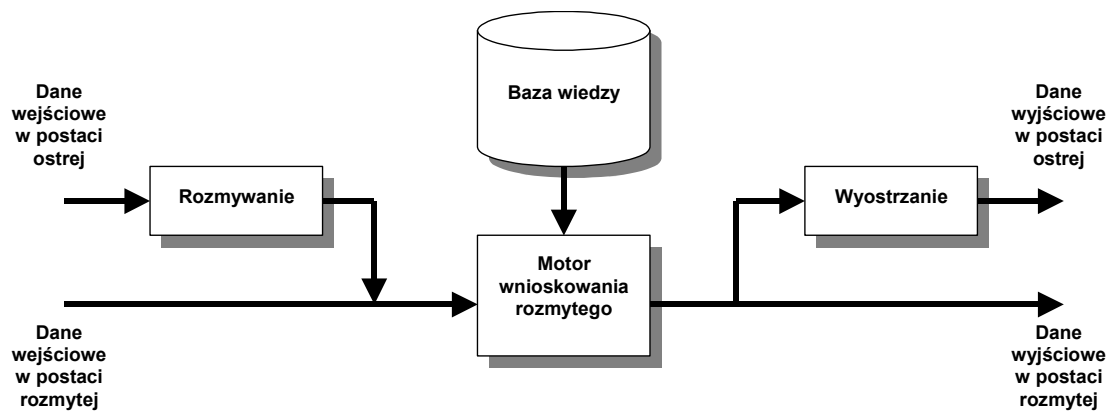
$$y^* = \frac{\sum_{i=1}^I \tau_i y^{(i)}}{\sum_{i=1}^I \tau_i}, \quad (2.74)$$



gdzie  $\tau_i$ -wysokość  $i$ -tego wyjściowego zbioru rozmytego. Szczegółowy opis operatorów można znaleźć w pracach [48] i [49].

## 2.3 Modele systemów rozmytych

Systemy rozmyte opierają swoje działanie na bazie wiedzy zapisanej w postaci rozmytych warunkowych reguł **if-then**. Samo wnioskowanie z wykorzystaniem bazy reguł opiera się na tzw. motorze wnioskowania rozmytego (ang. *Fuzzy Inference Engine*), wykorzystując fragmenty wiedzy zawarte w regułach do stworzenia odpowiedzi systemu na zadane warunki wejściowe. Motor pracuje z wykorzystaniem reguł wnioskowania rozmytego przedstawionych w podrozdziale 2.2.1. Prawie każdy system rozmyty składa się z podstawowych bloków funkcjonalnych, przedstawionych na poniższym rysunku:



Rysunek 2.3: Schemat blokowy systemu rozmytego.

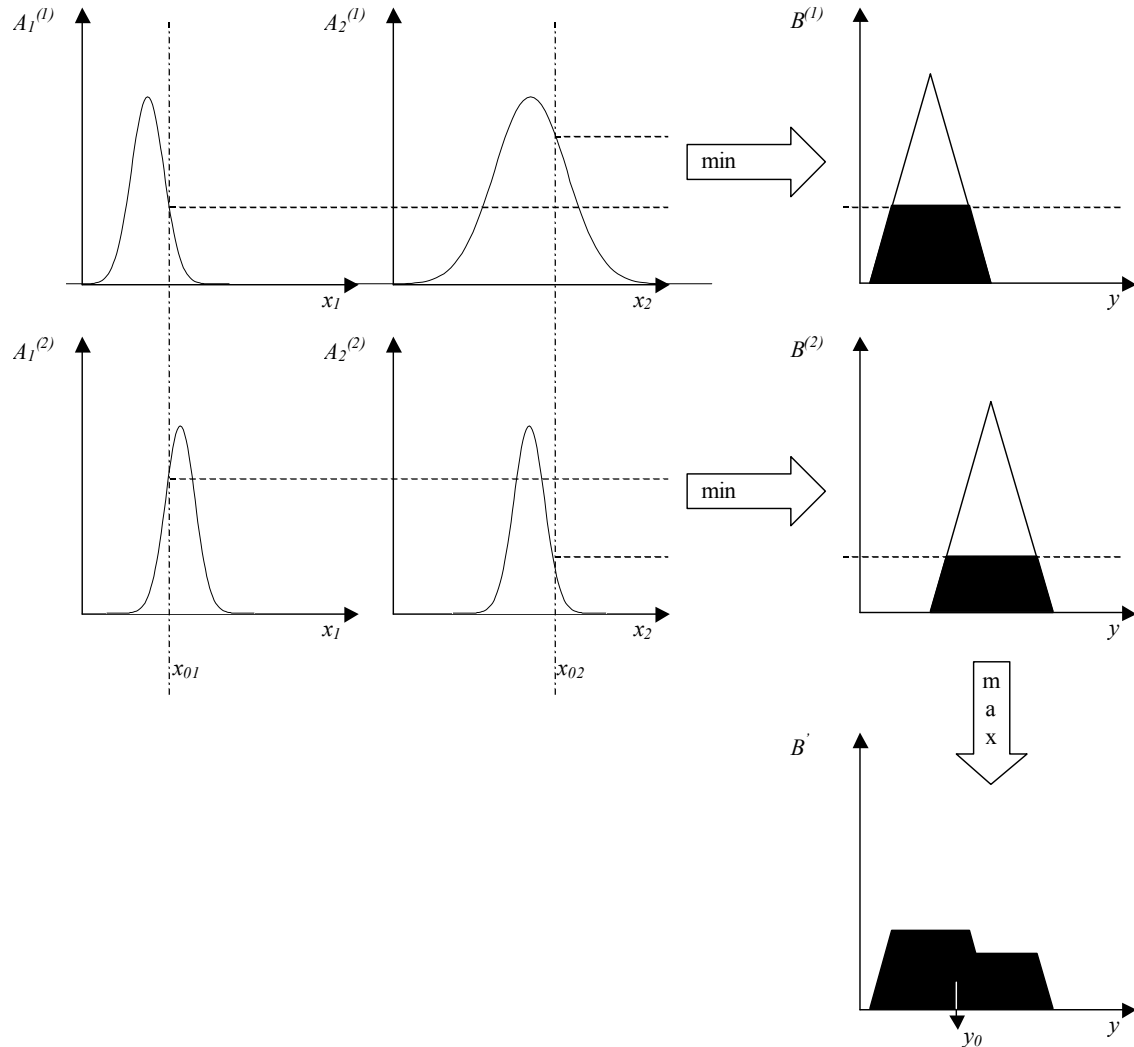
W zależności od zastosowanych danych wejściowych oraz oczekiwanego wyjścia, jak również zależnie od wybranego modelu, bloki funkcjonalne **Rozmywanie** oraz **Wyostrażanie** mogą nie być potrzebne [48].

Historycznie, pierwszym lingwistycznym modelem systemu rozmytego był system E. H. Mamdaniego i S. Assiliana [116]. W systemie tym przesłanki i konkluzje stanowiły zbiory rozmyte. Zastosowano koniunkcyjną metodę interpretacji reguł. Do modelowania łącznika „i” przesłanek zastosowano operator minimum ( $t$ -norma) oraz operator maksimum ( $s$ -norma) jako operator agregacji. Do wyostrażenia rozmytego wyniku zastosowano metodę COG. Wyjściowa, numeryczna odpowiedź systemu na zadane warunki wejściowe  $\underline{x}_0$ , traktowane jako rozmyte singletony, dana jest następującym wzorem:

$$\begin{aligned}
 y^* &= COG\left(\max_{i=1}^I\left(\min\left(\mu_{\underline{A}^{(i)}}(\underline{x}_0), \mu_{R^{(i)}}(\underline{x}, y)\right)\right)\right) = \\
 &= COG\left(\max_{i=1}^I\left(\min\left(\mu_{A_1^{(i)}}(x_{01}), \mu_{A_2^{(i)}}(x_{02}), \dots, \mu_{A_N^{(i)}}(x_{0N}), \mu_{B^{(i)}}(y)\right)\right)\right),
 \end{aligned}
 \tag{2.75}$$

gdzie  $\mu_A$  – funkcje przynależności.

Przykładowy system Mamdaniego-Assiliana, o dwóch wejściach i dwóch regułach, wykorzystujący gaussowskie funkcje przynależności w przesłankach oraz trójkątne funkcje przynależności w konkluzjach reguł przedstawiono na poniższym rysunku:



Rysunek 2.4: System Mamdaniego-Assiliana dla dwóch wejść i dwóch reguł warunkowych.

Zastosowana w systemie operacja wyostrażania COG jest kłopotliwa w implementacji i złożona obliczeniowo. Dlatego też często stosuje się uproszczenia, zmniejszające nakład obliczeń niezbędny na uzyskanie wyniku. Przykładowo, jako *t-normę* można zastosować iloczyn algebraiczny, a operację agregacji można sprowadzić do średniej arytmetycznej:

$$\mu_{B'}(y) = \frac{1}{I} \sum_{i=1}^I \mu_{\underline{A}^{(i)}}(x_0) \cdot \mu_{B^{(i)}}(y). \quad (2.76)$$

Korzystając z powyższego wzoru oraz metody COG, otrzymujemy wyjściową wartość systemu [43]:

$$y_0 = \frac{\sum_{i=1}^I \mu_{\underline{A}^{(i)}}(\underline{x}_0) \int_{\mathbb{Y}} y^{(i)} \cdot \mu_{B^{(i)}}(y) dy}{\sum_{j=1}^I \mu_{\underline{A}^{(j)}}(\underline{x}_0) \int_{\mathbb{Y}} \mu_{B^{(j)}}(y) dy}. \quad (2.77)$$

Wyliczając wcześniej i oznaczając pola powierzchni pod funkcją przynależności zbioru rozmytego  $i$ -tej konkluzji reguły jako  $\mathcal{P}(\mu_{B^{(i)}}(y))$  oraz przyjmując  $y^{(i)}$  jako środek ciężkości otrzymujemy [43]:

$$y_0 = \frac{\sum_{i=1}^I \mu_{\underline{A}^{(i)}}(\underline{x}_0) \cdot \mathcal{P}(\mu_{B^{(i)}}(y)) \cdot y^{(i)}}{\sum_{j=1}^I \mu_{\underline{A}^{(j)}}(\underline{x}_0) \cdot \mathcal{P}(\mu_{B^{(j)}}(y))}. \quad (2.78)$$

Do ustalenia wyjściowej wartości systemu wystarczy jednokrotne obliczenie pola powierzchni pod poszczególnymi funkcjami przynależności oraz podanie ich środków ciężkości. Zastosowanie operatora iloczynu algebraicznego w miejsce modelowanego spójnika „i” rozmytych reguł warunkowych **if-then**, przy założeniu koniunkcyjnej interpretacji tych reguł jest korzystne dla wydajności obliczeniowej systemu. W literaturze system taki nazywany jest systemem rozmytym *Larsena* [48].

Opisane powyżej kłopoty wydajnościowe doprowadziły do opracowania systemu *Takagi-Sugeno-Kanga* (TSK) [163], [169]. Baza wiedzy systemu jest różni się od bazy wiedzy systemu Mamdaniego zastosowaniem z konkluzji funkcji dających wartości ostre, zamiast zbiorów rozmytych:

$$\mathcal{R} = \{\mathcal{R}^{(i)}\}_{i=1}^I = \left\{ \text{jeżeli } \mathbf{i} \text{ } (X_{0n} \text{ jest } A_n^{(i)}) \text{ to } y^{(i)} = f^{(i)}(\underline{X}_0) \right\}_{i=1}^I, \quad (2.79)$$

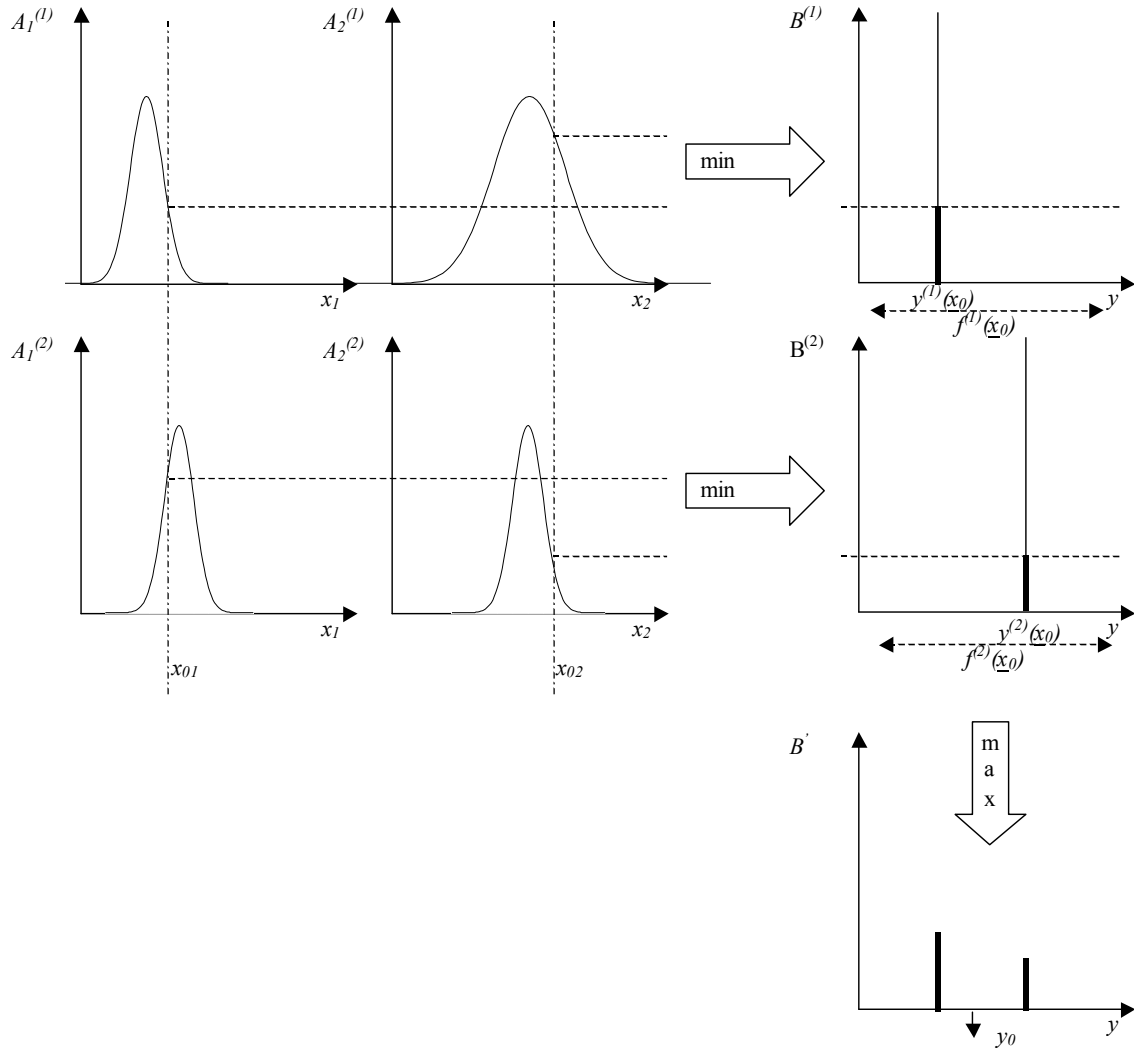
gdzie  $\underline{X}_0 = \underline{x}_0 = [x_{01}, \dots, x_{0N}]^T$  – wejściowy zbiór singletonów. Konkluzję  $y^{(i)}$  należy traktować jako singleton, o położeniu określonym przez funkcję  $f^{(i)}(\underline{X}_0)$ . Każdy singleton prezentuje wyjściową wartość numeryczną dla pojedynczej reguły rozmytej. Wartości te są następnie agregowane za pomocą operatora średniej arytmetycznej [163], [169]:

$$y_0 = \frac{\sum_{i=1}^I F^{(i)}(\underline{x}_0) \cdot y^{(i)}(\underline{x}_0)}{\sum_{j=1}^J F^{(j)}(\underline{x}_0)}, \quad (2.80)$$

gdzie  $F^{(i)}(\underline{x}_0)$  – stopień aktywacji  $i$ -tej reguły, który jest wyliczany z zastosowaniem  $t$ -normy, najczęściej operatora minimum lub iloczynu algebraicznego:

$$F^{(i)}(\underline{x}_0) = \mu_{A_1}^{(i)}(x_{01}) \star_T \dots \star_T \mu_{A_N}^{(i)}(x_{0N}). \quad (2.81)$$

Przykładowy system TSK o dwóch wejściach i dwóch regułach, wykorzystujący gaussowskie funkcje przynależności w przesłankach przedstawiono na poniższym rysunku:



Rysunek 2.5: System *Takagi-Sugeno-Kanga* (TSK) dla dwóch wejść i dwóch reguł warunkowych.

Funkcja  $f^{(i)}(\underline{x}_0)$  dla  $i=1,..,I$  najczęściej jest kombinacją liniową (wielomianem pierwszego stopnia) argumentów wejściowych:

$$y^{(i)}(\underline{x}_0) = p_0^{(i)} + p_1^{(i)} x_{01} + \dots + p_N^{(i)} x_{0N} = \underline{p}^{(i)T} \underline{x}_0', \quad (2.82)$$

gdzie  $\underline{p}^{(i)T}$  - wektor parametrów  $i$ -tej funkcji  $f_i(\underline{x}_0)$ ,  $\underline{x}_0'$  - rozszerzony wektor danych wejściowych:

$$\underline{x}_0' = \begin{bmatrix} 1 \\ \underline{x}_0 \end{bmatrix}. \quad (2.83)$$

W przypadku, gdy  $f_i(\underline{x}_0)$  jest kombinacją liniową danych wejściowych, system nazywany jest systemem TSK pierwszego rzędu. Ponieważ położenie wyjściowego singletonu jest uzależnione od wartości  $f_i(\underline{x}_0)$ , systemy TSK nazywane są również systemami z ruchomymi singletonami.

Niewątpliwą zaletą systemu TSK w porównaniu do systemów klasy Mamdaniego jest znaczne uproszczenie implementacji oraz większa efektywność obliczeniowa. Połączenie opisu lingwistycznego (w przesłankach) oraz analitycznego (funkcja położenia wyjściowego singletonu) może stanowić uproszczenie w przypadku projektowania systemów, co do których znana jest analitycznie część opisywanego zjawiska. Wadą systemów TSK jest duża liczba parametrów koniecznych do zdefiniowania dla każdej reguły oraz nieintuicyjne formowanie ostatecznego wyniku (w porównaniu do modelu Mamdaniego). Dodatkowo, ze względu na użycie singletonów w konkluzjach reguł, nie jest możliwe zastosowanie interpretacji logicznej, jak również wielu operatorów wyostrzania. Tej wady pozbawiony jest system zaprezentowany w podrozdziale 2.3.1.

### 2.3.1 System rozmyty z parametryzowanymi konkluzjami

W pracach [46], [48], [113] E. Czogała i J. Łęski przedstawili koncepcję systemu rozmytego z parametryzowanymi konkluzjami, stanowiącą połączenie systemów Mamdaniego i TSK. W systemach tego typu zbiory rozmyte konkluzji  $B^{(i)}(\theta)$  są określone za pomocą funkcji przynależności, tak jak w systemie Mamdaniego oraz dodatkowo za pomocą parametrów  $\theta$  funkcji określającej położenie środka ciężkości zbioru konkluzji, w zależności od argumentów wejściowych systemu. Przyjmując wejściowy zbiór lingwistyczny  $\underline{X}$ , jako zbiór singletonów  $\underline{x}_0$ , bazę wiedzy systemu z parametryzowanymi konkluzjami można przedstawić jako:

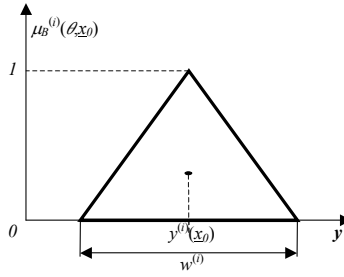
$$\mathcal{R} = \{R_{MISO}^{(i)}\}_{i=1}^I = \left\{ \text{jeżeli } \bigwedge_{n=1}^N (x_{0n} \text{ jest } A_n^{(i)}) \text{ to } \{Y \text{ jest } B^{(i)}(\theta, \underline{x}_0)\} \right\}_{i=1}^I. \quad (2.84)$$

Ostatecznie wyjściowy zbiór przynależności dany jest wzorem [48]:

$$\mu_{B'}(y, \underline{x}_0) = \bigoplus_{i=1}^I \mu_{B^{(i)}}(y, \underline{x}_0) = \bigoplus_{i=1}^I \Psi(\mu_{A^{(i)}}(\underline{x}_0), \mu_{B^{(i)}}(y)), \quad (2.85)$$

gdzie  $\Psi$  - *t-norma* dla systemu koniunkcyjnego lub implikacja dla systemu z logiczną interpretacją reguł. Z tego powodu, w zależności od zastosowanej metody wnioskowania, wynikowa funkcja przynależności  $\mu_{B'}(y, \underline{x}_0)$  może zawierać część nieinformatywną, dlatego do ustalenia ostrej, wyjściowej wartości systemu rozmytego zastosowano metodę MICO $G_\alpha$  [48].

Obecność zbiorów rozmytych reprezentujących wartości lingwistyczne w przesłankach i konkluzjach systemu pozwala na zastosowanie różnych interpretacji reguł warunkowych. W niniejszej rozprawie założono, że rozmyte zbiory w konkluzjach systemu mają formę trójkąta równoramiennego, a więc środek ciężkości leży na odcinku wyznaczającym symetrię trójkąta:

Rysunek 2.6: Funkcja przynależności konkluzji  $i$ -tej reguły.

Położenie konkluzji jest ustalane każdorazowo dla zestawu danych wejściowych i najczęściej jest opisane kombinacją liniową wartości wejściowych, analogicznie jak ma to miejsce w przypadku systemu TSK:

$$y^{(i)}(\underline{x}_0) = p_0^{(i)} + p_1^{(i)}x_{01} + \dots + p_N^{(i)}x_{0N} = \underline{p}^{(i)T} \underline{x}_0', \quad (2.86)$$

gdzie  $\underline{p}^{(i)T}$  - wektor parametrów konkluzji  $i$ -tej reguły warunkowej,  $\underline{x}_0'$  - rozszerzony wektor danych wejściowych:

$$\underline{x}_0' = \begin{bmatrix} 1 \\ \underline{x}_0 \end{bmatrix}. \quad (2.87)$$

Poziom aktywacji  $F^{(i)}(\underline{x}_0)$  oraz pole podstawy trójkąta w konkluzji reguły  $w^{(i)}$  określają pole powierzchni figury powstałej w wyniku operacji wnioskowania dla pojedynczej reguły rozmytej:

$$\mathcal{P}(\mu_{B^{(i)*}}(y, \underline{x}_0)) = g(F^{(i)}(\underline{x}_0), w^{(i)}), \quad (2.88)$$

gdzie  $\mu_{B^{(i)*}}^{(i)*}$  - funkcja przynależności zbioru rozmytego konkluzji  $i$ -tej reguły, po usunięciu części nieinformatywnej,  $g(F^{(i)}(\underline{x}_0), w^{(i)})$  - funkcja zależna od zastosowanej metody wnioskowania. Ostatecznie, wyjściową, ostrą wartość systemu można wyznaczyć następująco [48]:

$$y_0 = \frac{\sum_{i=1}^I g(F^{(i)}(\underline{x}_0), w^{(i)}) \cdot y^{(i)}(\underline{x}_0)}{\sum_{j=1}^J g(F^{(j)}(\underline{x}_0), w^{(j)})}. \quad (2.89)$$

W przypadku systemu koniunkcyjnego, wynik wnioskowania dla pojedynczej reguły jest wyliczany z użyciem operatora  $t$ -normy, najczęściej minimum (*Mamdaniego*) lub iloczynu algebraicznego (*Larsena*). Dla operatora minimum oraz trójkątnej funkcji przynależności w konkluzji, funkcja  $g(F^{(i)}(\underline{x}_0), w^{(i)})$  określa pole trapezu odciętego z trójkąta konkluzji na poziomie aktywacji reguły i przyjmuje postać [111]:

$$\begin{aligned}
g(F^{(i)}(\underline{x}_0), w^{(i)}) &= 2 \int_{\frac{y^{(i)}(\underline{x}_0) - w^{(i)}}{2}}^{\frac{y^{(i)}(\underline{x}_0)}{2}} \min(F^{(i)}(\underline{x}_0), \mu_{B^{(i)*}}(y, \underline{x}_0)) dy = \\
&= 2 \int_{\frac{y^{(i)}(\underline{x}_0) - w^{(i)}}{2}}^{\frac{\frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0) + y^{(i)}(\underline{x}_0) - \frac{w^{(i)}}{2}}{2}} \left( \frac{2}{w^{(i)}(\underline{x}_0)} y - \frac{2}{w^{(i)}(\underline{x}_0)} y^{(i)}(\underline{x}_0) + 1 \right) dy + \\
&\quad + 2 \int_{\frac{\frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0) + y^{(i)}(\underline{x}_0) - \frac{w^{(i)}}{2}}{2}}^{\frac{y^{(i)}(\underline{x}_0)}{2}} (F^{(i)}(\underline{x}_0)) dy = \\
&= \frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0) (2 - (F^{(i)}(\underline{x}_0))^2).
\end{aligned} \tag{2.90}$$

Wtedy wyjściowa wartość systemu  $y_0$  jest następująca:

$$y_0 = \frac{\sum_{i=1}^I \frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0) (2 - (F^{(i)}(\underline{x}_0))^2) \cdot y^{(i)}(\underline{x}_0)}{\sum_{j=1}^J \frac{w^{(j)}}{2} F^{(j)}(\underline{x}_0) (2 - (F^{(j)}(\underline{x}_0))^2)}. \tag{2.91}$$

Dla systemu koniunkcyjnego, przy zastosowaniu jako operatora  $t$ -normy iloczynu algebraicznego, otrzymujemy odpowiednio:

$$g(F^{(i)}(\underline{x}_0), w^{(i)}) = \frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0), \tag{2.92}$$

oraz wyjściową, ostrą wartość:

$$y_0 = \frac{\sum_{i=1}^I \frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0) \cdot y^{(i)}(\underline{x}_0)}{\sum_{j=1}^J \frac{w^{(j)}}{2} F^{(j)}(\underline{x}_0)} = \frac{\sum_{i=1}^I w^{(i)} \cdot F^{(i)}(\underline{x}_0) \cdot y^{(i)}(\underline{x}_0)}{\sum_{j=1}^J w^{(j)} \cdot F^{(j)}(\underline{x}_0)}. \tag{2.93}$$

W przypadku systemu z logiczną interpretacją reguł postać funkcji  $g$  jest uzależniona od zastosowanej implikacji. Dla implikacji spełniających własność silnej negacji (wg. Fodora, [58]), zachodzi [48]:

$$\alpha_i = 1 - F^{(i)}(\underline{x}_0). \tag{2.94}$$

Tak więc po usunięciu części nieinformatywnej, można wyliczyć pole powierzchni figury kreślonej w konkluzji każdej reguły, w wyniku zastosowania określonych implikacji, czyli można podać wzory opisujące funkcję  $g$  w zależności od zastosowanej implikacji rozmytej.

Poniżej zestawiono postać funkcji  $g$  dla najczęściej spotykanych implikacji rozmytych [43]:

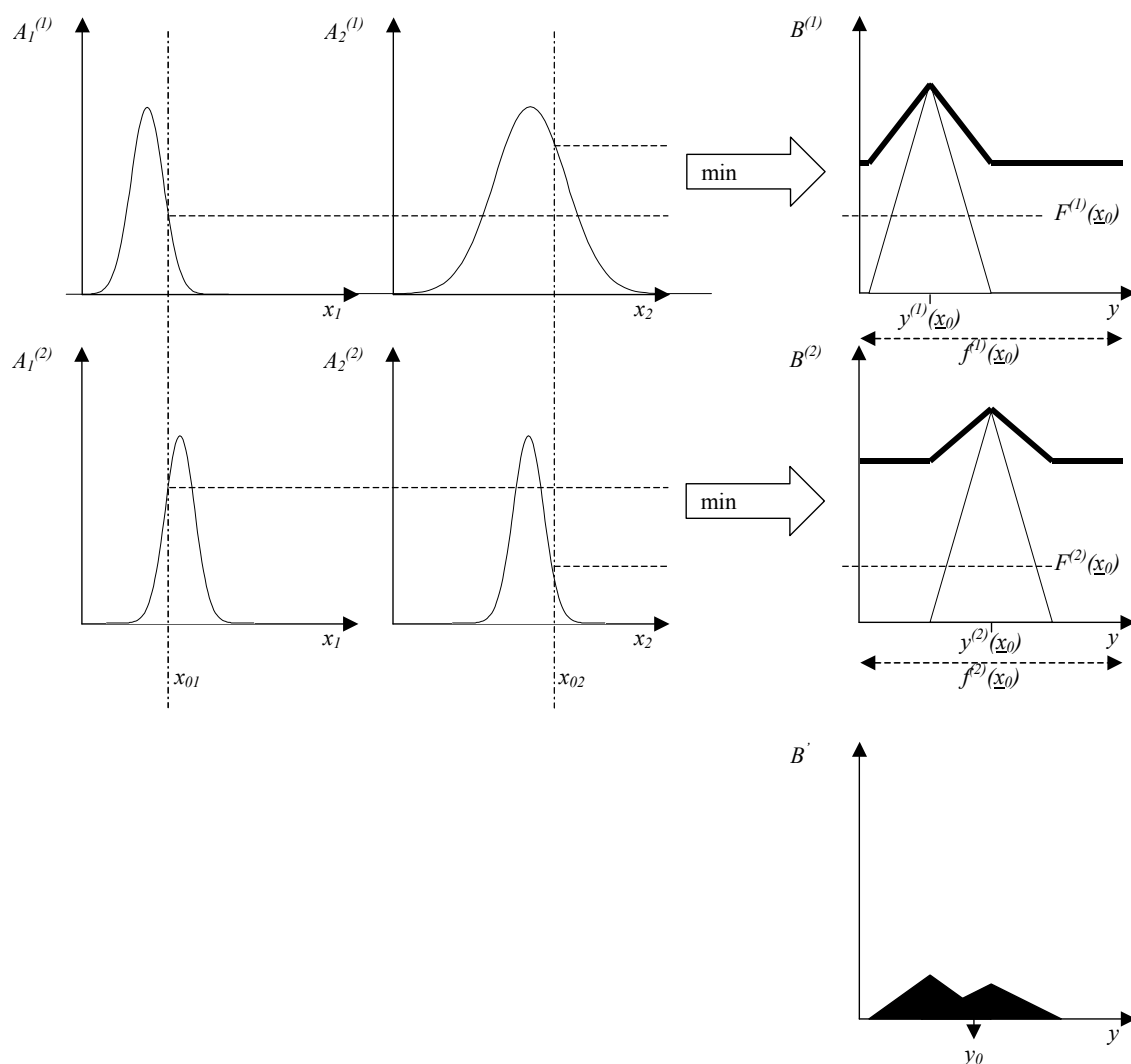
Tabela 2.2: Postać funkcji  $g$  dla wybranych implikacji rozmytych

Implikacja	$g(F^{(i)}(\underline{x}_0), w^{(i)})$	
Łukasiewicza	$w^{(i)} \left( F^{(i)}(\underline{x}_0) - \frac{1}{2} (F^{(i)}(\underline{x}_0))^2 \right)$	( 2.95 )
Fodora	$\begin{cases} w^{(i)} \left( \frac{1}{2} - F^{(i)}(\underline{x}_0) + (F^{(i)}(\underline{x}_0))^2 \right), & F^{(i)}(\underline{x}_0) \geq \frac{1}{2} \\ w^{(i)} \left( F^{(i)}(\underline{x}_0) - (F^{(i)}(\underline{x}_0))^2 \right), & F^{(i)}(\underline{x}_0) < \frac{1}{2} \end{cases}$	( 2.96 )
Reichenbacha	$\frac{w^{(i)}}{2} (F^{(i)}(\underline{x}_0))$	( 2.97 )
Kleenea-Dienes	$\frac{w^{(i)}}{2} (F^{(i)}(\underline{x}_0))^2$	( 2.98 )
Zadeha	$\begin{cases} w^{(i)} \left( F^{(i)}(\underline{x}_0) - \frac{1}{2} \right), & F^{(i)}(\underline{x}_0) \geq \frac{1}{2} \\ 0 & F^{(i)}(\underline{x}_0) < \frac{1}{2} \end{cases}$	( 2.99 )
Goguena	$w^{(i)} \left( 1 - \frac{1}{2} F^{(i)}(\underline{x}_0) \right)$	( 2.100 )
Gödel	$w^{(i)} \left( 1 - F^{(i)}(\underline{x}_0) + \frac{1}{2} (F^{(i)}(\underline{x}_0))^2 \right)$	( 2.101 )
Reschera	$w^{(i)} (1 - F^{(i)}(\underline{x}_0))$	( 2.102 )

Powyższe równania funkcji  $g$  wraz z wybranym operatorem  $\star_T$  (najczęściej minimum lub iloczyn algebraiczny), niezbędnym do wyliczenia poziomów aktywacji dla wielowejściowych reguł rozmytych oraz znajomość parametrów konkluzji pozwalają na stworzenie w pełni funkcjonującego systemu o parametrach jakościowych przewyższających systemy TSK i Mamdaniego.

Przykładowy system rozmyty z parametryzowanymi konkluzjami, o dwóch wejściach i dwóch regułach rozmytych przedstawiono na poniższym rysunku:





Rysunek 2.7: Przykładowy system rozmyty z parametryzowanymi konkluzjami o dwóch wejściach i dwóch regułach rozmytych. Zastosowano rozmytą implikację Reichenbacha.

Największym problemem stojącym przed projektantem systemu rozmytego z parametryzowanymi konkluzjami jest poprawne zdefiniowanie bazy wiedzy, w szczególności podanie parametrów opisujących poszczególne funkcje przynależności oraz znalezienie parametrów konkluzji dla każdej reguły rozmytej. Wektor opisujący parametry konkluzji dla każdej reguły systemu musi być znany na etapie wyliczania wartości wyjściowych. Rozwiązanie polegające na pobraniu tych danych od eksperta-człowieka jest wysoce kłopotliwe, ze względu na mało intuicyjną postać (najczęściej kombinacja liniowa parametrów wejściowych), dlatego jednym z możliwych sposobów jest automatyczne ustalenie parametrów systemu na podstawie danych uczących z zastosowaniem *metody najmniejszych kwadratów* lub też włączenie parametrów jako kolejnych wymiarów w przestrzeni poszukiwań, w trakcie automatycznego generowania reguł na podstawie danych numerycznych. W przypadku systemów ewolucyjno-

rozmytych, to ostatnie podejście jest rzadko stosowane, ze względu na dużą złożoność obliczeniową oraz kłopoty z oszacowaniem wartości parametrów konkluzji.

Zastosowanie metody najmniejszych kwadratów do wyznaczenia parametrów konkluzji reguł systemu rozmytego z parametryzowanymi konkluzjami może być dwojaki:

- *Uczenie globalne* – wykonywane jednocześnie dla wszystkich reguł tworzących bazę wiedzy systemu rozmytego.
- *Uczenie lokalne* – wykonywane dla pojedynczej reguły systemu rozmytego.

W zapisie wektorowym, wektor ostrych wartości wyjściowych systemu rozmytego z parametryzowanymi konkluzjami dla zbioru wejściowego o liczebności  $N_T$ , można wyznaczyć następująco [111]:

$$\underline{Y}_0 = \underline{D}(\underline{x}_0) \cdot \underline{P}, \quad (2.103)$$

gdzie  $\underline{x}_0$  – macierz wartości wejściowych dla systemu MISO,  $\underline{Y}_0$  – wektor wartości wyjściowych,  $\underline{P}$  – wektor parametrów systemu rozmytego,  $\underline{D}$  – macierz aktywacji. Organizację macierzy i wektorów oraz sposób obliczenia ich poszczególnych pozycji przedstawiono poniżej:

$$\underline{Y}_0 = [y_0(\underline{x}_0(1)), \dots, y_0(\underline{x}_0(N_T))]^T, \quad (2.104)$$

$$\underline{P} = [\underline{p}^{(1)T}, \dots, \underline{p}^{(I)T}]^T, \quad (2.105)$$

$$\underline{D} = \begin{bmatrix} \bar{g}^{(1)}(\underline{x}_0(1)) \cdot \underline{x}_0'(1)^T & \dots & \bar{g}^{(I)}(\underline{x}_0(1)) \cdot \underline{x}_0'(1)^T \\ \vdots & \ddots & \vdots \\ \bar{g}^{(1)}(\underline{x}_0(N_T)) \cdot \underline{x}_0'(N_T)^T & \dots & \bar{g}^{(I)}(\underline{x}_0(N_T)) \cdot \underline{x}_0'(N_T)^T \end{bmatrix}, \quad (2.106)$$

gdzie  $y_0(\underline{x}_0(i))$  – wartość wyjściowa systemu, wyliczona dla  $i$ -tego wektora danych wejściowych,  $\underline{p}^{(i)}$  – wektor parametrów dla  $i$ -tej reguły rozmytej systemu. Wartości funkcji  $\bar{g}^{(i)}$  stanowi znormalizowaną wartość wyjściową wyliczoną dla pojedynczej reguły, przed operacją agregacji i jest zależna od zastosowanej implikacji. Można ją wyliczyć w następujący sposób:

$$\bar{g}^{(i)}(x_0(n_T)) = \frac{g(F^{(i)}(x_0(n_T)), w^{(i)})}{\sum_{j=1}^J g(F^{(j)}(x_0(n_T)), w^{(j)})}, \quad (2.107)$$

gdzie  $n_T = 1 \dots N_T$ ,  $g$  – funkcja zależna od zastosowanej implikacji (Tabela 2.2),  $F^{(i)}$  – poziom aktywacji  $i$ -tej reguły,  $w^{(i)}$  – nośnik trójkątnej funkcji przynależności konkluzji  $i$ -tej reguły.

Średniokwadratowy wskaźnik jakości systemu rozmytego z parametryzowanymi konkluzjami można przedstawić w następującej formie:

$$J(\underline{P}) = (\underline{T} - \underline{Y}_0)^T (\underline{T} - \underline{Y}_0) = (\underline{T} - \underline{D} \cdot \underline{P})^T (\underline{T} - \underline{D} \cdot \underline{P}), \quad (2.108)$$

gdzie  $\underline{T}$  – wektor wartości wyjściowych ze zbioru uczącego:

$$\underline{T} = [t_0(1), \dots, t_0(N_T)]^T. \quad (2.109)$$

Minimalizując powyższe kryterium jakości możemy otrzymać optymalny zbiór parametrów konkluzji:

$$\min_P ((\underline{T} - \underline{D} \cdot \underline{P})^T (\underline{T} - \underline{D} \cdot \underline{P})) = J_{\min}(\underline{P}) \quad (2.110)$$

Różniczkując kryterium jakości  $J$  względem  $P$  otrzymujemy:

$$\frac{\partial J(\underline{P})}{\partial \underline{P}} = -2\underline{D}^T (\underline{T} - \underline{D}\underline{P}). \quad (2.111)$$

Optymalną wartość parametrów można obliczyć przyrównując powyższą pochodną do 0:

$$\frac{\partial J(\underline{P})}{\partial \underline{P}} = 0, \quad (2.112)$$

$$-2\underline{D}^T (\underline{T} - \underline{D}\underline{P}) = 0, \quad (2.113)$$

$$\underline{D}^T (\underline{T} - \underline{D}\underline{P}) = 0. \quad (2.114)$$

Przekształcając powyższe równanie otrzymujemy:

$$\underline{P} = (\underline{D}^T \underline{D})^{-1} \underline{D}^T \underline{T}. \quad (2.115)$$

Pozwala to wyliczyć wartości wskaźników dla przypadku uczenia globalnego. W przypadku uczenia lokalnego, minimalizowane kryterium dla  $i$ -tej reguły przyjmuje następującą postać:

$$J(p^{(i)}) = (\underline{T} - \underline{D}_1 \underline{p}^{(i)})^T \underline{G}^{(i)} (\underline{T} - \underline{D}_1 \underline{p}^{(i)}), \quad (2.116)$$

gdzie  $\underline{G}^{(i)}$  – macierz wag,  $\underline{D}_1$  – macierz zdefiniowana następująco:

$$\underline{D}_1 = \begin{bmatrix} x'_0(1)^T \\ \vdots \\ x'_0(N_T)^T \end{bmatrix}, \quad (2.117)$$

Diagonalna macierz  $\underline{G}^{(i)}$  określa wagę poszczególnych zestawów danych ze zbioru uczącego:

$$\underline{G}^{(i)} = \begin{bmatrix} g(F^{(i)}(x_0(1))) & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & g(F^{(i)}(x_0(N_T))) \end{bmatrix}. \quad (2.118)$$

Stosując *ważony estymator najmniejszych kwadratów* parametry konkluzji można wyliczyć (analogicznie, jak to ma miejsce dla metody uczenia globalnego) z poniższego równania [111]:

$$\underline{p}^{(i)} = (\underline{X}^T \underline{G}^{(i)} \underline{X})^{-1} \underline{X}^T \underline{G}^{(i)} \underline{T}. \quad (2.119)$$

Jak widać z ostatecznych równań określających  $\underline{P}$  oraz  $\underline{p}^{(i)}$ , wyliczenie wartości parametrów wymaga operacji odwracania macierzy, o wymiarach odpowiednio  $I \cdot (N+1) \times I \cdot (N+1)$  dla uczenia globalnego oraz  $(N+1) \times (N+1)$  dla uczenia lokalnego, co może wymagać sporego nakładu obliczeniowego w przypadku dużej liczby wejść systemu oraz sporej wielkości bazy wiedzy. Przeprowadzone badania własne wskazują porównywalność obu metod w zakresie jakości generowanych wyników jak i szybkości działania.

Istotnym elementem pojawiającym się na etapie wyznaczania parametrów konkluzji jest problem złego uwarunkowania macierzy podlegających odwracaniu, odpowiednio  $(\underline{D}^T \underline{D})$  dla uczenia globalnego i  $(\underline{X}^T \underline{G}^{(i)} \underline{X})$  dla uczenia lokalnego. Problem ten został rozwiązany poprzez zastosowanie korekty polegającej na dodaniu niewielkiej wartości ( $1 \cdot 10^{-7}$ ) do wszystkich elementów leżących na diagonalnej odwracanej macierzy.

W literaturze [43], [111] można spotkać rozwiązanie iteracyjne dla zagadnienia szacowania parametrów metodą uczenia globalnego jak i lokalnego, które pozwala na ominięcie operacji odwracania macierzy. Badania własne wykazują, że estymacja parametrów systemu rozmytego o kilkunastu wejściach, kilku regułach i zbiorze uczącym rzędu paruset rekordów, pozwalają na wykonanie obliczeń metodą standardową, na średniej klasy komputerze PC, w akceptowalnym czasie (implementacja w środowisku MATLAB). Szczegółowe informacje dotyczące czasu wykonania oraz parametry wykorzystywanego sprzętu opisano w rozdziale 6.

## 2.4 Podsumowanie

W rozdziale przedstawiono wybrane elementy teorii zbiorów rozmytych oraz modele systemów rozmytych, a w szczególności system rozmyty z parametryzowanymi konkluzjami, wykorzystywany na potrzeby dowodu prawdziwości postawionej w niniejszej rozprawie tezy. W celu uzyskania systemu rozmytego o dobrej jakości, rozumianej jako różnica pomiędzy wartościami wyjściowymi generowanymi przez system rozmyty, a wartościami wyjściowymi zawartymi w zbiorze referencyjnym (uczącym i testowym), niezbędne jest określenie parametrów funkcji przynależności (bazy danych) systemu rozmytego i liczby reguł (bazy reguł) oraz parametrów konkluzji. W niniejszej rozprawie do określenia funkcji przynależności przesłanek i konkluzji oraz bazy reguł wykorzystano metody ewolucyjne, natomiast parametry konkluzji systemu wyliczano z wykorzystaniem metody najmniejszych kwadratów.

W kolejnym rozdziale przedstawiono opis metod ewolucyjnych, wykorzystywanych na etapie generowania systemu rozmytego, natomiast wieloetapową metodę automatycznego wyodrębniania reguł **if-then** systemu rozmytego z parametryzowanymi konkluzjami szczegółowo opisano w rozdziale 5.

## 3

# Metody ewolucyjne

## 3.1 Techniki naśladowania ewolucji biologicznej

Metody ewolucyjne inspirowane są ewolucją biologiczną, zachodzącą wśród organizmów żywych. Pierwszą jednolitą koncepcję ewolucji w naturze przedstawił w XIX w. J. B. Lamarck, wg którego organizmy żywe doskonalały się i przekazują te cechy swojemu potomstwu, wpływając w ten sposób na lepsze przystosowanie się gatunku do otaczającego go środowiska. K. Darwin w 1859r. opublikował dzieło „*O powstaniu gatunków drogą doboru naturalnego, czyli o utrzymaniu się doskonalenia ras w walce o byt*”, gdzie opisano mechanizm doboru naturalnego oparty na czterech przesłankach: współzawodnictwie, zmienności, nadprodukcji i rozmnażaniu. Zgodnie z teorią Lamarka oraz A. Wallace’a (który niezależnie od Darwina również sformułował w 1858 r. teorię doboru naturalnego) cechy w ujęciu teorii Darwina były przekazywane z pokolenia na pokolenie w drodze rozmnażania (reprodukcji). Wg tej teorii, osobniki współzawodniczą między sobą o prawo do reprodukcji, wydając przy tym dużą ilość potomstwa (nadprodukcja) o różnych cechach (zmienność), które nie zawsze bierze udział w reprodukcji kolejnego pokolenia. Równolegle G. Mendel prowadził badania nad nośnikiem przekazującym cechy. Połączenie prac Darwina i Mendla położyło podstawy opisujące prawa rządzące dziedzicznym przekazywaniem cech oraz doskonaleniem gatunków. Wg bieżącej wiedzy [140], nośnikami informacji genetycznej są *chromosomy*, zawarte w jądrze komórkowym, składające się z *genów*. Gen [141] stanowi elementarną jednostkę informacji, kodującą pewną cechę. Zbiór cech osobniczych nazywany jest *fenotypem* [141], a ich kodowanie w chromosomie za pomocą genów *genotypem* [141]. Położenie genu w chromosomie zwane jest *locus* [111]. W procesie reprodukcji następuje częściowa wymiana materiału genetycznego, w procesie *krzyżowania* [142]. Poza krzyżowaniem, drugim czynnikiem powodującym zmiany w sekwencji genów są *mutacje* [143]. W naturze zachodzą one stosunkowo rzadko i w szczególności wiążą się z przyczynami zewnętrznymi, takimi jak np. promieniowanie lub chemiczne błędy w kopiowaniu genów. Mendel w swoich pracach wprowadził również pojęcie genu *dominującego* i *recesywnego* [144]. Gen dominujący maskuje cechę przekazywaną przez gen recesywny. To tłumaczy przyczynę, dla której potomstwo posiada czasem cechy (pomijając efekt powodowany przez mutację), które nie są uśrednieniem cech rodziców, bądź w ogóle wśród rodziców nie występują. Na potrzeby niniejszej rozprawy przyjęto, że cechy kodowane są jednostopniowo (nie ma genów recesywnych), natomiast możliwa jest różnica pomiędzy *genotypem* i *fenotypem*, tak jak to ma np. miejsce w przypadku dwójkowego kodowania cech lub gdy wektor parametrów numerycznych reprezentuje obiekt-regułę rozmytą lub system rozmyty.

Opisane powyżej skrótowo zmiany sugerują ich losowy charakter. Zaiste, natura w powyższych działaniach działa „na oślep” nie kierując się żadną heurystyką, istnieje natomiast czynnik, który powoduje, że osobniki lepiej przystosowane, posiadające bardziej pożądane cechy w zestawieniu z otoczeniem, mają większe szanse na przeżycie oraz wydanie potomstwa. Jest to wskazany przez Darwina *dobór naturalny*, który jednocześnie powoduje eliminację osobników gorzej przystosowanych. Miara określająca cechy osobnika jest *przystosowanie* (ang. *Fitness*). Oczywiście dobór naturalny preferuje osobniki o lepszym przystosowaniu.

Powyższa wiedza oraz obserwacje świata naturalnego doprowadziły do pojawienia się syntetycznych metod, w szczególności informatycznych, naśladujących procesy rekombinacji i doboru naturalnego w zamkniętych środowiskach. Skrótowe informacje dotyczące historii rozwoju metod ewolucyjnych opisano w podrozdziale 1.1. Można je również znaleźć w [55], [70] i [111].

Obliczenia ewolucyjne, jakkolwiek wszystkie bazujące na podobieństwie do ewolucji naturalnej, rozwijały się równolegle w wielu ośrodkach badawczych i zostały historycznie podzielone na kilka głównych nurtów:

1. Algorytmy genetyczne – cechą wyróżniającą są chromosomy o kodowaniu dwójkowym, choć w ostatnich latach znacznie spopularyzowano algorytmy genetyczne RCGA - kodowane zmiennoprzecinkowo (ang. *Real Coded Genetic Algorithms*). Stosowane są zarówno krzyżowanie jak i mutacja, a kolejne, nowe pokolenie zastępuje poprzednie. Za osobę, która zapoczątkowała ten nurt uznawany jest John Holland, choć algorytmy genetyczne zawdzięczają swoją niewątpliwą popularność pracom Davida Goldberga.
2. Programowanie genetyczne – gdzie z zastosowaniem algorytmów genetycznych poszukuje się rozwiązania problemu programistycznego przez przeszukiwanie przestrzeni możliwych rozwiązań. Reprezentacja jest zwykle dostosowana do rozwiązywanego problemu. Badania w tym nurcie zapoczątkował Lawrence Fogel.
3. Strategie ewolucyjne – cechą szczególną są geny kodowane zmiennoprzecinkowo. Historycznie pierwsze strategie składały się z populacji o jednym osobniku, z czasem pojawiły się strategie o liczniejszej populacji. Najczęściej wykorzystywanym operatorem była mutacja, obecnie wykorzystuje się również krzyżowanie i inne operatory. Kolejne, nowe pokolenie bazuje na osobnikach rodzicielskich i potomnych. Ten nurt zapoczątkowali Ingo Rechenberg i Hans-Paul Schwefel.

Wszystkie powyższe metody prowadzą do rozwiązania zagadnienia optymalizacji. Obliczenia ewolucyjne przeglądają przestrzeń dostępnych rozwiązań, wśród których znajduje się poszukiwane optimum, w sposób ukierunkowany przez odpowiednik doboru naturalnego. Rozpoczynając z początkowym (najczęściej wybranym losowo lub heurystycznie) pokoleniem startowym, ewoluują w kierunku coraz to lepszych rozwiązań. Odbyna się to w kolejnych pokoleniach za pomocą reprodukcji z wykorzystaniem operatorów genetycznych oraz selekcji osobników, proporcjonalnej do ich przystosowania (wybór osobników prezentujących lepsze rozwiązania problemu). Z matematycznego punktu widzenia, zagadnienie optymalizacji sprowadza się do poszukiwania wektora parametrów  $\underline{x} \in \mathcal{M} \subseteq \mathbb{R}^N$  takiego, aby wybrane kryterium

oceny  $f: \mathcal{M} \rightarrow \mathbb{R}$  było maksymalizowane (najczęstszy przypadek w obliczeniach ewolucyjnych), gdzie  $\mathcal{M}$  jest zbiorem dopuszczalnych rozwiązań w  $N$ -wymiarowym zagadnieniu optymalizacji. Oczywiście najczęściej kryterium optymalizacji będzie wielomodalne, tzn. będzie posiadać wiele maksimów lokalnych  $\underline{x}^{\uparrow L}$ :

$$\exists_{\varepsilon > 0} \forall_{\underline{x} \in \mathcal{M}} \rho(\underline{x}, \underline{x}^{\uparrow L}) < \varepsilon \Rightarrow f(\underline{x}) \leq f(\underline{x}^{\uparrow L}), \quad (3.1)$$

gdzie  $\rho(\bullet, \bullet)$  oznacza miarę pseudometryczną w przestrzeni rozwiązań. Jeśli  $\mathcal{M} \neq \mathbb{R}^n$ , rozwiązywane zagadnienie optymalizacji zawiera ograniczenia [5], [119]. Dzielą się one na nierównościowe  $g$  oraz równościowe  $h$  [111]:

$$\mathcal{M} = \{\underline{x} \in \mathbb{R}^N \mid \forall_j g_j(\underline{x}) \geq 0 \wedge h_j(\underline{x}) = 0\}. \quad (3.2)$$

Rozwiązanie problemu optymalizacji z ograniczeniami jest możliwe do zrealizowania za pomocą metod klasycznych, jednakże wymagana jest wtedy znajomość analitycznej postaci funkcji, która nie zawsze jest dana, może nie być różniczkowalna i dodatkowo zmienna w czasie. Wprowadzenie do klasycznych metod optymalizacji można znaleźć w pracy [122], pobeżną klasyfikację tych metod można również znaleźć w pracy [5].

Ponieważ metody ewolucyjne w dużej mierze opierają się na pseudolosowym modyfikowaniu kierunku poszukiwań, bardzo trudno jest określić dla nich warunki zbieżności oraz opisać w sposób usystematyzowany, językiem matematycznym, prawa rządzące zachowaniem się populacji. Projektanci systemów opierają się w dużej mierze na wynikach eksperymentalnych, które niezbicie dowodzą przydatności obliczeń ewolucyjnych w przedstawionych zastosowaniach. Istnieją również twierdzenia i hipotezy, sformułowane dla algorytmów genetycznych i poprzez analogię działania rozszerzone na strategie ewolucyjne, dowodzące zbieżności metod. Są to *twierdzenie o schematach* oraz *hipoteza cegiełek* znana również pod nazwą *hipotezy bloków budujących*, która jednak budzi sporo wątpliwości wśród badaczy [70]. Obie pozycje są opisane w podrozdziale 3.2.

## 3.2 Algorytmy genetyczne

W 1975 r. J. Holland [85] zaproponował *prosty algorytm genetyczny* (SGA, ang. *Simple Genetic Algorithm*), który przetwarza dwie populacje – populację bazową  $Q$  oraz populację potomną  $O$ . Poszukiwanie odbywa się w kierunku maksymalizacji funkcji przystosowania. Osobniki są kodowane za pomocą łańcuchów binarnych o ustalonej długości. Po zainicjalizowaniu zerowego pokolenia (w sposób losowy, systematyczny lub heurystyczny) następuje jego ocena. Następnie w kolejnych etapach następuje przejście głównej pętli algorytmu, aż do momentu zaistnienia warunków zatrzymania algorytmu. W każdym przejściu głównej pętli następuje kompletna wymiana osobników z populacji bazowej, odbywająca się w następujących krokach:

- Wybór osobników z  $Q^{(k)}$  do krzyżowania – zgodnie z regułami symulowanej selekcji naturalnej, wybór powinien preferować osobniki o lepszym dopasowaniu (większej

wartości funkcji przystosowania). Tradycyjnie stosuje się wybór z wykorzystaniem tzw. reprodukcji proporcjonalnej (ruletkowej) wprowadzonej przez J. Hollanda [85]. Szczegółowo problem ten został omówiony w podrozdziale 3.2.3. W ten sposób jest tworzona populacja tymczasowa  $\mathcal{J}^{(k)}$ .

- Operacje genetyczne na populacji tymczasowej  $\mathcal{J}^{(k)}$  prowadzą do uzyskania nowych osobników. W pierwszej kolejności wykonywana jest operacja krzyżowania. Dla każdej wylosowanej pary podejmowana jest decyzja (zgodnie z prawdopodobieństwem krzyżowania), czy osobniki rodzicielskie będą podlegać krzyżowaniu. Jeśli wynik jest pozytywny, nowe osobniki zastępują osobniki rodzicielskie, w przeciwnym wypadku osobniki rodzicielskie pozostają niezmienione. Kolejnym krokiem jest przeprowadzenie mutacji. Każdy osobnik podlega mutacji z określonym prawdopodobieństwem. Przegląd operatorów genetycznych znajduje się w podrozdziale 3.2.4.
- Ostatecznie powstałe po operacjach pokolenie  $\mathcal{O}^{(k)}$  staje się nowym pokoleniem bazowym  $\mathcal{Q}^{(k+1)}$  w kolejnym przebiegu pętli algorytmu.

Prosty algorytm genetyczny zaprezentowano poniżej:

```

inicjalizacja pokolenia zerowego  $\mathcal{Q}^{(0)}$ ;
ustawienie wskaźnika pokoleń  $k = 0$ ;
ocena pokolenia  $\mathcal{Q}^{(0)}$ ;
while (not warunek zatrzymania) do
     $\mathcal{J}^{(k)}$  = próbkowanie  $\mathcal{Q}^{(k)}$  celem wyboru rodziców do reprodukcji;
     $\mathcal{O}^{(k)}$  = krzyżowanie i mutacja  $\mathcal{J}^{(k)}$ ;
    ocena  $\mathcal{O}^{(k)}$ ;
     $\mathcal{Q}^{(k+1)}$  =  $\mathcal{O}^{(k)}$ ;
     $k = k + 1$ ;
end

```

### 3.2.1 Podstawy analityczne

W przypadku prostego algorytmu genetycznego mają zastosowanie niektóre rozważania analityczne, tłumaczące zasadę działania oraz dowodzące zbieżności. Należy jednak stwierdzić, że na dzień dzisiejszy nie istnieją metody dowodzące zbieżności w stricte deterministycznym sensie. Pomocne mogą być przedstawione poniżej: twierdzenie o schematach [85] i hipoteza bloków budujących.

*Twierdzenie o schematach* obowiązuje dla algorytmów genetycznych o dwójkowym kodowaniu chromosomów.

*Schematem*  $S$  nazywamy zbiór chromosomów, które na pewnych (ustalonych zgodnie ze wzorcem) pozycjach są zgodne z jego wzorcem podobieństwa  $P^S$  [85]:



$$P^S = \{p_{\mathcal{L}-1}^S \dots p_0^S \mid p_l^S \in \{0, 1, \#\}\} \quad (3.3)$$

gdzie #-symbol dowolny (0 lub 1),  $\mathcal{L}$ - długość ciągu kodującego,  $l=1, 2, \dots, \mathcal{L}$ . Schemat często jest utożsamiany ze swoim wzorcem prawdopodobieństwa.

Rząd schematu  $o(S)$  to liczba pozycji ustalonych (0 lub 1) we wzorcu prawdopodobieństwa  $P^S$  tego schematu [119]:

$$o(S) = \sum_{l=1}^{\mathcal{L}} (p_l^S \in \{0, 1\}). \quad (3.4)$$

Długość definiująca (rozpiętość) schematu  $\delta(S)$  to odległość pomiędzy dwiema skrajnymi pozycjami ustalonymi we wzorcu prawdopodobieństwa schematu [5]:

$$\delta(S) = \max_{p_l^S \in \{0, 1\}} (l) - \min_{p_l^S \in \{0, 1\}} (l) + 1, \quad (3.5)$$

gdzie  $l=1, 2, \dots, \mathcal{L}$ .

W procesie reprodukcji, osobniki rodzicielskie są wybierane proporcjonalnie do wartości funkcji przystosowania. Zakładając, że w  $k$ -tym pokoleniu znajduje się  $\xi$  reprezentantów schematu  $S$ :  $\xi(S, k)$ , możemy określić oczekiwaną liczbę osobników pasujących do tego samego schematu w następnym pokoleniu  $\xi(S, k+1)$ .

Średnie przystosowanie schematu  $S$  w  $k$ -tym pokoleniu  $f(S, k)$  można wyliczyć następująco [111]:

$$f(S, k) = \frac{\sum_{i=1}^{\xi} f_i(k)}{\xi}, \quad (3.6)$$

gdzie  $f_i(k)$ -przystosowanie  $i$ -tego elementu pasującego do schematu  $S$ .

Średnie przystosowanie  $k$ -tego pokolenia można wyliczyć następująco [70]:

$$\overline{f(k)} = \frac{\sum_{i=1}^M f(k)}{M}, \quad (3.7)$$

gdzie  $M$  – liczebność pokolenia.

Prawdopodobieństwo reprodukcji schematu jest proporcjonalne do jego przystosowania i wynosi [119]:

$$p_r(S) = \frac{f(S, k)}{\sum_{i=1}^M f(k)}, \quad (3.8)$$

gdzie  $f_T(k)$  – sumaryczne przystosowanie wszystkich osobników pokolenia  $k$ .

Na tej podstawie możemy wyliczyć oczekiwaną liczbę ciągów pasujących do schematu  $S$  w kolejnym kroku algorytmu [111]:

$$E(\xi(S, k+1)) = \xi(s, k) \cdot M \cdot p_r(S) = \xi(s, k) \cdot \frac{f(S, k)}{f(k)}, \quad (3.9)$$

przy czym powyższy wzór nie uwzględnia zmian związanych z operatorami genetycznymi, tj. krzyżowaniem oraz mutacją. W celu uwzględnienia działania powodowanego przez operatory genetyczne, niezbędna jest określenie, jak zmieni się liczba ciągów pasujących do schematu w wyniku zastosowania tych operacji. W przypadku operacji krzyżowania jednopunktowego, punkt cięcia jest losowany równomiernie spośród  $(\mathcal{L}-1)$  możliwych pozycji, zatem prawdopodobieństwo przetrwania schematu  $S$  wynosi [119]:

$$p_{rCS}(S) \geq 1 - p_{rC} \cdot \frac{\delta(S)}{\mathcal{L}-1}, \quad (3.10)$$

gdzie  $p_{rC}$  - prawdopodobieństwo krzyżowania. Nierówność wynika z faktu, iż w wyniku rozcięcia ciągu pomiędzy pozycjami ustalonymi może powstać osobnik pasujący do schematu.

Prawdopodobieństwo przetrwania schematu przy wykonywaniu operacji mutacji można oszacować następująco [111]:

$$p_{rMS}(S) = (1 - p_{rM})^{o(S)}, \quad (3.11)$$

przy czym najczęściej  $P_{rM} \ll 1$ , zatem powyższe równanie można przybliżyć zależnością liniową [111]:

$$p_{rMS}(S) \approx 1 - o(S) \cdot p_{rM}, \quad (3.12)$$

Ostatecznie oszacowanie wzrostu schematu można przedstawić następująco [111]:

$$E(\xi(S, k+1)) \geq \xi(s, k) \cdot \frac{f(S, k)}{f(k)} \left( 1 - p_{rC} \frac{\delta(S)}{\mathcal{L}-1} - o(S) \cdot p_{rM} \right). \quad (3.13)$$

Podsumowując twierdzenie o schematach, można wywnioskować, że *krótkie, niskiego rzędu i oceniane powyżej średniej schematy uzyskują wykładniczo rosnącą liczbę łańcuchów w kolejnych pokoleniach* [119].

*Hipoteza o blokach budujących* twierdzi, iż *algorytm genetyczny poszukuje działania zbliżonego do optymalnego przez zestawianie krótkich, niskiego rzędu schematów o dużej wydajności działania, zwanych blokami budującymi*. Rozważania na temat poprawności powyższej hipotezy można znaleźć w pracy A. Bethke [13], gdzie dzięki zastosowaniu funkcji Walsh'a i transformowaniu schematów pokazano możliwości określenia zbieżności do wyniku optymalnego lub suboptymalnego, w zależności od zastosowanej funkcji przystosowania. Hipoteza ta sugeruje, że sposób kodowania, zastosowany w algorytmach genetycznych jest istotny dla jego poprawnego działania (zbieżności).

Algorytm w każdym pokoleniu przetwarza co najmniej  $2^{\mathcal{L}}$  schematów i nie więcej niż  $2^M$  schematów. J. Holland w pracy [85] wykazał, że efektywnie jest przetwarzanych przynajmniej

$M^3$  schematów, natomiast A. Bertoni i M. Dorigo w [12] wykazali, że jest to prawdą wyłącznie przy pewnych ograniczeniach związanych z wielkością populacji i nie jest prawdziwe w ogólnym przypadku. Opisywana właściwość algorytmów genetycznych jest nazywana *ukrytą równoległością*.

Pewne szczególne przypadki algorytmów genetycznych doczekały się głębszej analizy. Z. Michalewicz i A. Szalas [165] badali algorytmy genetyczne z wykorzystaniem twierdzenia Banacha o punkcie stałym [179]. Więcej na ten temat można również znaleźć w [119]. D. Goldberg i P. Segrest [72] przedstawili analizę zbieżności algorytmów genetycznych z zastosowaniem skończonych łańcuchów Markowa [180]. G. Rudolph w pracy [153] dowiódł, że algorytmy genetyczne w klasycznej formie nigdy nie zbiegają się do optimum globalnego, natomiast zbieżne są ich modyfikacje zachowujące z pokolenia na pokolenie najlepsze rozwiązanie (*model elitarny*). Eiben i wsp. [57] proponowali analizę z wykorzystaniem ujednolicenia algorytmów genetycznych i metody *symulowanego wyżarzania* [43], [100] podając warunki, dla których proces ewolucyjny znajduje optimum z prawdopodobieństwem 1.

Pomimo braku deterministycznych dowodów ogólnej poprawności działania algorytmów genetycznych, ich zastosowanie zarówno w sferze nauki jak i w przemyśle dowodzi poprawności przyjętych założeń i tworzy z nich uniwersalne narzędzie do rozwiązywania problemów, których rozwiązanie metodami klasycznej optymalizacji nastęcza trudności.

### 3.2.2 Kodowanie osobników

Kodowanie osobników jest zależne od rodzaju rozwiązywanego zadania. Klasyczny algorytm genetyczny wykorzystuje kodowanie binarne. Przyjęty sposób kodowania ma kluczowe znaczenie dla sposobu przeszukiwania przestrzeni rozwiązań przez algorytm genetyczny. Zagadnienie kodowania sprowadza się do odnalezienia poprawnych funkcji przekształcających *przestrzeń genotypu w przestrzeń fenotypu* i odwrotnie [111]:

$$\begin{aligned}\tilde{x} &= \mathcal{K}(x) \\ x &= \mathcal{D}(\tilde{x}),\end{aligned}\tag{3.14}$$

gdzie  $\tilde{x}$  - punkt w przestrzeni genotypu,  $x$  - punkt w przestrzeni fenotypu, funkcje kodującą i dekodującą stanowią odwzorowania [111]:

$$\begin{aligned}\mathcal{K} : \mathcal{M} &\rightarrow \{0,1\}^{\mathcal{L}} \\ \mathcal{D} : \{0,1\}^{\mathcal{L}} &\rightarrow \mathcal{M}\end{aligned}\tag{3.15}$$

Funkcje kodująca i dekodująca powinny spełniać następujące postulaty [111]:

- Każde rozwiązanie zadania może być przedstawione jako genotyp, ponieważ w przeciwnym wypadku, część rozwiązań z założenia byłaby nieosiągalna:

$$\forall_{x \in \mathcal{M}} \exists_{\tilde{x} \in \tilde{\mathcal{M}}} \mathcal{D}(\tilde{x}) = x.\tag{3.16}$$

- Kodowanie nie powinno wprowadzać dodatkowej wielomodalności:

$$\| \underline{x}_1 - \underline{x}_2 \|_f \leq \| \underline{x}_1 - \underline{x}_3 \|_f \Rightarrow \| \tilde{x}_1 - \tilde{x}_2 \|_g \leq \| \tilde{x}_1 - \tilde{x}_3 \|_g . \quad (3.17)$$

Często ciąg binarny genotypu reprezentuje wektor wartości po stronie fenotypu, dlatego też jest on podzielony na  $Q$  podciągów o długości  $n_i$ , zatem:

$$\mathcal{L} = \sum_{q=1}^Q n_i . \quad (3.18)$$

Najczęściej przyjmuje się stałe  $n_i$  m.in. ze względu na prostszą implementację struktur przechowujących, a przede wszystkim ze względu na prostotę działania operatorów genetycznych. W przypadku optymalizacji numerycznej, gdzie wartości w przestrzeni fenotypu  $x_i \in \mathcal{R}^Q$ , długość podciągu binarnego  $n_i$  jest najmniejszą liczbą całkowitą, spełniającą nierówność [111]:

$$10^{m_i} (x_i^{\rightarrow} - x_i^{\leftarrow}) \leq 2^{n_i} - 1 , \quad (3.19)$$

gdzie  $[x_i^{\leftarrow}, x_i^{\rightarrow}]$  – przedział zmienności  $x_i$ ,  $m_i$  – precyzja rozwiązania (liczba cyfr znaczących).

Zatem funkcja dekodująca przyjmuje postać [111]:

$$x_i = \mathcal{D}(\tilde{x}_i) = v_i + \frac{\text{dec}(\tilde{x}_i) \cdot (v_i - u_i)}{2^{n_i} - 1} , \quad (3.20)$$

gdzie  $\text{dec}(\bullet)$  jest operatorem zamiany wartości binarnej na dziesiętną wg naturalnego kodu dwójkowego [111]:

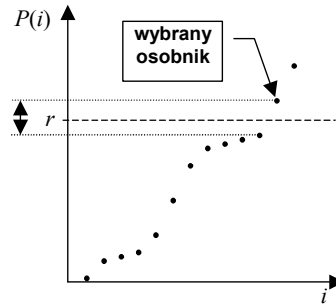
$$\text{dec}(\tilde{x}_i) = \sum_{j=1}^{n_i} \tilde{x}_{ij} 2^{j-1} , \quad (3.21)$$

gdzie  $\tilde{x}_{ij} \in \{0,1\}$  - wartość  $j$ -tego bitu w  $i$ -tym podciągu genotypu. Taka reprezentacja jest naturalna dla konwersji i przechowywania wartości numerycznych w postaci binarnej, jednakże nie spełnia ona własności monotoniczności, wprowadzając dodatkową wielomodalność do rozwiązywanego problemu. Jednym z rozwiązań jest zastosowanie kodu *Gray'a* [139]. D. Goldberg [71] eksperymentował również z *niechlujnym kodowaniem*. Innym rozwiązaniem jest reprezentacja zmiennoprzecinkowa (RCGA, ang. *Real Coded Genetic Algorithms*), w którym przestrzeń genotypu najczęściej jest tożsama z przestrzenią fenotypu. Takie kodowanie jest również stosowane w *strategiach ewolucyjnych*, opisanych w podrozdziale 3.3. Stosowanie kodowania zmiennoprzecinkowego wymusiło opracowanie odmiennych operatorów od tych, które były stosowane dla kodowania dwójkowego, przykładowo krzyżowania uśredniającego czy mutacji nierównomiernej. Przykłady takich operatorów opisano w podrozdziale 3.3.4.

### 3.2.3 Reprodukacja

Sposób przeprowadzenia reprodukcji ma kluczowe znaczenie dla poprawnego i skutecznego działania algorytmu genetycznego. W prostym algorytmie genetycznym wykorzystuje się klasyczną metodę ruletkową. Szczegółowy opis metody wraz z przykładami można znaleźć m.in. w [154]. Implementację mechanizmu najprościej jest przeprowadzić wyliczając

dystrybuantę z funkcji przystosowania dla pokolenia  $Q^{(k)}$  a następnie losując liczbę  $r$  z wykorzystaniem generatora liczb losowych o rozkładzie równomiernym [192]. Wybierany jest osobnik reprezentujący zakres wartości dystrybuanty, do którego przynależy wylosowana wartość  $r$ :



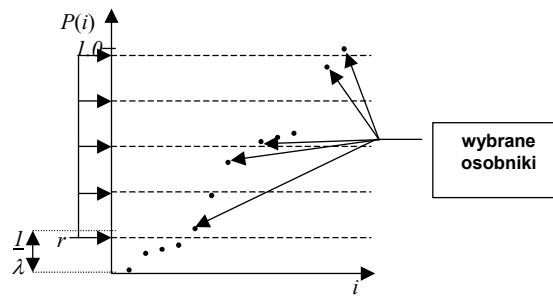
Rysunek 3.1: Metoda proporcjonalna (przykład).

W metodzie proporcjonalnej prawdopodobieństwo wylosowania osobnika jest wprost proporcjonalne do jego wartości funkcji przystosowania. Cechą szczególną tego modelu jest zależność od stosowanej funkcji przynależności – jeżeli w funkcji przynależności można wydzielić stałą wartość dodawaną do wartości zmiennej:

$$f' = f + c, \quad (3.22)$$

gdzie  $c > 0$ , wtedy proporcje liczby potomków będą odmienne niż w przypadku, gdy  $c = 0$ . Jest to narzędzie pozwalające na sterowanie naciskiem selektywnym, przy czym w przypadku, gdy funkcja przynależności nie jest dobrze udokumentowana, może przesądzić o zbieżności do optimum lokalnego [5] poprzez zmniejszenie różnorodności populacji.

Innym modelem selekcji jest selekcja ze *stochastycznym próbkowaniem*, gdzie wyboru dokonuje się poprzez równomierne próbkowanie (ze stałym krokiem) na odcinku wartości proporcjonalnych do znormalizowanych wartości funkcji przystosowania osobników w pokoleniu  $Q^{(k)}$ . Punkt  $r$  pierwszego próbkowania jest ustalany na podstawie realizacji zmiennej losowej o rozkładzie równomiernym z zakresu  $\left[0, \frac{1}{\lambda}\right)$ , gdzie  $\lambda$  – liczba osobników wybieranych do populacji tymczasowej  $\mathcal{J}^{(k)}$ , natomiast kolejne punkty  $(\lambda - 1)$  znajdują się w równych odstępach  $\left(\frac{1}{\lambda}\right)$  od pierwszego, losowanego punktu:

Rysunek 3.2: Metoda ze stochastycznym próbkowaniem (przykład,  $\lambda=5$ ).

Kolejną metodą wyboru osobników do reprodukcji jest *metoda turniejowa*, w której w pierwszej kolejności losowane jest  $q_t$  osobników ( $q_t \in [2, \mu]$ , losowanie o rozkładzie równomiernym), a następnie szereguje się je wg określonej rangi (nie zawsze są to wartości funkcji przystosowania) i wybiera się najlepszego z nich, wstawiając go do pokolenia tymczasowego  $\mathcal{J}^{(k)}$ . Proces ten trwa aż do momentu zapełnienia populacji rodziców  $\mathcal{J}^{(k)}$ . Znałe są odmiany bez zwracania (raz wybrany osobnik nie podlega ponownie selekcji) oraz ze zwracaniem (częściej stosowane w praktyce, ze względu na łatwość implementacji).

Następną odmianą reprodukcji jest *metoda rangowa (rankingowa)*. W metodzie tej prawdopodobieństwo reprodukcji jest podane w formie rangi – wartości określającej „jakość” osobnika. Wybór osobników następuje proporcjonalnie do nadanych im rang.

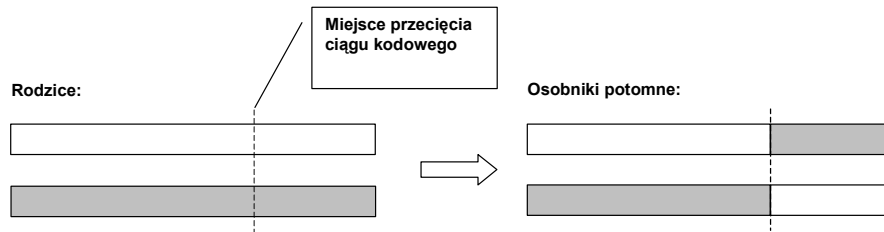
Odmianą reprodukcji rankingowej jest *metoda progowa* (stosowana głównie dla populacji o dużej liczebności), w której osobniki szereguje się wg wartości funkcji przystosowania, wybierając do pokolenia tymczasowego  $\mathcal{J}^{(k)}$  najlepsze (o największej wartości funkcji przystosowania) na liście.

W literaturze opublikowano wiele wariantów powyższych algorytmów selekcji, jak również inne, rzadziej spotykane. De Jong badał *model wartości oczekiwanej* [53]. Brindle w [21] badała różnice między sześcioma metodami selekcji. Dokładniejsze informacje można znaleźć m.in. w [5], [85], [70], [119].

### 3.2.4 Operatory genetyczne

Analogicznie jak to ma miejsce w naturze, operatorem pierwszoplanowym w algorytmach genetycznych jest operator krzyżowania. Mutacja stanowi operator drugoplanowy.

Krzyżowanie polega na wymianie części chromosomu pomiędzy dwoma osobnikami. Holland [85] zaproponował krzyżowanie jednopunktowe:

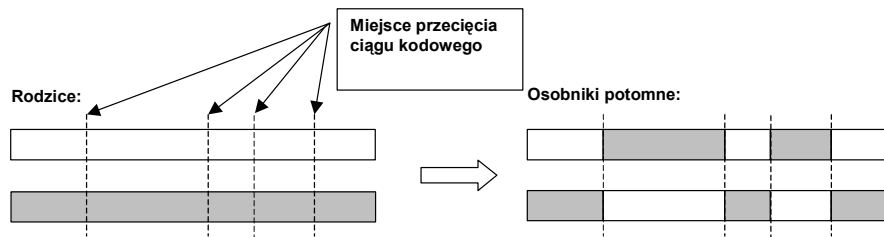


Rysunek 3.3: Ilustracja graficzna krzyżowania jednopunktowego.

W operacji biorą udział dwa osobniki rodzicielskie, a w jej wyniku powstają dwa osobniki potomne, zastępujące w pokoleniu tymczasowym swoich rodziców. Formalnie operację tę można zapisać następująco:

$$\begin{aligned} \underline{x}_\alpha &= [b_{\alpha,1}, \dots, b_{\alpha,\mathcal{L}}]^T & \Rightarrow & \quad \underline{x}_1 = [b_{\alpha,1}, \dots, b_{\alpha,\zeta}, b_{\beta,\zeta+1}, \dots, b_{\beta,\mathcal{L}}]^T \\ \underline{x}_\beta &= [b_{\beta,1}, \dots, b_{\beta,\mathcal{L}}]^T & & \quad \underline{x}_2 = [b_{\beta,1}, \dots, b_{\beta,\zeta}, b_{\alpha,\zeta+1}, \dots, b_{\alpha,\mathcal{L}}]^T, \end{aligned} \quad (3.23)$$

gdzie  $\underline{x}_\alpha, \underline{x}_\beta$  - chromosomy osobników rodzicielskich,  $\underline{x}_1, \underline{x}_2$  - chromosomy osobników potomnych,  $b_{v,i} \in \{0,1\}$  - pozycja ciągu kodowego,  $\zeta$  - punkt rozcięcia. Odmianą krzyżowania jednopunktowego jest krzyżowanie wielopunktowe, gdzie rozcięcie i wymiana materiału genetycznego następuje w wielu miejscach:



Rysunek 3.4: Ilustracja graficzna krzyżowania wielopunktowego (przykład dla czteropunktowego).

Istnieją dwie wersje losowania punktów rozcięcia – ze zwracaniem i bez zwracania.

$$\begin{aligned} \underline{x}_\alpha &= [b_{\alpha,1}, \dots, b_{\alpha,\mathcal{L}}]^T & \Rightarrow & \quad \underline{x}_1 = [b_{\alpha,1}, \dots, b_{\alpha,\zeta_1}, b_{\beta,\zeta_1+1}, \dots, b_{n_1,\zeta_C}, b_{n_2,\zeta_C+1}, \dots, b_{n_2,\mathcal{L}}]^T \\ \underline{x}_\beta &= [b_{\beta,1}, \dots, b_{\beta,\mathcal{L}}]^T & & \quad \underline{x}_2 = [b_{\beta,1}, \dots, b_{\beta,\zeta_1}, b_{\alpha,\zeta_1+1}, \dots, b_{n_2,\zeta_C}, b_{n_1,\zeta_C+1}, \dots, b_{n_1,\mathcal{L}}]^T, \end{aligned} \quad (3.24)$$

gdzie  $\zeta_c, C$  – liczba punktów rozcięcia,  $c=1,2,\dots,C$  – punkt rozcięcia,  $n_1=\alpha$  i  $n_2=\beta$  dla  $C$  nieparzystego oraz  $n_1=\beta$  i  $n_2=\alpha$  dla  $C$  parzystego.

Kolejną odmianą operatora jest *krzyżowanie równomierne*, w którym potencjalnej wymianie podlegają wszystkie bity ciągu kodowego osobników rodzicielskich, w zależności od wylosowanej wartości  $r$ , o rozkładzie równomiernym [192] na przedziale  $[0,1)$ . Kolejne bity ciągu kodowego są tworzone następująco:

$$\begin{aligned} x_{1l} &= \begin{cases} x_l^{(\alpha)}, & r_l < p_k \\ x_l^{(\beta)}, & r_l \geq p_k \end{cases} \\ x_{2l} &= \begin{cases} x_l^{(\beta)}, & x_{1l} = x_l^{(\alpha)} \\ x_l^{(\alpha)}, & x_{1l} = x_l^{(\beta)} \end{cases} \end{aligned} \quad (3.25)$$

gdzie  $p_k$  – parametr krzyżowania.

Przedstawione pokrótce operatory krzyżowania są podstawą do opracowywania ich bardziej skomplikowanych wersji. Szczegółowe opisy innych operatorów można znaleźć w literaturze, m.in. w [70].

Operator mutacji stanowi dopełnienie operatorów stosowanych w klasycznym algorytmie genetycznym i zazwyczaj jest stosowany po operatorze krzyżowania. Operatory te zmieniają (najczęściej nieznacznie) ciąg kodowy osobnika. Prawdopodobieństwo mutacji jest znacznie mniejsze od prawdopodobieństwa krzyżowania (zwykle  $P_{rM} \in [0.001, 0.1]$ ) lub  $P_{rM} = \frac{1}{L}$ . To ostatnie podejście powoduje zmniejszanie prawdopodobieństwa mutacji wraz ze wzrostem długości ciągu kodującego. W praktyce mutację przeprowadza się losując dla każdego bitu zmienną losową  $r_l$ , o rozkładzie równomiernym na przedziale  $[0,1)$  i jeżeli wartość  $r_l < P_{rM}$ , wykonywana jest zamiana bitu:

$$x_l = \begin{cases} (x_l \oplus 1), & r_l < P_{rM} \\ x_l, & r_l \geq P_{rM} \end{cases} \quad (3.26)$$

gdzie  $\oplus$  - operacja binarna XOR.

Odpowiednie dobranie wartości  $P_{rM}$  ułatwia wybrnięcie z obszaru przyciągania minimum lokalnego ale zbyt duża wartość może powodować błędzenie algorytmu po przestrzeni wyników. Dlatego też wykorzystuje się modyfikację prawdopodobieństwa mutacji, zmniejszając jego wartość wraz ze wzrostem numeru pokolenia. Takie podejście powoduje, że w początkowym etapie algorytm genetyczny przeszukuje całą przestrzeń rozwiązań (zwiększenie *eksploracji* kosztem *eksploatacji*), a dla końcowych numerów pokoleń ułatwia dotarcie do optimum globalnego.

Do rzadziej stosowanych operatorów należy operator inwersji, zamieniający kolejność bitów w określonym podciągu ciągu kodowego osobnika oraz operatory krzyżowania, w których biorą udział więcej niż 2 osobniki rodzicielskie [5].

Zastosowanie mutacji jest niezbędne do umożliwienia przeszukiwania całej przestrzeni rozwiązań. W wyniku działania klasycznych operatorów krzyżowania (nawet wielopunktowych) często nie można osiągnąć wszystkich punktów w przestrzeni poszukiwań (jest to uzależnione od rozkładu osobników pokolenia startowego). Wszystkie osobniki potomne leżą w hiperkostce rozpiętej pomiędzy osobnikami macierzystymi. Dopiero zastosowanie operatora mutacji pozwala na „wysunięcie” poszukiwań poza tę hiperkostkę. Przykład obrazujący taką sytuację można znaleźć m.in. w [119]. Dlatego też dobrze dobrane operatory genetyczne (albo ich łączny zestaw) spełniają postulat *spójności przestrzeni poszukiwań*, który zapewnia, że algorytm



genetyczny może osiągnąć dowolny (dozwolony) punkt przestrzeni poszukiwań startując z dowolnego, innego punktu tej przestrzeni. Niespełnienie tego postulatu powoduje, iż algorytm, który rozpocznie swoje działanie na pokoleniu początkowym o wyjątkowo niekorzystnych cechach, z założenia jest skazany na niepowodzenie. Drugim istotnym postulatem jest postulat o *braku obciążenia operatorów*. Ze względu na charakter operatorów i przy określonych typach kodowania, pewne kierunki poszukiwania są preferowane bardziej niż inne. Operator nie jest *obciążony* jeśli przy braku nacisku selektywnego generuje wszystkie genotypy z jednakowym prawdopodobieństwem.

### 3.2.5 Warunki zatrzymania

Najczęstszym warunkiem zatrzymania algorytmu genetycznego jak i strategii ewolucyjnej jest osiągnięcie określonej liczby iteracji. Ze względu na charakter mechanizmów ewolucyjnych, określenie liczby iteracji jest uzależnione od wielkości przestrzeni poszukiwań, liczebności pokoleń, zastosowanych metod selekcji i operatorów genetycznych, a najczęściej ograniczeniami czasowymi. Przyjmuje się przy tym, że wynik osiągnięty po wskazanej liczbie iteracji jest wystarczająco bliski optimum. J. Arabas w pracy [5] wskazuje różne kryteria zatrzymania algorytmu:

- Kryterium *minimalnej poprawy* – polegające na monitorowaniu zmian i zatrzymaniu algorytmu w momencie, w którym wzrost funkcji przystosowania osobników (najlepszego, średniej) staje się niezadowalająco niski (z reguły w kilku kolejnych pokoleniach),
- Kryterium *zadowalającego poziomu funkcji przystosowania* – polegające na monitorowaniu zmian i zatrzymaniu algorytmu w momencie, w którym funkcja przystosowania osiąga założony wcześniej poziom – podejście to najczęściej wiąże się z koniecznością znajomości szacunkowych wartości funkcji przystosowania uznawanych za dobre,
- Kryterium *maksymalnego kosztu* – polegające na zakończeniu obliczeń w momencie, w którym zostaje osiągnięty założony koszt obliczeń lub też gdy koszt obliczeń jest nieproporcjonalnie wysoki w stosunku do osiąganego poprawy funkcji przynależności,
- Kryterium *zaniku różnorodności* – polegające na zatrzymaniu algorytmu w momencie, w którym różnorodność genetyczna pokolenia bazowego spada poniżej pewnego minimum,
- Kryterium *zaniku samoczynnie adaptowanego zasięgu operatora mutacji* – stosowane w przypadku użycia mutacji z adaptacją zasięgu – polegające na zatrzymaniu algorytmu, gdy zaniknie samoczynna adaptacja zasięgu dla mutacji.

## 3.3 Strategie ewolucyjne

*Strategie ewolucyjne* rozwijały się równolegle z algorytmami genetycznymi. Początkowo największą różnicą w stosunku do algorytmów genetycznych było wykorzystanie kodowania

zmiennoprzecinkowego oraz przykładanie dużego znaczenia do operatora mutacji – operator krzyżowania został wprowadzony w późniejszym czasie.

### 3.3.1 Modele strategii ewolucyjnych

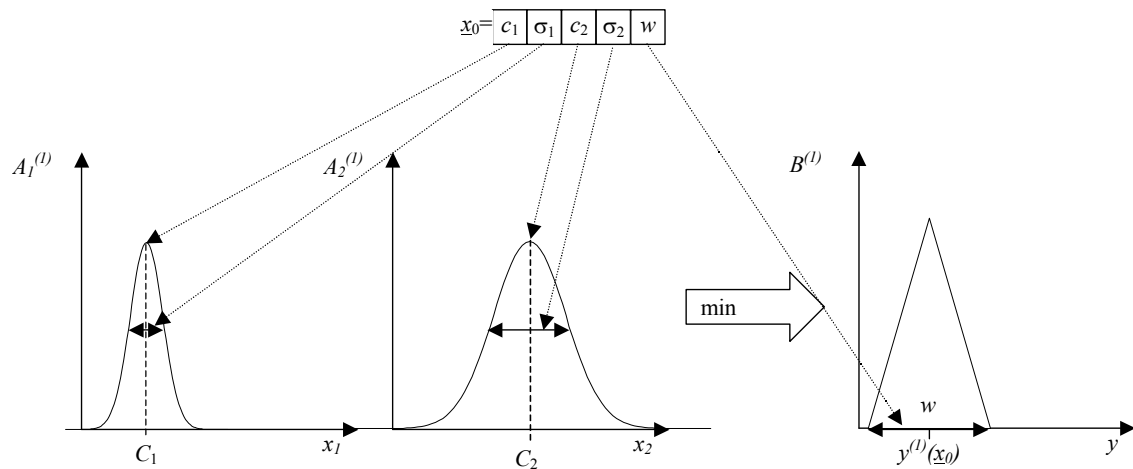
Historycznie pierwszym modelem strategii ewolucyjnych, wprowadzonym przez I. Rechenberga i H. P. Schwefela, był model składający się z jednego osobnika, a jedynym wykorzystywanym operatorem była mutacja. Nowy osobnik, który w wyniku mutacji osiągnął poprawę i spełnił wszystkie ograniczenia zastępował starego. Utrzymywanie jednoosobnikowej populacji jest intuicyjnie nieefektywne, dlatego też wraz z rozwojem pojawiły się strategie ewolucyjne wieloelementowe. Wzrost liczebności populacji umożliwił wprowadzenie operatora krzyżowania. Najbardziej znane strategie ewolucyjne to [111]:

- strategia (1+1) – opisywana powyżej, może być uznana za szczególny przypadek symulowanego wyżarzania, cechując się przy tym niewielką odpornością w zastosowaniach do funkcji wielomodalnych,
- strategia ( $\mu+1$ ) – strategia, w której pokolenie bazowe składa się z  $\mu$  osobników, natomiast w wyniku działania operatorów genetycznych tworzony jest jeden nowy osobnik, który następnie jest włączany do populacji bazowej w kolejnym pokoleniu, o ile spowoduje to polepszenie średniego dostosowania – najczęściej stosuje się wymianę za osobnika najgorszego,
- strategia ( $\mu, \lambda$ ) – strategia, w której pokolenie bazowe składa się z  $\mu$  osobników, a w wyniku działania operatorów genetycznych otrzymuje się pokolenie tymczasowe składające się z  $\lambda$  osobników, z których następnie wybiera się  $\mu$  najlepszych, stanowiących kolejne pokolenie,
- strategia ( $\mu+\lambda$ ) – strategia analogiczna do strategii ( $\mu, \lambda$ ), przy czym nowe pokolenie powstaje poprzez wybór osobników najlepszych, spośród połączonych populacji bazowej ( $\mu$  - osobników) oraz potomnej ( $\lambda$  - lambda osobników),
- strategii ewolucyjne z określonym maksymalnym czasem życia osobników – najczęściej stosuje się modyfikacje ( $\mu, \lambda$ ), ( $\mu+\lambda$ ) np. ( $\mu, \mathcal{K}, \lambda$ ) – gdzie  $\mathcal{K}$  - parametr określający maksymalny czas życia osobnika. W strategiach tych nawet najlepszy osobnik nie może trwać dłużej niż  $\mathcal{K}$  pokoleń.

Powyższe przykłady nie wyczerpują wszystkich przypadków i stanowią jedynie przegląd najczęściej stosowanych strategii. Szczegółowe informacje można znaleźć w [5], [111], [119].

### 3.3.2 Kodowanie osobników

Ze względu na formę zmiennoprzecinkową genotypu, w strategiach ewolucyjnych często zachodzi tożsamość genotypu z reprezentowanym przez niego fenotypem. Kodowanie jest uzależnione od charakteru rozwiązywanego zadania. Często jest to wektor danych numerycznych, opisujący cechy osobnika, np. parametry funkcji przynależności oraz zakodowane typy funkcji przynależności reguły rozmytej:



Rysunek 3.5: Kodowanie reguły o dwóch wejściach, dla systemu z parametryzowaną konkluzją.

Często osobnik ma zakodowane dodatkowe informacje wykorzystywane przez operatory genetyczne, np. przez operator mutacji – jest to niezbędne do tzw. samoadaptacji operatora.

### 3.3.3 Podstawy analityczne

Poza intuicyjnymi wnioskami, które można przenieść na pole strategii ewolucyjnych z dziedziny algorytmów genetycznych, opisanych w podrozdziale 3.2.1 główną rolę spełnia *twierdzenie o zbieżności*, które dowodzi zbieżności strategii ewolucyjnej typu (1+1) w deterministycznym sensie z prawdopodobieństwem 1, przy odpowiednio dużej liczbie iteracji (pokoleń).

### 3.3.4 Operatory ewolucyjne

Głównym i początkowo jedynym operatorem stosowanym w strategiach ewolucyjnych była mutacja. Ze względu na zmiennoprzecinkową reprezentację, niezbędne było opracowanie operatorów odmiennych niż opisane w podrozdziale 3.2.4. Analogicznie jak ma to miejsce w operatorach genetycznych 3.2.4, istnieją postulaty, które powinny być spełniane przez operatory zastosowane w strategiach ewolucyjnych, a w szczególności dotyczące operatora mutacji. Są to postulat o *spójności przestrzeni poszukiwań* i postulat o *braku obciążeń operatorów*. Więcej na ten temat można znaleźć w [5]. Operator mutacji w przypadku kodowania zmiennoprzecinkowego wymaga określenia zakresu zmian, któremu powinien podlegać. Zakres ten może być dodatkowo parametrem zależnym od numeru pokolenia i sytuacji w populacji (operator z samoadaptacją). W związku z tym, reprezentacja genotypu została rozszerzona o dodatkowe parametry – wyznaczniki sposobu działania dla operatora mutacji.

W algorytmie ewolucyjnym I. Rechenberga i H. P. Schwefela osobnik składał się z dwóch zmiennopozycyjnych wartości:

$$\tilde{\underline{x}} = [\underline{x}^T, \underline{\sigma}^T]^T, \quad (3.27)$$

gdzie  $\underline{x}=[x_1, \dots, x_L]^T$  – wektor wartości reprezentujących rozwiązanie z przestrzeni poszukiwań,  $\underline{\sigma}=[\sigma_1, \dots, \sigma_L]^T$  – wektor odchyłeń standardowych, najczęściej wielowymiarowa zmienna losowa o rozkładzie normalnym (gaussowskim) [181], zerowym wektorze wartości średnich i diagonalnej macierzy kowariancji. W wyniku mutacji wartość  $\tilde{\underline{x}}$  zmienia się zgodnie z poniższym równaniem:

$$\tilde{\underline{x}}_i : \begin{cases} \sigma'_i = \sigma_i \cdot e^{(\tau_1 r + \tau_2 r'_i)} \\ x'_i = x_i + \sigma'_i r'_i \end{cases}, \quad (3.28)$$

gdzie  $\sigma'$  – krok mutacji,  $r, r_i, r'_i$  – realizacje zmiennej losowej o rozkładzie normalnym [181],  $\tau_1, \tau_2$  – parametry, zależne od optymalizowanego kryterium. Często przyjmowane wartości  $\tau_1$  i  $\tau_2$  to [111]:

$$\tau_1 \propto \frac{1}{\sqrt{2 \cdot l}}, \quad (3.29)$$

$$\tau_2 \propto \frac{1}{\sqrt{2 \sqrt{l}}}. \quad (3.30)$$

Rozkład normalny został wybrany ze względu na analogię do świata rzeczywistego, gdzie mniejsze zmiany wynikające z mutacji są obserwowane częściej niż zmiany duże. Często też stosowany jest rozkład Cauchy'ego [128] zamiast rozkładu normalnego, ponieważ dla wartości argumentów oddalonych od środka krzywych dystrybuanta Cauchy'ego przyjmuje większe wartości niż dystrybuanta rozkładu normalnego [5].

Innym rozwiązaniem jest zastosowanie funkcji, która wraz ze względny wzrostem numeru pokolenia zmniejsza perturbacje wprowadzane przez operator mutacji – w początkowym okresie działania strategii duże zmiany mają niebagatelne znaczenie dla eksploracji przestrzeni poszukiwań, natomiast w końcowej fazie cyklu małe zmiany pomagają w eksploatacji obszaru optimum.

Dla operatora mutacji z samoadaptacją I. Rechenberg zaproponował *regułę 1/5 sukcesów*, po raz pierwszy zastosowaną dla strategii ewolucyjnej (1+1). W myśl tej reguły, co ustaloną liczbę  $k$  pokoleń dokonywana jest korekta kroku mutacji  $\sigma$  zgodnie z poniższymi zasadami:

$$\sigma' \leftarrow \begin{cases} n_d \cdot \sigma, & \mathcal{V}(k) < \frac{1}{5} \\ \sigma & \mathcal{V}(k) = \frac{1}{5} \\ n_i \cdot \sigma & \mathcal{V}(k) > \frac{1}{5} \end{cases}, \quad (3.31)$$

gdzie  $V(k)$  – współczynnik sukcesów w  $k$  poprzednich pokoleniach (sukces oznacza, że w wyniku mutacji powstał osobnik lepszy od średniego przystosowania),  $n_d, n_i$  – parametry określające szybkość (odpowiednio zmniejszania i zwiększania) kroku mutacji. H. P. Schwefel w [160] stosuje następujące wartości  $n_d = 0.82$  i  $n_i = 1/0.82$ .

Często stosowanym operatorem jest *mutacja nierównomierna* [119], gdzie po wybraniu elementu  $x_l$  ciągu kodowego do zmutowania, jego nowa wartość jest obliczana w następujący sposób:

$$x'_l \leftarrow \begin{cases} x_l + \Delta(k, \max(x_l) - x_l) & \text{gdy } r = 1 \\ x_l - \Delta(k, x_l - \min(x_l)) & \text{gdy } r = -1 \end{cases} \quad (3.32)$$

gdzie operatory  $\max(\bullet)$  i  $\min(\bullet)$  określają odpowiednio górny i dolny przedział zmienności argumentu,  $r$  jest realizacją zmiennej losowej o rozkładzie jednorodnym, dwupunktowym  $\{-1, 1\}$ . Funkcja  $\Delta(k, y)$  jest zdefiniowana następująco:

$$\Delta(l, y) = y \cdot \left( 1 - r \left( 1 - \frac{k}{K} \right)^b \right), \quad (3.33)$$

gdzie  $r$  – realizacja zmiennej losowej o rozkładzie równomiernym na przedziale  $[0, 1]$ ,  $b$  – parametr mutacji,  $K$  – maksymalna liczba pokoleń,  $k$  – numer bieżącego pokolenia,  $k=1, 2, \dots, K$ .

Operator krzyżowania (rekombinacji) w postaci wymiany fragmentów chromosomu pomiędzy osobnikami macierzystymi, znanej z algorytmów genetycznych, jest stosowany w strategiach ewolucyjnych niezwykle rzadko, ze względu na to, że wymiana fragmentów chromosomu nie w każdym zadaniu ma interpretację logiczną. Korzystanie z takiej postaci operatora częstokroć pociąga za sobą konieczność sprawdzania, czy nowo powstały osobnik należy do hiperprzestrzeni dopuszczalnych rozwiązań. W związku z tym, dla strategii ewolucyjnych oraz algorytmów genetycznych, w których stosuje się kodowanie zmiennoprzecinkowe, powstały dedykowane operatory *krzyżowania arytmetycznego* [120], [121]. Najczęściej stosowane jest tzw. *krzyżowanie uśredniające*, gdzie z dwóch osobników rodzicielskich  $\underline{x}^\alpha$  i  $\underline{x}^\beta$  powstaje jeden osobnik potomny. Wartości poszczególnych genów chromosomu osobnika potomnego są wyliczane następująco [111]:

$$x'_l = r_l \cdot x_{\alpha,l} + (1 - r_l) \cdot x_{\beta,l}, \quad (3.34)$$

gdzie  $\underline{x}^\alpha = [x_{\alpha,1}, \dots, x_{\alpha,L}]$ ,  $\underline{x}^\beta = [x_{\beta,1}, \dots, x_{\beta,L}]$ ,  $r_l$  – realizacja zmiennej losowej o rozkładzie równomiernym na przedziale  $[-d, 1+d]$ ,  $d \geq 0$ . Dla  $d=0$  powstały osobnik znajduje się w hiperprostokącie wyznaczonym przez osobniki rodzicielskie. Gdy  $d>0$ , osobnik potomny może leżeć poza tą przestrzenią. Jeśli przestrzeń poszukiwań jest wypukła i  $d=0$ , osobnik powstały w wyniku działania operatora krzyżowania jest osobnikiem dopuszczalnym. Analogicznym operatorem jest operator *rekombinacji liniowej*, który korzysta z jednokrotnego losowania zmiennej o przedstawionych powyżej założeniach. Wartości poszczególnych genów chromosomu osobnika potomnego są wyliczane następująco [111]:

$$x'_l = r \cdot x_{\alpha,l} + (1-r) \cdot x_{\beta,l} \quad (3.35)$$

W wyniku działania rekombinacji liniowej, powstają osobniki potomne, które leżą na prostej łączącej osobniki rodzicielskie. Odmianą rekombinacji liniowej jest *rozszerzony operator rekombinacji liniowej*, który tworzy osobniki potomne w pobliżu rodziców z większym prawdopodobieństwem, a osobniki oddalone od rodziców z prawdopodobieństwem mniejszym [111]:

$$x'_l = x_{\alpha,l} + s \cdot d \cdot a \frac{x_{\beta} - x_{\alpha}}{\|x_{\beta} - x_{\alpha}\|}, \quad (3.36)$$

gdzie  $d$  – zakres rekombinacji,  $s$  – zmienna losowa o rozkładzie dwupunktowym  $\{-1,1\}$  (dla rozkładu równomiernego powstaje *rekombinacja nieukierunkowana* dla rozkładu nierównomiernego *rekombinacja ukierunkowana*),  $a$  – parametr:

$$a = 2^{-r \cdot k_p}, \quad (3.37)$$

gdzie  $r$  – realizacja zmiennej losowej o rozkładzie równomiernym na przedziale  $[0,1]$ ,  $k_p$  – precyzja rekombinacji (minimalny krok rekombinacji).

W wyniku działania krzyżowania arytmetycznego możemy również uzyskać dwóch potomków z każdej pary rodziców, posługując się następującym wzorem:

$$\begin{aligned} x'_{\alpha,l} &= a \cdot x_{\alpha,l} + (1-a) \cdot x_{\beta,l} \\ x'_{\beta,l} &= (1-a) \cdot x_{\alpha,l} + a \cdot x_{\beta,l} \end{aligned} \quad (3.38)$$

gdzie  $a$  – parametr krzyżowania, jeśli parametr  $a$  jest stały to operator nazywany jest *krzyżowaniem jednorodnym*, natomiast jeśli jest zmienny (zależny od wieku populacji), operator nazywany jest *krzyżowaniem niejednorodnym*. J. Schwefel nazywał ten operator *krzyżowaniem natychmiastowym* [119], natomiast w algorytmach genetycznych operator ten znany jest jako *gwarantowane średnie krzyżowanie* [52] (dla  $a=0.5$ ).

Odmianą krzyżowania arytmetycznego jest *krzyżowanie max-min arytmetyczne* [80], gdzie w wyniku krzyżowania dwójki osobników rodzicielskich otrzymujemy cztery osobniki potomne, z których następnie wybierane są dwa najlepsze (o największej wartości funkcji przystosowania):

$$\begin{aligned} x'_{\alpha,l} &= a \cdot x_{\alpha,l} + (1-a)x_{\beta,l} \\ x'_{\beta,l} &= (1-a)x_{\alpha,l} + a \cdot x_{\beta,l} \\ x'_{\gamma,l} &= \min(x_{\alpha,l}, x_{\beta,l}) \\ x'_{\delta,l} &= \max(x_{\alpha,l}, x_{\beta,l}) \end{aligned} \quad (3.39)$$

gdzie  $a$  – parametr krzyżowania.

Powyższe przykłady nie wyczerpują wszystkich możliwości implementacji operatorów. Więcej informacji na ten temat można znaleźć m.in. w publikacji Z. Michalewicz [119].

### 3.4 Programowanie ewolucyjne

Stosunkowo młodą kategorię (pomimo jej powstania w 1966r.) stanowi *programowanie ewolucyjne*, zapoczątkowane przez L. Fogla [66]. Mechanizmy ewolucyjne zostały po raz pierwszy zastosowane do poszukiwania automatu o skończonej liczbie stanów. Każdy chromosom prezentował jeden automat skończony (jedno rozwiązanie zadania). Celem było odnalezienie automatu, który będzie w stanie najlepiej „przewidywać” przyszłe zachowanie się otoczenia. Więcej szczegółów na ten temat można znaleźć w pracach [62] i [63], [64]. W [66] można znaleźć m.in. przykład zastosowania programowania ewolucyjnego do podziału ciągu liczb całkowitych na pierwsze i złożone, a w [65] zastosowanie do zarządzania dawkami leków sterujących ciśnieniem krwi podczas operacji chirurgicznej.

### 3.5 Podsumowanie

Opisane w podrozdziałach 3.2 i 3.3 algorytmy genetyczne i strategie ewolucyjne rozwijały się równolegle. U ich podstaw znajdowały się odmienne zastosowania i przeznaczenie. Ich obecna postać w zastosowaniach znacznie odbiega od pierwowzorów i coraz trudniej jest jednoznacznie wyznaczyć granicę pomiędzy nimi, ze względu na podobieństwo kodowania (algorytmy genetyczne kodowane zmiennoprzecinkowo) i wspólne operatory genetyczne, zmienną liczebność populacji (w szczególności algorytmy genetyczne o zmiennej liczbie osobników).

Ciekawe porównanie obu technik, opracowane pod kątem strategii ewolucyjnych można znaleźć w [84], a porównania niektórych wybranych cech w [5] i [119].

W niniejszej rozprawie pominięto szczegółowe opisy zaawansowanych technik, takich jak choćby algorytmy genetyczne ze zmienną liczbą osobników w populacjach, algorytmy genetyczne i strategie ewolucyjne, w których wykorzystuje się mechanizmy nisz i gatunków, programowanie ewolucyjne i genetyczne, diploidalny aparat genetyczny czy determinację płci, ponieważ nie mają one zastosowania w mechanizmach zaimplementowanych w celu wykazania tez postawionych w niniejszej rozprawie. Wyczerpujące informacje o tych technikach można znaleźć w [70] i [119].





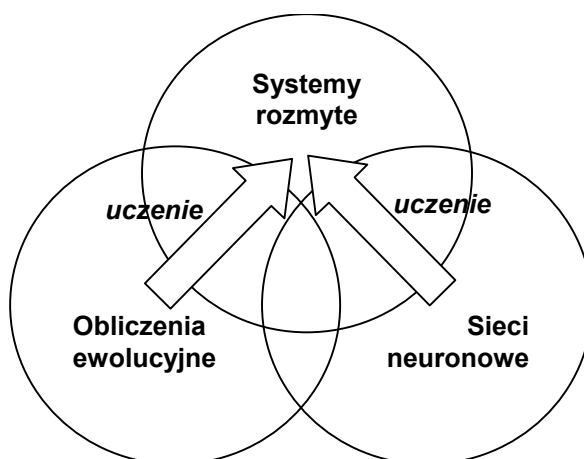
## Systemy ewolucyjno-rozmyte

Dynamiczny rozwój systemów rozmytych opartych na bazach wiedzy złożonych z rozmytych reguł warunkowych **if-then**, przy jednoczesnym kłopotcie z definiowaniem reguł dla tych baz przyspieszyły w latach 90 ubiegłego stulecia poszukiwania metod mogących zastąpić doświadczenie ekspertów, którzy stanowili dotychczas jedyne źródło informacji. Hybrydowe połączenie systemów rozmytych oraz mechanizmów ewolucyjnych pozwoliło na opracowanie metod, które cechują się zdolnościami do uczenia się na podanych przykładach i do uogólniania posiadanej wiedzy na przypadki im wcześniej nieznane.

Największa konkurencja systemów ewolucyjno-rozmytych – systemy neuronowo-rozmyte [68], [97], [113], [126], [154] w klasycznej formie cechują się słabym uogólnianiem posiadanej wiedzy. W przypadku wielomodalnych funkcji celu, stosowane w nich metody optymalizacji gradientowej prowadzą do zbieżności do minimum lokalnego. Dopiero ostatnie prace, łączące elementy sieci neuronowych z algorytmami globalnej optymalizacji, takimi jak symulowane wyżarzanie [43], wyżarzanie deterministyczne [150], [151], czy nawet zastosowanie losowej zmiany kierunków poszukiwań [48], pozwoliły na przezwycięzenie kłopotów związanych z efektem „przeuczenia” (ang. *Overlearning*). Więcej informacji o zastosowaniu metod globalnej optymalizacji w procesie uczenia sztucznych sieci neuronowych można znaleźć w [48] oraz [111]. Systemy neuronowo-rozmyte doczekały się wielu modeli połączeń części rozmytej oraz sieci neuronowej. Przykładowo, w pracy [89] przedstawiono uczenie sieci neuronowej na podstawie zarówno wiedzy eksperta, zawartej w postaci rozmytych reguł warunkowych if-then, jak i danych numerycznych. Inną metodą jest generowanie reguł rozmytych na podstawie wag połączeń w sieci neuronowej. Jedną z pierwszych prac, w której wykorzystano wielowarstwowy perceptron do generowania reguł rozmytych jest praca [123]. Stosunkowo wiele rozwiązań bazuje na sieciach neuronowych o radialnych funkcjach bazowych (RBFN, ang. *Radial Basis Function Networks*). Wykorzystuje się również fakt równoważności pomiędzy tym typem sieci neuronowych, a pewnymi typami systemów rozmytych [95], [97].

Stosunkowo najmniej prac poświęconych jest połączeniu systemów rozmytych, sieci neuronowych i obliczeń ewolucyjnych w jednej hybrydzie, co wydaje się być naturalnym następstwem prowadzonych obecnie badań i przyszłościowym kierunkiem rozwoju.

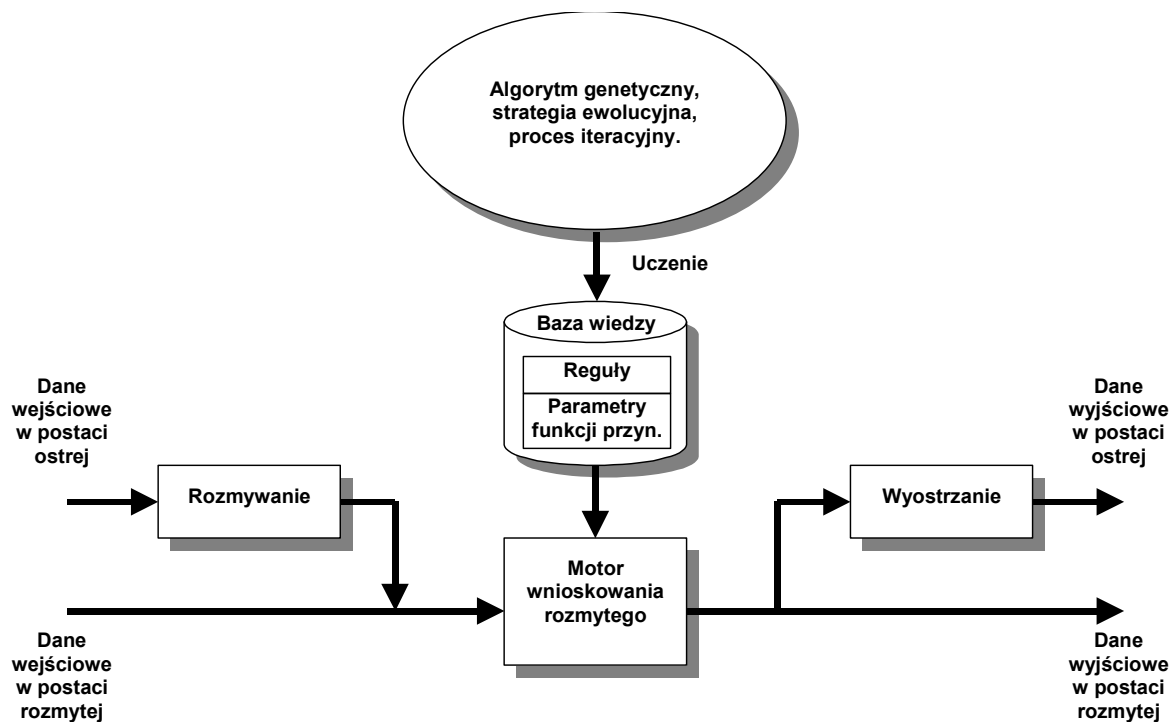
O. Cordón i wsp. w [32] zestawili przegląd rozwiązań z zakresu systemów ewolucyjno-rozmytych, które pojawiły się na przestrzeni ostatniego dziesięciolecia.



Rysunek 4.1: Hybrydowe połączenia metod obliczeń miękkich.

System genetyczno-rozmyty (ewolucyjno-rozmyty) [4], [35], [81], [133], [157], stanowi połączenie systemu wnioskowania rozmytego i systemu uczenia opartego na metodach ewolucyjnych. Korzystając z algorytmów genetycznych i strategii ewolucyjnych, rozwiązywane są zagadnienia od prostego dostrajania parametrów istniejącego systemu rozmytego do kompletnie zautomatyzowanego generowania całego systemu na podstawie dostarczonych danych uczących [54]. W literaturze najczęściej spotykane są systemy genetyczno-rozmyte (GFS, ang. *Genetic Fuzzy Systems*) lub ewolucyjno-rozmyte (EFS, ang. *Evolutionary Fuzzy Systems*), gdzie część rozmyta jest klasycznym systemem wnioskowania rozmytego, opartym na bazie wiedzy (GFRBS, ang. *Genetic Fuzzy Rule-based Systems*) [35]. Uczenie systemu można zinterpretować jako wyznaczanie reguł na podstawie zaobserwowanych związków przyczynowo-skutkowych, natomiast zdolność systemu do uogólniania posiadanej wiedzy należy traktować jako wnioskowanie polegające na wyznaczeniu przewidywanych skutków dla zaistniałych przyczyn. Tak więc zastosowanie metod ewolucyjnych do systemów tej klasy sprowadza się do rozwiązania następujących zadań [111]:

- wyodrębnienia zbioru reguł **if-then**, pokrywających zbiór danych uczących wraz ze wstępnym oszacowaniem ich parametrów lub też dołączenie do istniejących reguł podanych przez eksperta-człowieka reguł wygenerowanych automatycznie,
- redukcji istniejącej bazy reguł, celem usunięcia redundancji i ew. sprzeczności,
- optymalizacji bazy wiedzy poprzez dokładne, lokalne dostrajanie parametrów reguł z wykorzystaniem zbioru uczącego.



Rysunek 4.2: Zależność pomiędzy częścią ewolucyjną i rozmytą systemu ewolucyjno-rozmytego.

W literaturze znane są trzy podejścia do rozwiązania zagadnienia generowania rozmytych reguł warunkowych **if-then** na podstawie danych numerycznych [32], [78], [79], [111]:

- Podejście Michigan [86], w którym fenotyp osobnika stanowi pojedynczą regułę rozmytą (najczęściej MISO), a po stronie genotypu znajdują się wartości opisujące zbiory rozmyte przesłanek i konkluzji tej reguły. Baza reguł systemu stanowiącego rozwiązanie zadania jest tworzona z najlepszych osobników.
- Podejście Pittsburgh [162], w którym fenotyp osobnika stanowi kompletną bazę wiedzy systemu rozmytego (również najczęściej jest to system MISO), a po stronie genotypu znajduje się komplet wartości opisujących wszystkie zbiory rozmyte przesłanek i konkluzji dla każdej reguły bazy wiedzy. Bazę wiedzy rozwiązania stanowi osobnik o najlepszym przystosowaniu.
- Uczenie iteracyjne [79], [175], w którym kodowanie osobników jest takie jak w podejściu Michigan, ale w przeciwieństwie do tego podejścia, algorytm jest uruchamiany wielokrotnie i za każdym razem jest wybierany wyłącznie najlepszy osobnik. Proces ten trwa, dopóki wszystkie dane ze zbioru uczącego nie zostaną pokryte przez wygenerowane reguły.

Najbardziej dojrzałym podejściem wydaje się rozwiązanie, w którym każdym chromosom osobnika zawiera komplet danych opisujących system rozmyty, a więc liczbę reguł, typy funkcji przynależności i parametry określające zbiory rozmyte w funkcjach przynależności

zdefiniowanych dla wejścia i wyjścia, jednakże wielowymiarowość takiego podejścia powoduje ogromną złożoność obliczeniową. Równie obiecująca wydaje się metoda uczenia iteracyjnego z wykorzystaniem mechanizmów pokrywania danych, szczegółowo opisana w [33], [79] i [175], która została zaadoptowana na potrzeby przeprowadzenia badań w niniejszej rozprawie doktorskiej.

Z punktu widzenia optymalizacyjnego, zadanie automatycznego znalezienia systemu rozmytego sprowadza się do sparametryzowania bazy wiedzy systemu rozmytego i odnalezienia optymalnych wartości tychże parametrów, przy zastosowanym kryterium jakościowym. Parametry bazy wiedzy tworzą przestrzeń rozwiązań, w której wykonywane są poszukiwania. Pierwszym istotnym etapem jest identyfikacja przedmiotu poszukiwań. Przestrzeń poszukiwań nie jest jednolita, lecz raczej może być postrzegana jako połączenie elementów opisujących system rozmyty. W zależności od przyjętego modelu systemu rozmytego oraz stanu posiadanej wiedzy, dostarczonej w inny sposób (np. od eksperta-człowieka), zadania będą wymagały odmiennego podejścia do kodowania, zastosowanych operatorów i przyjętej strategii. Kryterium jakościowym najczęściej jest błąd wyniku generowanego przez uzyskany system w porównaniu do danych uczących (np. błąd średniokwadratowy). Należy jednak zauważyć, że jest to kryterium słuszne, lecz możliwe do zastosowania jedynie dla przypadków, w których ocenie podlega cały system rozmyty, jak to ma miejsce w przypadku Pittsburgh. W przypadku zastosowania podejścia Michigan lub uczenia iteracyjnego, nie jest możliwe bezpośrednie zastosowanie tego kryterium, dlatego też do oceny reguł stosuje się kryteria złożone, oceniające poszczególne elementy reguły, takie jak przesłanka, konkluzja czy zależności pomiędzy ocenianym elementem i elementami innymi.

Operacje odnajdywania bazy wiedzy, redukcji bazy wiedzy oraz lokalnego dostrajania mogą być postrzegane jako procesy niezależne i tak też można je traktować, jednakże ostatecznym dążeniem jest stworzenie systemu ewolucyjno-rozmytego, który wykorzystuje w sposób zautomatyzowany wszystkie te techniki. Na wejściu powinny być podawane dane numeryczne (uczące), na podstawie których system powinien uczyć się, wraz ze zdolnością do uogólniania posiadanej wiedzy na przypadki, których nie zawarto w danych uczących. System taki powinien być uniwersalny i nie powinien zawierać w swojej implementacji żadnej wiedzy heurystycznej związanej z numerycznymi danymi uczącymi. W kolejnych podrozdziałach znajduje się przegląd znanych rozwiązań dotyczących procesów generowania bazy wiedzy, jej redukcji i lokalnego dostrajania.

## 4.1 Strojenie bazy danych systemu rozmytego

W procesie dostrajania funkcji przynależności pojedynczy osobnik reprezentuje całą bazę parametrów funkcji przynależności związanych z wartościami lingwistycznymi dotyczącymi rozpatrywanego systemu rozmytego. Najczęściej stosowane funkcje przynależności to funkcje trójkątne (zarówno symetryczne jak i niesymetryczne), trapezoidalne oraz gaussowskie. Struktura chromosomu jest odmienna dla systemów rozmytych z ustaloną i zmienną bazą danych [32]. Liczba parametrów dla każdej funkcji zależy od zastosowanego typu funkcji przynależności i zwykle nie przekracza czterech wartości zmiennoprzecinkowych (np. dla trapezoidalnej funkcji przynależności). Najczęściej stosowane jest kodowanie zmiennoprzecinkowe [109], choć znane są również rozwiązania z kodowaniem binarnym [158].

W przypadku dostrajania modelu lingwistycznego systemu rozmytego [36], chromosom koduje całość rozmytego podziału przestrzeni wejść i wyjścia. Zmianom podlega również całość chromosomu tak, aby zachować semantykę bazy reguł. Przy tym podejściu zakłada się ustaloną liczbę terminów lingwistycznych dla każdego wejścia (może być różna dla różnych wejść systemu), dzięki czemu otrzymujemy stałą długość chromosomu (dla systemów klasy Mamdaniego), co znacznie ułatwia implementację metody. W przypadku zastosowania metody do systemów TSK [107] stała długość chromosomów nie jest przeszkodą, jeśli zostanie założona maksymalna liczba zmiennych lingwistycznych dla wejścia.

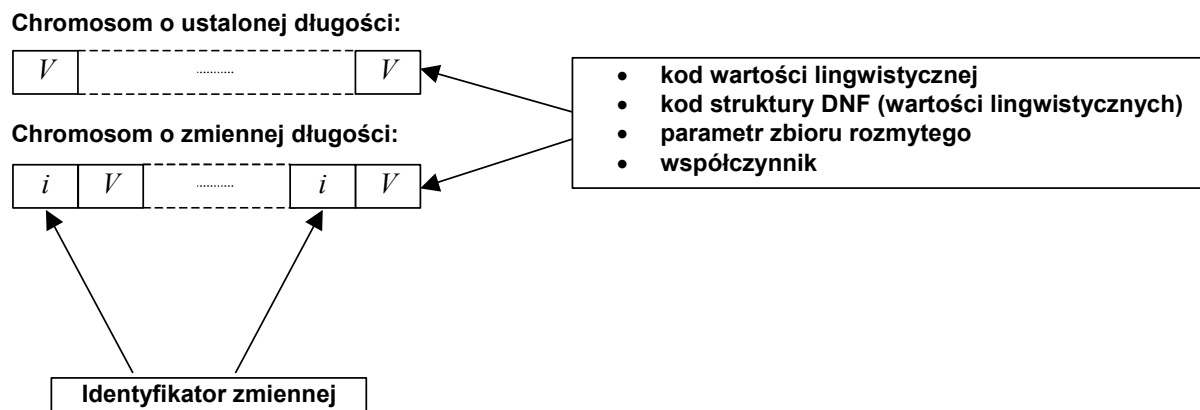
Jeśli strojone są bezpośrednio zbiory rozmyte przesłanek i konkluzji, operację można uznać za szczególny przypadek uczenia bazy wiedzy.

## 4.2 Uczenie bazy reguł

Uczenie bazy reguł dotyczy jedynie systemów lingwistycznych z predefiniowanym zbiorem terminów, ponieważ w przypadku pozostałych systemów, dopasowywanie reguł wiąże się ze zmianami w funkcjach przynależności. W przypadku uczenia bazy reguł istnieją trzy sposoby podejścia do tego zagadnienia, analogicznie jak to opisano w części początkowej rozdziału 4. Podejście Michigan do rozważanego zagadnienia zostało zastosowane i szczegółowo omówione w pracach [16], [92], [172], podejście Pittsburgh w [83], [137], [170], a uczenie iteracyjne w [36], [42] i [73].

Znane metody kodowania osobników to reprezentacja macierzowa [170], tabela decyzyjna [137] oraz najczęściej wykorzystywana, lista reguł [73], [83], [172]. Kodowanie macierzowe i tabela decyzyjna nadają się do zastosowań o małej (1-3) liczbie wejść systemu rozmytego, oraz wymagają podejścia Pittsburgh. Lista reguł jest znacznie bardziej uniwersalna i nie wprowadza ograniczeń co do zastosowanego podejścia.

Kodowanie listy reguł najczęściej sprowadza się do łańcucha binarnego o stałej długości i dysjunkcyjnej postaci normalnej (DNF, ang. *Disjunctive Normal Form*) [73], [115], [172]. Kodowanie reguły w łańcuchu o stałej długości pozwala to na łatwe przejście od podejścia Michigan do podejścia Pittsburgh, za pomocą konkatenacji ciągów kodowych (genotypów) osobników prezentujących pojedyncze reguły w reprezentacji Michigan [32]. Naturalnym podejściem do kodowania jest zastosowanie uporządkowanych par: (<id>, <wartość>) zakodowanych numerycznie. W przypadku stałej długości chromosomów, identyfikator może być pobrany jako pozycja wartości, na której występuje. Na poniższym rysunku przedstawiono graficznie metodę kodowania dla zagadnienia uczenia bazy reguł:



Rysunek 4.3: Kodowanie dla zadania uczenia bazy reguł.

W przypadku systemów klasy TSK,  $V$  może przyjmować wartości parametrów funkcji wyjścia. W pozostałych przypadkach  $V$  określa zmienne lingwistyczne lub bezpośrednie wartości opisujące zbiory rozmyte w przesłankach / konkluzjach. W niektórych przypadkach, dodatkowo do procesu uczenia reguł dołącza się proces strojenia funkcji przynależności prezentujących zmienne lingwistyczne [42].

### 4.3 Uczenie bazy wiedzy

Uczenie bazy wiedzy polega na jednoczesnym odnajdywaniu rozmytych reguł (uczenie bazy reguł) i wartości opisujących zbiory rozmyte zmiennych lingwistycznych (uczenie bazy danych) systemu rozmytego. Taka heterogeniczna przestrzeń poszukiwań znacznie zwiększa koszt obliczeniowy algorytmu genetycznego / ewolucyjnego. Charakter zadania sugeruje możliwość wykorzystanie chromosomów o zmiennej długości lub genomów złożonych z dwóch chromosomów, gdzie jeden z nich reprezentuje parametry bazy danych systemu rozmytego, a drugi bazę reguł. Podobnie jak w poprzednich przypadkach, znane są trzy podejścia do rozwiązania tego zagadnienia – podejście Michigan [130], [174], Pittsburgh [11], [24], [107], [115], [129] oraz iteracyjne uczenie reguł [36], [38]. Proces uczenia bazy wiedzy można również zastosować w systemie rozmytym klasy Mamdaniego, w którym zastosowano bezpośrednie kodowanie wartości opisujących zbiory rozmyte przesłanek i konkluzji [24], [36], [38], [174], w systemie lingwistycznym Mamdaniego [115], [129] oraz TSK [37], [107], [127]. W związku z koniecznością zakodowania w pojedynczym osobniku zarówno bazy reguł jak i parametrów funkcji przynależności, możliwe jest wykorzystanie zasad kodowania przedstawionych w podrozdziałach 4.1 i 4.2, łącznie. Zastosowane w takim wypadku operatory genetyczne (w szczególności operator krzyżowania) muszą uwzględniać obie zakodowane cechy. W literaturze znane są różne rozwiązania tego problemu – traktowanie chromosomu jako jednej całości, bez rozdzielania struktur bazy reguł i parametrów zbiorów rozmytych [107], rozdzielanie bazy reguł i parametrów oraz niezależne, równoległe zastosowanie operatora do obu struktur [129], lub krzyżowanie szeregowe, w którym wymiana materiału genetycznego dokonuje się w jednej ze struktur, co pociąga za sobą konieczność wprowadzenia korekt w strukturze drugiej [115]. Zastosowanie operatora mutacji również może pociągać za sobą konieczność wprowadzenia zmian w zależnej części chromosomu.

## 4.4 Inne techniki

Programowanie genetyczne jest jedną z rzadziej wykorzystywanych technik ewolucyjnych do automatycznego generowania systemów rozmytych [1], [30], [69], [82]. Zbiór reguł rozmytych jest prezentowany za pomocą drzew binarnych, naturalnej struktury dla tego typu obliczeń miękkich. Stosowane są również hybrydy programowania genetycznego z algorytmami genetycznymi [155] lub symulowanym wyżarzaniem [156].

W przypadku problemów wielowymiarowych, podczas identyfikacji systemów rozmytych wykładniczo rośnie liczba rozmytych reguł warunkowych. Redukcja bazy reguł staje się istotnym elementem projektowania optymalnych systemów rozmytych. Pomimo istnienia metod optymalizacji liczby reguł bazujących m.in. na sieciach neuronowych oraz metod analitycznych, zastosowanie algorytmów genetycznych ze względu na ich zdolność do optymalizacji globalnej przyniosło na tym polu spore sukcesy. Przykłady takich zastosowań można znaleźć w pracach [36], [75], [90], [93], [152]. Alternatywnym podejściem do problemu zbyt dużej wymiarowości jest analiza i usuwanie zbędnych cech z przestrzeni poszukiwań. Do tego celu również zastosowano algorytmy genetyczne [25], [87], [106], [161]. Zmniejszanie komplikacji zadania może polegać również na redukcji liczby zmiennych lingwistycznych w przesłankach rozmytych reguł warunkowych. Zredukowana w ten sposób baza reguł cechuje się różną długością przesłanek dla różnych reguł [74].

Innowacyjne podejście z jednoczesnym wykorzystaniem dwóch mechanizmów ewolucyjnych zostało przedstawione w pracach [39], [40], gdzie jeden z algorytmów genetycznych służy do odnalezienia bazy danych systemu rozmytego (parametrów funkcji przynależności), a drugi algorytm na podstawie tych informacji generuje bazę reguł. Wartość przystosowania osobników dla obu algorytmów jest ustalana dla wyniku działania obu algorytmów łącznie.

Poza klasycznymi strukturami zawierającymi typowe reguły **if-then** stosowane są również techniki zaawansowane, takie jak:

- Użycie reguł z podwójną konkluzją [33], gdzie algorytm genetyczny zastosowany został jako selektor poprawiający dokładność systemu rozmytego.
- Dobieranie wag reguł rozmytych w klasyfikatorze dla przypadku *konsylium ekspertów*, z wykorzystaniem metod ewolucyjnych [91].
- Zastosowanie algorytmów genetycznych do optymalizacji bazy wiedzy hierarchicznego modelu rozmytego [41].
- Wykorzystanie mechanizmów równoległych algorytmów genetycznych i mechanizmów nisz oraz algorytmów koewolucyjnych, w których rozwiązania są poszukiwane poprzez jednoczesne ewoluowanie np. trzech populacji, jak to zaprezentowano w pracy [27], z jednoczesnym zapewnieniem mechanizmów migracji z jednej populacji do drugiej. Podobne podejście zaprezentowano również w [149]. W pracy [136] zaproponowano zastosowanie koewolucyjnych algorytmów kooperujących do uczenia bazy reguł i dostrajania wartości funkcji przynależności. Inną możliwością jest zastosowanie algorytmów koewolucyjnych do jednoczesnego wyboru reguł i ustalania ich wag, jak to zaprezentowano w pracy [3].

Powyższe przykłady dotyczą systemów rozmytych, korzystających z bazy reguł. W literaturze znane są przypadki wykorzystania algorytmów genetycznych i strategii ewolucyjnych do projektowania innych systemów, które nie przechowują wiedzy w postaci warunkowych, rozmytych reguł **if-then**. Są to najczęściej systemy genetyczno-rozmyto-neuronowe. Algorytmy genetyczne bywają również stosowane do optymalizacji algorytmów podziału, w szczególności algorytmu *rozmytych c-średnich* (FCM, ang. *fuzzy c-means*) [6], [76], [125], [173], [176], lub też bezpośrednio wykorzystywane jako metoda podziału [22].

## 4.5 Przykłady systemów ewolucyjno-rozmytych

W poniższych podrozdziałach zawarto przykładowe rozwiązania generowania systemów rozmytych z bazami wiedzy w postaci rozmytych reguł warunkowych **if-then**, za pomocą mechanizmów ewolucyjnych.

### 4.5.1 System ewolucyjno-rozmyty Mamdaniego-Assiliana oparty na trójetapowej metodzie uczenia

W pracach [36], [78], [79], [80] oraz [111] przedstawiono projekt trójetapowej metody uczenia systemu ewolucyjno-rozmytego. Obejmuje on przypadki zarówno z ustaloną jak i zmienną bazą danych systemu rozmytego. Proces generowania systemu rozmytego o nieustalonej bazie danych został przedstawiony poniżej. Proces ten składa się z trzech głównych etapów:

1. Pokrycie zbioru danych rozmytymi regułami **if-then** celem uzyskania bazy wiedzy systemu rozmytego.
2. Redukcja uzyskanej w pierwszym etapie nadmiarowej bazy wiedzy.
3. Dostrojenie parametrów bazy wiedzy z zastosowaniem mechanizmów ewolucyjnych.

W części rozmytej zastosowano system rozmyty Mamdaniego-Assiliana o bazie wiedzy w postaci rozmytych reguł warunkowych:

$$\mathcal{R} = \left\{ R^{(i)} \right\}_{i=1}^I = \left\{ \text{jeżeli } \bigwedge_{n=1}^N (x_{0,n} \text{ jest } A_n^{(i)}) \text{ to } (Y \text{ jest } B^{(i)}) \right\}_{i=1}^I. \quad (4.1)$$

W przesłankach i konkluzji stosowano trójkątne funkcje przynależności, jako operator *t-norma* operację minimum, koniunkcyjną interpretację reguł oraz jako operator agregacji ważoną sumę wyostrzonych wartości wyjściowych liczonych dla poszczególnych reguł z zastosowaniem operatora COG.

#### Etap 1

Pokrywanie wykonano iteracyjnie. W każdym kroku pętli realizującej algorytm pokrywania odnajdywana jest jedna reguła rozmyta. Operacja ta wykonywana jest z wykorzystaniem algorytmu genetycznego o kodowaniu zmiennoprzecinkowym, opisanego w dalszej części tego punktu. Po zakończeniu działania algorytmu genetycznego następuje wyliczenie i uzupełnienie



pokrycia danych uczących o wartości pokrycia wyliczone dla odnalezionej reguły. Dane uczące, których łączne pokrycie (przez wszystkie dotychczas wygenerowane reguły) jest większe od założonej wartości, zostają usunięte ze zbioru uczącego. Pętla realizująca algorytm pokrywania kończy swe działanie, gdy wszystkie dane ze zbioru uczącego zostaną pokryte w założonym stopniu, stanowiącym parametr metody.

Wynikiem działania algorytmu genetycznego jest pojedyncza reguła rozmyta, o największej wartości w sensie przyjętego kryterium jakości. Przyjęto kodowanie, w którym każdy osobnik reprezentuje pojedynczą regułę rozmytą **if-then** [111]:

$$\tilde{x}_m^k = [a_{m,1}^k, b_{m,1}^k, c_{m,1}^k, \dots, a_{m,N}^k, b_{m,N}^k, c_{m,N}^k, a_m^k, b_m^k, c_m^k]^T, \quad (4.2)$$

gdzie  $m$ -numer osobnika,  $k$ -numer pokolenia,  $a_{m,n}, b_{m,n}, c_{m,n}$ -parametry trójkątnej funkcji przynależności przesłanki  $n$ -tego wejścia,  $a, b, c$ -parametry trójkątnej funkcji przynależności konkluzji (Rysunek 2.1).

Algorytm generowania populacji początkowej tworzy osobniki o funkcjach przynależności należących do odpowiadających im przedziałów zmienności wartości wejściowych oraz wyjścia, nieznacznie rozszerzonych. Algorytm ten został szczegółowo opisany w pracy [111]. W strategii zastosowano następujące operatory genetyczne:

- Max-min-arytmetyczną metodę krzyżowania [80].
- Mutację nierównomierną [119].

Algorytm genetyczny zatrzymuje się po osiągnięciu  $K$  pokoleń. Nowe pokolenie bazowe jest tworzone z pokolenia bazowego i potomnego, z wykorzystaniem selekcji elitarniej.

Istotnym elementem jest zastosowana złożona funkcja oceny osobników-reguł. Na ostateczną ocenę składają się następujące oceny cząstkowe [34], [111]:

1. Średnia kompatybilność ocenianej reguły z danymi ze zbioru uczącego.
2. Średni stopień pokrycia danych ze zbioru uczącego przez ocenianą regułę.
3. Stopień dopasowania konkluzji.
4. Szerokość nośników przesłanek i konkluzji - preferowane są reguły o wąskich nośnikach w funkcjach przynależności przesłanek i konkluzji, w porównaniu z przedziałami zmienności argumentów wejściowych i wyjścia.
5. Symetryczność funkcji przynależności – preferowane są symetryczne funkcje przynależności.
6. Unikalność odnalezionej reguły w porównaniu z wcześniej odnalezionymi regułami.

Końcowa ocena osobnika jest wyliczana jako iloczyn arytmetyczny ocen od 1 do 5 lub od 1 do 6 [34], [111].

Alternatywnie dla systemów rozmytych Mamdaniego-Assiliana z ustaloną bazą danych opracowano odmienny sposób postępowania. Baza wiedzy takiego systemu przyjmuje następującą postać:

$$\mathcal{R} = \{R^{(i)}\}_{i=1}^I = \left\{ \text{jeżeli } \bigwedge_{n=1}^N (x_{0,n} \text{ jest } L_{j_n}^{(i)}) \text{ to } (Y \text{ jest } L_{j_y}^{(i)}) \right\}_{i=1}^I, \quad (4.3)$$

gdzie  $j_n$  – indeks zastosowanej wartości lingwistycznej dla  $n$ -tej zmiennej wejściowej,  $j_{N+1}$  – indeks zastosowanej wartości lingwistycznej dla zmiennej wyjściowej,  $L$  – zmienna lingwistyczna, charakteryzowana za pomocą zbioru przynależności  $\mu_L$ . Zmienne lingwistyczne  $L_{j_n}^{(i)}$  oraz  $L_{j_y}^{(i)}$  przynależą do odpowiadających im zbiorów terminów lingwistycznych (odpowiednio dla  $n$ -tego wejścia oraz dla wyjścia). Ponieważ wartości zmiennych lingwistycznych reprezentują parametry odpowiadających im zbiorów rozmytych, wymagana jest specjalna, dualna konstrukcja chromosomu. Chromosom reprezentujący  $\rho$ -tego osobnika-regułę systemu z ustaloną bazą danych składa się z dwóch elementów:

$$\tilde{x}_\rho = [\underline{C}_{\rho,1}^T, \underline{C}_{\rho,2}^T]^T, \quad (4.4)$$

gdzie  $\underline{C}_{\rho,1}, \underline{C}_{\rho,2}$  mają następującą postać:

$$\begin{cases} \underline{C}_{\rho,1} = [j_{\rho,1}, j_{\rho,2}, \dots, j_{\rho,N}, j_{\rho,y}]^T, \\ \underline{C}_{\rho,2} = [a_{\rho,L_{j_1}}, b_{\rho,L_{j_1}}, c_{\rho,L_{j_1}}, a_{\rho,L_{j_2}}, b_{\rho,L_{j_2}}, c_{\rho,L_{j_2}}, \dots, a_{\rho,L_{j_N}}, b_{\rho,L_{j_N}}, c_{\rho,L_{j_N}}, a_{\rho,L_{j_y}}, b_{\rho,L_{j_y}}, c_{\rho,L_{j_y}}]^T. \end{cases} \quad (4.5)$$

W pierwszej części chromosomu kodowane są identyfikatory zmiennych lingwistycznych  $\rho$ -tego chromosomu. Pozycja  $j_{\rho,n}$  oznacza indeks terminu lingwistycznego ze zbioru terminów związanych z wejściem  $n$ ,  $j_{\rho,y}$  odnosi się do wyjścia. W drugiej części chromosomu znajdują się związane z terminami lingwistycznymi parametry funkcji przynależności. Również w tym przypadku zastosowano specjalny sposób generowania pokolenia startowego, szczegółowo opisany w [111]. Powyższy sposób kodowania chromosomów wymusił opracowanie odmiennego od stosowanego w przypadku systemu rozmytego o zmiennej bazie danych operatora krzyżowania. Zastosowanie krzyżowania max-min-arytmetycznego ma sens jedynie w przypadku, gdy pierwsze części chromosomów osobników rodzicielskich są identyczne. W przeciwnym przypadku niezbędne jest wymieszanie elementów chromosomu tak, aby relacje pomiędzy pierwszą i drugą częścią chromosomów potomnych były identyczne z tymi, które są zawarte w osobnikach rodzicielskich. Operacja ta jest podobna do binarnego krzyżowania jednopunktowego (stosowanego w klasycznym algorytmie genetycznym), wykonywanego jednocześnie na pierwszej i drugiej części chromosomu. Więcej informacji na ten temat oraz przykład obrazujący zastosowaną metodę krzyżowania można znaleźć w pracy [111].

Modyfikacji uległ również operator mutacji. Jeżeli mutacji poddawany jest element należący do drugiej części chromosomu, stosowana jest mutacja nierównomierna [119], w przeciwnym wypadku można zastosować mutację opisaną w [111], która zmienia identyfikator zmiennej lingwistycznej, co oczywiście pociąga za sobą konieczność zmian w drugiej części chromosomu i wymianę parametrów opisujących dotychczasowy zbiór rozmyty zmiennej lingwistycznej na parametry nowe, opisujące rozmyty zbiór przynależności nowej zmiennej lingwistycznej.

W algorytmie zastosowano stochastyczne próbkowanie oraz selekcję elitarną, podobnie jak to miało miejsce w przypadku systemu z nieustaloną bazą danych. Po wykonaniu algorytmu genetycznego zalecane jest poddanie części populacji końcowej operacjom optymalizującym bazę danych, w formie strategii ewolucyjnej. Szczegóły można znaleźć w [111]. Zastosowana funkcja przynależności ma prostszą formę od stosowanej w przypadku systemu rozmytego z nieustaloną bazą danych i pomija oceny cząstkowe związane bezpośrednio z parametrami funkcji przynależności. Wartość przystosowania osobnika jest wyliczana jako iloczyn arytmetyczny ocen cząstkowych 1., 2., 3. i 6 opisanych powyżej.

## Etap 2

Ponieważ zastosowane w pierwszym etapie mechanizmy (niezależnie od tego, czy wykorzystywano system rozmyty z ustaloną czy nieustaloną bazą danych) powodują wygenerowanie nadmiarowej liczby reguł, niezbędne jest zredukowanie ich liczby. W tym celu zastosowano prosty algorytm genetyczny. Każdy osobnik reprezentuje całą bazę reguł, a jego  $i$ -ta pozycja mówi, czy odpowiadająca tej pozycji reguła znajduje się w tej bazie (gdy wartość pozycji jest równa 1) czy też została z niej usunięta (wartość pozycji równa 0). Każdy ciąg jest  $I$ -bitowy, gdzie  $I$  określa liczbę reguł wygenerowaną w pierwszym etapie metody. Wszystkie za wyjątkiem jednego osobniki pokolenia startowego uzyskiwane są w sposób losowy, natomiast jeden osobnik reprezentuje pełną bazę reguł, uzyskaną w pierwszym etapie metody. Zastosowane krzyżowanie dwupunktowe i typową dla prostego algorytmu genetycznego mutację. Funkcja przynależności sprawdza zarówno jakość systemu rozmytego reprezentowanego przez osobnika, która jest mierzona błędem średniokwadratowym, jak również kompletność ocenianej bazy reguł. Szczegółowy opis funkcji można znaleźć w pracy [111].

## Etap 3

Kolejnym etapem jest dokładne lokalne dostrojenie parametrów trójkątnych funkcji przynależności dla bazy reguł uzyskanej w drugim etapie metody. Podobnie jak to ma miejsce w pierwszym etapie, zastosowano algorytm genetyczny ze zmiennoprzecinkową reprezentacją rozwiązań, przy czym pojedynczy osobnik reprezentuje pełną bazę wiedzy, uzyskaną w drugim etapie metody i zawiera wszystkie parametry rozmytych zbiorów przesłanek i konkluzji:

$$\tilde{\mathbf{x}}_m^k = [\underline{\mathbf{c}}_m^{k,(1)T}, \underline{\mathbf{c}}_m^{k,(2)T}, \dots, \underline{\mathbf{c}}_m^{k,(I)T}]^T, \quad (4.6)$$

gdzie  $I$ -liczba reguł po redukcji bazy reguł wykonanej w drugim etapie metody,  $m$ -numer osobnika,  $k$ -numer pokolenia. Elementy chromosomu złożone są z następujących wektorów wartości zmiennoprzecinkowych:

$$\underline{\mathbf{c}}_m^{k,(i)} = [a_{m,1}^{k,(i)}, b_{m,1}^{k,(i)} c_{m,1}^{k,(i)}, a_{m,2}^{k,(i)}, b_{m,2}^{k,(i)} c_{m,2}^{k,(i)}, \dots, a_{m,N}^{k,(i)}, b_{m,N}^{k,(i)} c_{m,N}^{k,(i)}, a_m^{k,(i)}, b_m^{k,(i)} c_m^{k,(i)}]^T, \quad (4.7)$$

gdzie  $a_{m,n}^{k,(i)}, b_{m,n}^{k,(i)} c_{m,n}^{k,(i)}$  - parametry trójkątnej funkcji przynależności związanej z  $n$ -tym wejściem systemu,  $a_m^{k,(i)}, b_m^{k,(i)} c_m^{k,(i)}$  - parametry trójkątnej funkcji przynależności konkluzji,  $i$ -numer reguły. Podobnie jak w pierwszym etapie metody, zastosowano specjalny algorytm generowania pokolenia startowego. Chromosom jednego z osobników tego pokolenia zawiera parametry

funkcji przynależności zgodne z wygenerowanymi w pierwszym etapie metody, przy czym wyłącznie dla reguł pozostałych po redukcji bazy reguł w etapie drugim, natomiast wartości funkcji przynależności dla pozostałych osobników są generowane losowo, przy czym muszą się zawierać w określonych przedziałach zmienności. Algorytm ten został szczegółowo omówiony w pracy [111]. Zastosowano operatory genetyczne zgodne z opisanymi w pierwszym etapie metody, próbkowanie stochastyczne oraz selekcję elitarną. Jako warunek zatrzymania algorytmu genetycznego przyjęto osiągnięcie określonej liczby pokoleń.

Praca [111] przedstawia szczegółowy opis prezentowanej metody, jak również wskazuje wartości parametrów sterujących wykonaniem algorytmów genetycznych.

#### 4.5.2 System ewolucyjno-rozmyty TSK oparty na dwuetapowej metodzie uczenia

W pracy [37] zaproponowano dwuetapową metodę uczenia na podstawie danych numerycznych systemu rozmytego TSK. W pierwszym etapie metody generowano zgrubnie bazę wiedzy systemu TSK, która następnie w etapie drugim poddawana była lokalnemu dostrojeniu parametrów funkcji przynależności. Poszukiwana baza wiedzy systemu TSK ma następującą postać:

$$\mathcal{R} = \{R^{(i)}\}_{i=1}^I = \left\{ \text{jeżeli } \bigwedge_{n=1}^N (x_{0,n} \text{ jest } A_n^{(i)}) \text{ to } y^{(i)}(\underline{x}_0) = \underline{p}^{(i)T} \cdot \underline{x}' \right\}_{i=1}^I, \quad (4.8)$$

gdzie  $\underline{p}$  – wektor parametrów rozmytej reguły systemu TSK,  $\underline{x}'$  - rozszerzony wektor wejściowy, tak jak to opisano w podrozdziale 2.3.1 dla systemu z parametryzowanymi konkluzjami.

Dla reguł systemu rozmytego TSK istotnym problemem jest oszacowanie zakresów zmienności parametrów wektora  $\underline{p}$ . Przyjęcie  $\mathcal{R}$  jako przestrzeni poszukiwań dla każdego parametru jest wysoce nieefektywne, dlatego też w pracy [37] zastosowano *kodowanie kątowe* z wykorzystaniem funkcji *arctan*. Poszczególne pozycje ciągu kodowego są przechowywane w postaci zakodowanej [111]:

$$\tilde{p}_n^{(i)} = \arctan(p_n^{(i)}). \quad (4.9)$$

Obliczenia ewolucyjne są wykonywane na tak zakodowanych parametrach, które mogą być rozkodowane za pomocą funkcji *tan*.

##### Etap 1

W pierwszym etapie wykorzystano rozmyty podział przestrzeni wejściowej. Autorzy pracy [37] wykorzystali podział siatkowy, choć możliwe są również inne metody, np. podział pochodzący od eksperta-człowieka lub wykorzystanie algorytmu rozmytego grupowania [48], [111]. Dla znanych wartości funkcji przynależności można wyliczyć poziom aktywacji każdej reguły dla każdego rekordu danych wejściowych ze zbioru uczącego. Pozwala to na wyznaczenie dla każdej reguły podzbioru zbioru uczącego, którego elementy aktywują regułę. W ten sposób ustalana jest liczba reguł w systemie. Dla tak powstałych podzbiorów stosowana jest strategia ewolucyjna  $(\mu+\lambda)$  celem wyodrębnienia rozmytych reguł. W tym celu wyznacza

się požądane wartości wyjściowe, na podstawie odpowiednich podzbiorów danych uczących, dla  $i$ -tej reguły (wartości minimalną, średnią i maksymalną). Dodatkowo dla każdego podzbioru jest wyznaczany maksymalny stopień aktywacji reguły oraz podzbiór danych, które aktywują regułę w określonym lub większym stopniu, stanowiącym parametr metody.

Na podstawie wyliczonych powyżej wartości generowane jest dla każdej reguły z osobna pokolenie startowe, które podlega strategii ewolucyjnej. Algorytm generowania pokolenia startowego został szczegółowo przedstawiony w pracy [111]. Jako warunek zatrzymania strategii ewolucyjnej przyjęto osiągnięcie przez licznik pokoleń określonej wartości. Jako operator genetyczny zastosowano klasyczną dla strategii ewolucyjnej mutację z samoadaptacją. Nowe pokolenie powstawało z połączenia pokolenia dotychczasowego i zmutowanego z wykorzystaniem selekcji elitarniej. Jako funkcję przystosowania wykorzystano wskaźnik jakości systemu rozmytego oparty na ważonym błędzie średniokwadratowym. Algorytm zgrubnego wyodrębniania rozmytych reguł TSK oraz parametry sterujące jego przebiegiem zostały szczegółowo opisane w pracy [111].

W wyniku wykonania strategii ewolucyjnej dla każdej reguły, otrzymujemy zgrubne oszacowanie zbioru reguł. W kolejnym etapie zbiór ten jest poddawany lokalnemu strojeniu parametrów.

## Etap 2

W drugim etapie wykorzystano algorytm genetyczny o zmiennoprzecinkowym kodowaniu osobników. Każdy chromosom ma zakodowaną całą bazę wiedzy systemu rozmytego (zarówno parametry funkcji przynależności przesłanek jak i parametry konkluzji). Zatem każdy chromosom ma następującą postać:

$$\tilde{x}_m^k = [\underline{C}_m^{k,(1)T}, \underline{C}_m^{k,(2)T}, \dots, \underline{C}_m^{k,(I)T}]^T, \quad (4.10)$$

gdzie  $I$ -liczba reguł uzyskanych w procesie zgrubnego szacowania parametrów reguł,  $m$ -numer osobnika,  $k$ -numer pokolenia

$$\underline{C}_m^{k,(i)} = [a_{m,1}^{k,(i)}, b_{m,1}^{k,(i)}, c_{m,1}^{k,(i)}, a_{m,2}^{k,(i)}, b_{m,2}^{k,(i)}, c_{m,2}^{k,(i)}, \dots, a_{m,N}^{k,(i)}, b_{m,N}^{k,(i)}, c_{m,N}^{k,(i)}, \tilde{p}_{m,0}^{k,(i)}, \tilde{p}_{m,1}^{k,(i)}, \dots, \tilde{p}_{m,N}^{k,(i)}]^T, \quad (4.11)$$

gdzie  $a_n^{m,k,(i)}, b_n^{m,k,(i)}, c_n^{m,k,(i)}$  - parametry trójkątnej funkcji przynależności związanej z  $n$ -tym wejściem systemu,  $\tilde{p}_0^{m,k,(i)}, \tilde{p}_1^{m,k,(i)}, \dots, \tilde{p}_N^{m,k,(i)}$  - zakodowane kątowno parametry konkluzji,  $i$ -numer reguły. Generowanie pokolenia startowego jest podobne do przedstawionego w trzecim etapie trójetapowej metody uczenia systemu Mamdaniego-Assiliania. Jeden z osobników jest generowany z wykorzystaniem danych uzyskanych w poprzednim etapie uczenia, parametry pozostałych osobników są generowane losowo, przy czym wartości parametrów opisujących trójkątne funkcje przynależności przesłanek muszą należeć do odpowiednich przedziałów zmienności natomiast połowa osobników otrzymuje parametry konkluzji takiej, jak osobnik pierwszy, a parametry konkluzji osobników pozostałych są generowane wokół wartości pochodzących z pierwszego etapu z rozkładem gaussowskim o niewielkim odchyleniu standardowym. Algorytm generowania pokolenia startowego został szczegółowo opisany w pracy [111]. W strategii zastosowano krzyżowanie max-min-arytmetyczne [80] oraz mutację nierównomierną [119]. Po wygenerowaniu każdego pokolenia stosowano strategię ewolucyjną

(1+1) dla pewnej losowo wybranej części osobników, z operatorem mutacji jako głównym operatorem genetycznym. Jako funkcję oceniającą przyjęto wskaźnik jakościowy wyliczany na podstawie błędu średniokwadratowego, liczonego dla danych uczących. Funkcja ta znalazła zastosowanie zarówno w algorytmie genetycznym etapu drugiego, jak i strategii ewolucyjnej (1+1), która może być traktowana w nadrzędnym algorytmie genetycznym jako operator genetyczny [111]. Warunkiem zatrzymania strategii było osiągnięcie określonej liczby pokoleń. W strategii zastosowano stochastyczne próbkowanie oraz selekcję elitarną. Szczegółowy opis nadrzędnego algorytmu genetycznego oraz strategii ewolucyjnej, jak również wartości parametrów sterujących przebiegiem obliczeń ewolucyjnych można znaleźć w pracy [111].

## 4.6 Podsumowanie

Przedstawione powyżej przykłady wskazują zakres zastosowań metod ewolucyjnych do generowania systemów rozmytych. W przeciwieństwie do sieci neuronowych oraz algorytmów grupowania, zastosowanie algorytmów genetycznych i strategii ewolucyjnych pozwala na naturalne zakodowanie wiedzy zawartej w zdefiniowanych terminach lingwistycznych, a wynik ich działania (system rozmyty) ułatwia zrozumienie odnalezionych automatycznie zależności, pozwalając na intuicyjną interpretację pojedynczych reguł. Podsumowując, metody ewolucyjne bywają używane do:

- Wyznaczenia funkcji przynależności terminów lingwistycznych, przy znanej bazy reguł.
- Wyznaczenia bazy reguł, przy założonym, znanym podziale przestrzeni argumentów w postaci terminów lingwistycznych.
- Naprzemiennego poszukiwania bazy reguł i funkcji przynależności dla przestrzeni argumentów.
- Jednoczesnego poszukiwania całej bazy wiedzy (zarówno reguł jak i funkcji przynależności powiązanych z poszczególnymi terminami lingwistycznymi).
- Jednoczesnego przeszukiwania całej przestrzeni argumentów związanych zarówno z bazą wiedzy jak i z zastosowanym mechanizmem wnioskowania.

Ostatnie opisywane podejście jest najbardziej dojrzałe, ale ze względu na znaczną wymiarowość przestrzeni poszukiwań, bardzo wymagające obliczeniowo. Częstokroć też wystarcza zastosowanie jednego z mniej skomplikowanych podejść, przy jednoczesnym wykorzystaniu wiedzy heurystycznej (np. dotyczącej podziału przestrzeni argumentów).

W następnym rozdziale niniejszej rozprawy wprowadzono trójetapową technikę uczenia systemu rozmytego z parametryzowanymi konkluzjami. Do odnalezienia bazy wiedzy oraz dostrajania zastosowano strategię ewolucyjną, natomiast do redukcji bazy wiedzy zmodyfikowaną wersję algorytmu genetycznego. Do wyznaczania parametrów konkluzji systemu rozmytego zastosowano metodę najmniejszych kwadratów.

## 5

# Zastosowanie metod ewolucyjnych do projektowania systemów rozmytych z parametryzowanymi konkluzjami

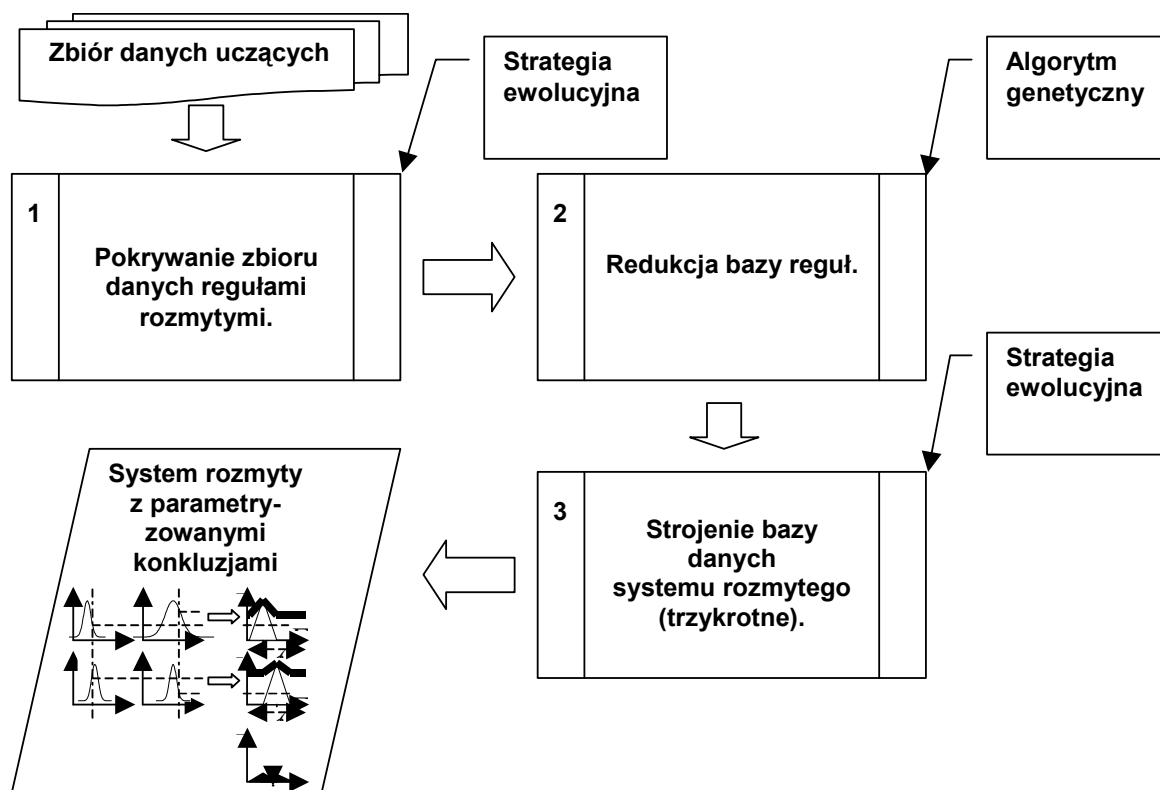
W niniejszej rozprawie, do automatycznego uzyskania bazy wiedzy systemu rozmytego z parametryzowanymi konkluzjami wykorzystano trójetapową metodę uczenia. Przyjęto następujące założenia dotyczące systemu rozmytego:

- Baza reguł systemu składa się z reguł rozmytych **if-then** z parametryzowaną konkluzją, o postaci przedstawionej w podrozdziale 2.3.1.
- Funkcje przynależności w przesłankach reguł mają postać funkcji gaussowskiej, natomiast w konkluzjach reguł znajdują się funkcje o postaci trójkąta równoramiennego.
- Parametry konkluzji stanowią kombinację liniową wartości wejściowych i nie podlegają procesowi ewolucyjnemu. Są one wyliczane z zastosowaniem metody najmniejszych kwadratów, metodą uczenia lokalnego lub globalnego.
- Modelowany system posiada zmienną bazę danych, obejmującą parametry gaussowskich funkcji przynależności przesłanek oraz parametry funkcji przynależności w konkluzjach reguł. Zmienna baza danych oznacza również, że operacje genetyczne są wykonywane bezpośrednio na ww. parametrach funkcji przynależności. Do bazy danych nie zaliczają się parametry kombinacji liniowej, opisującej położenie środka ciężkości konkluzji.
- Jako *t-normę* zastosowano operator minimum i iloczyn algebraiczny - badania przeprowadzono dla obu operatorów.
- Zastosowano logiczną interpretację reguł, z wykorzystaniem implikacji Łukasiewicza, Fodora, Reichenbacha, Kleene-Dienes, Zadeha, Goguena, Godela lub Reschera - badania przeprowadzono dla wszystkich ww. implikacji.

W kolejnych podrozdziałach rozprawy przedstawiono trzy etapy metody automatycznego uzyskania systemu rozmytego z parametryzowanymi konkluzjami na podstawie uczących danych numerycznych:

1. Pokrycie zbioru danych rozmytymi regułami **if-then** celem uzyskania bazy wiedzy systemu rozmytego (bazy reguł i bazy danych systemu jednocześnie). W tym celu wykorzystano strategię ewolucyjną ( $\mu+\lambda$ ) wraz z algorytmem pokrywania danych ze zbioru uczącego. W strategii przyjęto kodowanie zmiennoprzecinkowe zgodne z modelem Michigian. Szczegółowy opis znajduje się w podrozdziale 5.1.
2. Redukcja uzyskanej w pierwszym etapie bazy reguł, z wykorzystaniem zmodyfikowanego algorytmu genetycznego o klasycznym kodowaniu binarnym. Na tym etapie, optymalizacji podlega jedynie skład reguł w bazie reguł systemu rozmytego, natomiast baza danych systemu nie jest modyfikowana. Jeżeli baza reguł uzyskana podczas operacji pokrywania zawiera tylko kilka reguł, stosowano wyszukiwanie wyczerpujące zamiast algorytmu genetycznego, ze względu na większą szybkość. Szczegółowy opis znajduje się w podrozdziale 5.2.
3. Dokładne, lokalne strojenie bazy danych systemu rozmytego, zawierającego reguły pozostałe po operacji redukcji. W tym celu zastosowano strategię ewolucyjną ( $\mu+\lambda$ ), w której osobnik koduje całą bazę danych, zgodnie z modelem Pittsburgh. Modyfikacji podlegają wyłącznie parametry funkcji przynależności przesłanek i szerokość nośnika konkluzji (baza danych systemu rozmytego). Szczegółowy opis znajduje się w podrozdziale 5.3.

Schematycznie proces ten przedstawiono na poniższym rysunku:



Rysunek 5.1: Trójetapowa metoda uczenia systemu rozmytego.



## 5.1 Pokrywanie danych rozmytymi regułami if-then

Przegląd metod automatycznego wyodrębniania bazy wiedzy na podstawie danych numerycznych dla systemu rozmytego został przedstawiony w rozdziale 4. Zastosowanie strategii ewolucyjnej pozwala na odnalezienie reguły bliskiej lub równej optimum globalnemu w sensie przyjętego kryterium jakości (funkcji przystosowania, opisaney w dalszej części niniejszego podrozdziału). Dopuszczalne jest otrzymanie wyniku suboptymalnego ze względu na fakt iż wyodrębniona w ten sposób baza wiedzy podlega dalszym optymalizacjom, w szczególności redukcji i dostrajaniu. W rozdziale 4 opisano 3 podejścia do zagadnienia wyodrębniania bazy wiedzy systemu. Podejście Pittsburgh, jakkolwiek najbardziej dojrzałe z punktu widzenia obliczeń ewolucyjnych, jest kłopotliwe w zastosowaniu, ze względu na bardzo duże wymagania obliczeniowe związane z koniecznością kodowania całości informacji o systemie rozmytym w pojedynczym osobniku. Dodatkowo wymaga stosowania specjalizowanych operatorów ewolucyjnych, działających na chromosomach o różnej długości, które mogą powodować kłopoty ze zbieżnością algorytmu genetycznego lub strategii ewolucyjnej. Nie bez znaczenia jest również znaczna komplikacja implementacji związana ze stosowaniem chromosomów o różnej długości (różnej liczbie reguł w bazie wiedzy). W podejściu Michigan występuje problem konkurowania ze sobą najlepszych reguł opisujących określony zbiór danych uczących w jednej populacji, co może mieć negatywny wpływ na jakość systemu. Dlatego też w niniejszej rozprawie zastosowano iteracyjny algorytm pokrywania regułami zbioru uczącego. W każdym „przejściu” głównej pętli algorytmu pokrywającego wykorzystywana jest strategia ewolucyjna do odnalezienia najlepszej reguły.

W każdym kroku głównej pętli algorytmu pokrywającego ustalany jest nowy zbiór uczący, składający się z danych, których dotychczasowy stopień pokrycia przez wygenerowane uprzednio reguły nie przekroczył określonego minimalnego poziomu, stanowiącego parametr metody. Po odnalezieniu reguły jest ona włączana do bazy wiedzy, a pokryte przez nią dane ze zbioru uczącego są usuwane, jeśli wartość pokrycia przekroczy założony, minimalny próg. Proces trwa, dopóki w zbiorze uczącym pozostaje liczba rekordów większa lub równa minimalnej liczbie niezbędnej do wygenerowania pokolenia startowego. W kolejnych podrozdziałach przedstawiono szczegółowo algorytm pokrywania oraz zastosowaną strategię ewolucyjną odnajdującą rozmyte reguły **if-then** systemu z parametryzowanymi konkluzjami, dla zadanego zbioru uczącego. W opisie przyjęto zasadę zstępującą, „od ogółu do szczegółu” (ang. *top-down*).

### 5.1.1 Algorytm pokrywania zbioru uczącego

Algorytm pokrywania jest typowym procesem iteracyjnym, który kończy swoje działanie, gdy wszystkie  $N_T$  elementów w zbiorze uczącym  $E_{NT}$  zostanie pokryte przez odnalezione reguły w stopniu większym lub równym  $\varepsilon$ . Ponieważ zbiory uczące w przeprowadzanych eksperymentach składały się z minimum 100 rekordów, jako minimalną liczbę niepokrytych rekordów danych przyjęto wartość 5, co stanowi maksimum 5% liczby rekordów zbioru uczącego. Jest to wartość niezbędna do zainicjalizowania pokolenia startowego dla strategii ewolucyjnej.

```

 $C_V[i] \leftarrow 0$ ; dla  $i=1, \dots, N_T$ 
 $\mathcal{R} \leftarrow \emptyset$ ;
 $E_C \leftarrow E_{NT}$ 
while ( $\dim(C_V[j]) < \varepsilon$ )  $\geq 5$  ; dla  $j=1, \dots, \dim(C_V)$ 
     $R \leftarrow \text{znajdź\_regulę}(E_C)$ ;
     $C_V[j] \leftarrow C_V[j] + \text{pokrycie}(R, E_C[j])$  ; dla  $j=1, \dots, \dim(C_V)$ 
     $\mathcal{R} \leftarrow \mathcal{R} \cup R$  ;
    usuń z  $E_C$  rekordy, dla których  $C_V[j] \geq \varepsilon$  ; dla  $j=1, \dots, \dim(C_V)$ 
end

```

gdzie  $C_V$  - wartość pokrycia rekordów zawartych w  $E_C$  dla bieżącej bazy reguł  $\mathcal{R}$ ,  $\mathcal{R}$  - baza wiedzy systemu rozmytego w postaci zbioru rozmytych reguł warunkowych **if-then**,  $R$  - rozmyta reguła warunkowa **if-then** z parametryzowaną konkluzją,  $\varepsilon$  - minimalny poziom pokrycia rekordu ze zbioru uczącego przez reguły w bazie reguł  $\mathcal{R}$ , powodujący usunięcie rekordu danych ze zbioru uczącego. Wzrost tej wartości powoduje zwiększenie wymagań co do stopnia pokrycia rekordów ze zbioru uczącego, co przy przyjętych ograniczeniach pętli iteracyjnej daje możliwość niedeterministycznego sterowania liczbą generowanych reguł. Zapis  $\mathcal{R} \leftarrow \mathcal{R} \cup R$  oznacza dołączenie do bazy wiedzy  $\mathcal{R}$  reguły rozmytej  $R$ .

Funkcja  $\text{znajdź\_regulę}(E_C)$  wyszukuje z wykorzystaniem strategii ewolucyjnej najlepszą regułę rozmytą, opisującą dane zawarte w podzbiorze  $E_C$ . Jej szczegółowy opis znajduje się w podrozdziale 5.1.2. Funkcja  $\text{pokrycie}(R, E_C[j])$  wylicza cząstkową wartość pokrycia  $C_i(E_C[j])$  dla  $i$ -tej reguły  $R^{(i)}$  oraz  $j$ -tej pary wejście-wyjście ze zbioru  $E_C$ , co formalnie można zapisać następująco:

$$C_i(E_C[j]) = R_c^{(i)}(E_C[j]), \quad (5.1)$$

gdzie  $R_c^{(i)}(E_C[j])$  – stopień kompatybilności  $i$ -tej reguły  $R$  z  $j$ -tym rekordem danych z podzbioru  $E_C$  zbioru uczącego  $E_{NT}$ . Rekord ten zawiera informację o  $j$ -tym wektorze danych wejściowych  $\underline{x}_0(j)$  oraz  $j$ -tej oczekiwanej wartości wyjściowej  $t_0(j)$ . Uzyskana wartość  $C_i(E_C[j])$  jest następnie dodawana w odpowiadającej jej  $j$ -tej pozycji wartości pokrycia reguły:

$$C_v[j] = C_v[j] + C_i(E_C[j]), \quad (5.2)$$

natomiast stopień kompatybilności  $R_c^{(i)}(E_C[j])$ , gdzie  $E_C[j] = \{\underline{x}_0(j), t_0(j)\}$ , jest wyliczany następująco:

$$R_c^{(i)}(\underline{x}_0(j), t_0(j)) = \mu_{\underline{A}^{(i)}}(\underline{x}_0(j)) \star_{T_c} \mu_{B^{(i)}}(t_0(j)), \quad (5.3)$$

gdzie  $i$  – indeks reguły w bazie reguł,  $\star_{T_c}$  -  $t$ -norma stosowana w przypadku koniunkcyjnej interpretacji reguł. W niniejszej rozprawie przyjęto że jest ona identyczna z  $t$ -normą stosowaną do wyznaczenia poziomu aktywacji reguły i stanowi parametr metody (operator minimum lub iloczyn algebraiczny).

Ostatecznie, wartość pokrycia  $C_V$  przyjmuje w procesie iteracyjnym wartości, które można obliczyć w następujący sposób:

$$C_V(E_C[j]) = \sum_{\{i|R^{(i)} \in \mathcal{R}\}} R_c^{(i)}(E_C[j]). \quad (5.4)$$

Wyliczanie wartości  $\mu_{B^{(i)}}$  podczas obliczania kompatybilności reguły i danych uczących wymaga znajomości parametrów konkluzji. Ponieważ oceniana reguła  $R$  jest potencjalnym kandydatem należącym do ostatecznej wersji systemu rozmytego (jeżeli nie zostanie zredukowana w trakcie upraszczania bazy reguł, w drugim etapie trójetapowej metody uczenia), do wyliczenia parametrów zastosowano uczenie lokalne, ze zbiorem uczącym  $E_{NT}$  zamiast  $E_C$ .

### 5.1.2 Algorytm odnajdywania najlepszej reguły

W celu odnalezienia reguły pokrywającej zadany podzbiór  $E_C$  zbioru danych uczących  $E_{NT}$ , wykorzystano strategię ewolucyjną  $(\mu + \lambda)$ . Zastosowano kodowanie zmiennoprzecinkowe, wykorzystano model elitarny i stochastyczne próbkowanie populacji bazowej celem wyboru rodziców do wygenerowania pokolenia potomnego. Nowe pokolenie generowano za pomocą operatora periodycznej mutacji nierównomiernej oraz krzyżowania max-min-arytmetycznego. Kolejne pokolenie bazowe powstawało z połączenia pokolenia bazowego i potomnego, z wykorzystaniem selekcji elitarniej. Poniżej przedstawiono algorytm strategii ewolucyjnej odnajdywania najlepszej reguły pokrywającej zadany zbiór rekordów uczących:

```

wylicz  $\underline{x}_0 \leftarrow, \bar{x}_0 \leftarrow$  oraz  $t_0 \leftarrow, \bar{t}_0 \leftarrow$ ;
 $Q^{(0)} \leftarrow$  inicjalizuj( $\mu, N_T$ );
 $k = 0$ ;
while  $k < K$ 
  if  $k == 0$ 
     $Q^{(k)} \leftarrow$  uczenie_lokalne( $Q^{(k)}, E_{NT}$ );
     $\underline{f} \leftarrow$  oceń( $Q^{(k)}, E_C$ );
  end
   $\mathcal{J}^{(k)} \leftarrow$  próbkowanie_stochastyczne( $Q^{(k)}, \lambda$ ); wybór  $\lambda$  osobników
   $\mathcal{J}_C^{(k)} \leftarrow$  krzyżowanie( $\mathcal{J}^{(k)}$ );
   $O^{(k)} \leftarrow$  mutacja( $\mathcal{J}_C^{(k)}$ );
   $O^{(k)} \leftarrow$  uczenie_lokalne( $O^{(k)}, E_{NT}$ );
   $\underline{f}_O \leftarrow$  oceń( $O^{(k)}, E_C$ );
   $Q^{(k+1)} \leftarrow$  selekcja_elitarna  $\mu$  osobników z  $Q^{(k)} \cup O^{(k)}$  na podstawie  $\underline{f}$  i  $\underline{f}_O$ ;
   $\underline{f} \leftarrow$  przepisanie ocen wybranych osobników z  $\underline{f}$  i  $\underline{f}_O$  zgodnie z  $Q^{(k+1)}$ ;
   $k = k + 1$ ;
end

```

gdzie  $k$  - licznik pokoleń,  $K$  - maksymalna liczba pokoleń, równoważna warunkowi zatrzymania strategii ewolucyjnej,  $Q^{(k)}$  -  $k$ -te pokolenie bazowe,  $\mathcal{J}^{(k)}$  -  $k$ -te pokolenie rodziców,  $\mathcal{J}_C^{(k)}$  -  $k$ -te

pokolenie tymczasowe z osobnikami uzyskanymi po operacji krzyżowania,  $O^{(k)}$  –  $k$ -te pokolenie potomne. Funkcja inicjalizuj( $\mu, N_T$ ) generuje pokolenie startowe. Funkcja uczenie\_lokalne( $Q^{(k)}, E_{NT}$ ) ustala parametry konkluzji zadanego zbioru (pokolenia) reguł, korzystając z całego zbioru uczącego  $E_{NT}$ . Funkcja próbkowanie\_stochastyczne( $Q^{(k)}, \lambda$ ) wybiera  $\lambda$  osobników rodzicielskich z pokolenia bazowego, metodą próbkowania stochastycznego. Funkcja krzyżowanie( $\mathcal{J}^{(k)}$ ) wykonuje operację max-min-arytmetycznego krzyżowania, dając w wyniku pokolenie pośrednie  $\mathcal{J}_C^{(k)}$ , które następnie jest poddawane periodycznej mutacji nierównomiernej za pomocą funkcji mutacja( $\mathcal{J}_C^{(k)}$ ). Ostatecznie pokolenie potomne  $O^{(k)}$  zawiera  $\lambda$  osobników. Pokolenie to następnie jest oceniane (po wcześniejszym lokalnym douczeniu parametrów konkluzji). Ostatecznie kolejne pokolenie bazowe uzyskiwane jest z połączenia dotychczasowego pokolenia bazowego i pokolenia potomnego za pomocą selekcji elitarniej  $\mu$  najlepszych osobników. Funkcja oceń( $\bullet, \bullet$ ) ocenia wskazane pokolenie w relacji z zadanym zbiorem danych uczących. Zakłada się przy tym, że parametry konkluzji reguł są wyliczone wcześniej. Oceny osobników są przechowywane w wektorach  $\underline{f}$  i  $\underline{f}_O$  (odpowiednio wektor ocen pokolenia bazowego i pokolenia potomnego):

$$\underline{f} = [f_1, \dots, f_\mu], \quad (5.5)$$

$$\underline{f}_O = [f_1, \dots, f_\lambda], \quad (5.6)$$

gdzie  $f_i \in \mathfrak{R}$ ,  $f_i \geq 0$ . Wartości  $\underline{x}_0^{\leftarrow}, t_0^{\leftarrow}$ ,  $\underline{x}_0^{\rightarrow}, t_0^{\rightarrow}$  stanowią odpowiednio lewy i prawy kres przedziału zmienności argumentów oraz wyjścia i są wyznaczane w następujący sposób:

$$\begin{aligned} [\underline{x}_0^{\leftarrow}, \underline{x}_0^{\rightarrow}] &= [\underline{x}_0^- - 0.1 \cdot (\underline{x}_0^+ - \underline{x}_0^-), \underline{x}_0^+ + 0.1 \cdot (\underline{x}_0^+ - \underline{x}_0^-)], \\ [t_0^{\leftarrow}, t_0^{\rightarrow}] &= [t_0^- - 0.1 \cdot (t_0^+ - t_0^-), t_0^+ + 0.1 \cdot (t_0^+ - t_0^-)], \end{aligned} \quad (5.7)$$

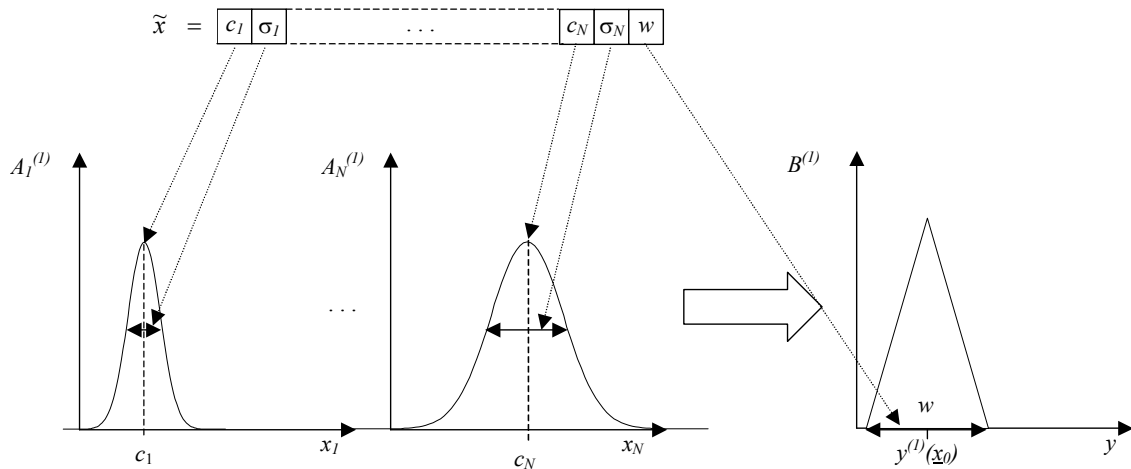
gdzie  $\underline{x}_0^- = [x_{0,1}^-, \dots, x_{0,N}^-]^T$  oraz  $\underline{x}_0^+ = [x_{0,1}^+, \dots, x_{0,N}^+]^T$ , a wartości  $x_{0,i}^-$ ,  $x_{0,i}^+$  dla  $i=1, \dots, N$  oraz  $t_0^-$ ,  $t_0^+$  są wyznaczane następująco:

$$\begin{aligned} x_{0,i}^- &= \min_{1 \leq n_T \leq N_T} (x_{0,i}(n_T)), \\ x_{0,i}^+ &= \max_{1 \leq n_T \leq N_T} (x_{0,i}(n_T)), \\ t_0^- &= \min_{1 \leq n_T \leq N_T} (t_0(n_T)), \\ t_0^+ &= \max_{1 \leq n_T \leq N_T} (t_0(n_T)). \end{aligned} \quad (5.8)$$

### Kodowanie osobników

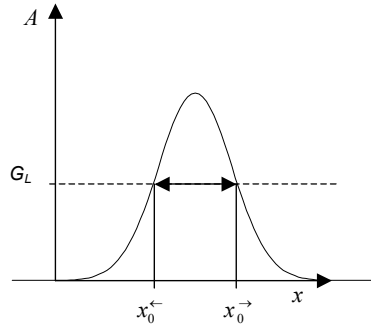
W rozważanym modelu, przyjęto kodowanie osobników o stałej długości, identycznie jak to ma miejsce w przypadku modelu Michigan dla systemów o nieustalonej bazie danych. W związku z założeniami poczynionymi wobec postaci funkcji przynależności zbiorów rozmytych w przesłankach (gaussowskie) i konkluzji (trójkątne) reguł, chromosom każdego

osobnika prezentuje parametry funkcji przynależności przesłanek i nośnika w konkluzji pojedynczej reguły rozmytej **if-then** z parametryzowaną konkluzją i składa się z  $\dim(\tilde{x})=2 \cdot N+1$  liczb rzeczywistych. Parametry opisujące położenie konkluzji stanowią *część niejawną* kodowania i nie są zawarte w chromosomie, bowiem nie podlegają procesowi ewolucyjnemu. Każdy zbiór rozmyty w przesłance jest charakteryzowany za pomocą dwóch wartości, gdzie pierwsza z nich określa położenie środka funkcji gaussowskiej, a druga określa jej dyspersję. Funkcja przynależności zbioru rozmytego konkluzji ma postać trójkąta równoramiennego, w związku z tym jest opisywana wyłącznie przez długość podstawy, tożsamą z nośnikiem zbioru. Położenie środka ciężkości w konkluzji, padające na oś poziomą w połowie długości podstawy jest każdorazowo wyliczane jako kombinacja liniowa argumentów wejściowych, w związku z tym nie jest konieczne jego przechowywanie w chromosomie osobnika. Położenie to również należy do *części niejawnej* kodowania reguły rozmytej z parametryzowaną konkluzją. Locus  $l$  w chromosomie jednoznacznie identyfikuje związany ze wskazywanym parametrem numer wejścia lub wyjścia. Jeżeli  $l=2 \cdot N+1$ , parametr należy do opisu konkluzji reguły, w przeciwnym wypadku jest to parametr opisujący jedną z funkcji przynależności zbiorów rozmytych przesłanki. Każdy zbiór rozmyty przesłanki reguły jest opisywany za pomocą dwóch cech. Numer wejścia  $n$  w przesłance może być ustalony jako  $n=\lceil l/2 \rceil$ , natomiast gdy  $(l \bmod 2)=1$ , pozycja opisuje środek krzywej gaussowskiej, w przeciwnym wypadku jej dyspersję. Poniższy rysunek graficznie prezentuje kodowanie jednej reguły, opowiadającej jednemu osobnikowi, zastosowanej w pierwszym etapie trójetapowej metody uczenia:



Rysunek 5.2: Kodowanie reguły w pierwszym etapie trójetapowej metody uczenia.

Wartości początkowe  $c_i$  oraz związane z nimi wartości  $\sigma_i$  dla  $i=1,2,\dots,N$  dla osobników pokolenia startowego  $Q^{(0)}$  muszą tworzyć w funkcjach przynależności przesłanek krzywe gaussowskie, których przynajmniej jedna połowa (mierzona od centrum krzywej w kierunku wzrastających lub malejących wartości argumentów) znajduje się w całości w przedziale zmienności odpowiadających im wartości wejściowych  $[\underline{x}_0^{\leftarrow}, \underline{x}_0^{\rightarrow}]$  i przyjmuje wartości większe od ustalonego poziomu  $G_L$ . Na poniższym rysunku przedstawiono sposób wyliczenia przedziału przy zadanej wartości  $G_L$ :



Rysunek 5.3: Ustalenie przedziału zmienności dla danego poziomu  $G_L$ .

Początkowe wartości opisujące parametry trójkąta w konkluzji spełniają warunek, zgodnie z którym całość nośnika zbioru rozmytego konkluzji przynależy do zakresu zmienności wartości wyjściowych  $[t_0^{\leftarrow}, t_0^{\rightarrow}]$ . Dodatkowo do wygenerowania pewnej liczby osobników startowych zastosowano algorytm grupowania metodą FCM [14], [48]. Do każdego pokolenia startowego dołączane są osobniki wygenerowane na podstawie podziału danych ze zbioru uczącego metodą FCM na zadaną liczbę grup (gdzie liczba ta stanowi parametr metody) oraz osobniki wygenerowane na podstawie podziału metodą FCM na 2 grupy. Jeśli liczba rekordów danych uczących jest zbyt mała do zastosowania algorytmu FCM, następuje redukcja liczby grup lub zaniechanie generowania osobników tą metodą – wtedy wszystkie osobniki są generowane na podstawie danych uczących oraz losowo.

Szczegółowy sposób inicjalizacji pokolenia startowego został przedstawiony poniżej.

### Inicjalizacja pokolenia startowego

Pokolenie startowe  $Q^{(0)}$  składa się z  $\mu = M_1 + M_{1,2} + M_2 + M_3$  osobników, z których:

$$\begin{aligned} M_1 &= \min(n_{FCM}, \dim(E_C) - t_2 + 1), \\ M_{1,2} &\in \{0, 2\}, \\ M_2 &= t - M_{FCM}, \\ M_3 &= \mu - t, \end{aligned} \quad (5.9)$$

gdzie  $M_1$  – liczba osobników generowanych na podstawie podziału przestrzeni wejściowej danych uczących metodą rozmytych c-średnich (FCM),  $M_2$  – liczba osobników generowanych na podstawie danych uczących,  $M_3$  – liczba osobników generowanych losowo,  $n_{FCM}$  – założona liczba grup, na jakie dzielone są dane wejściowe metodą FCM, tożsama z liczbą wygenerowanych tą metodą reguł, przy czym wartość ta może być zmniejszona w zależności od liczby rekordów uczących,  $M_{FCM}$  – łączna liczba osobników wygenerowanych z wykorzystaniem podziału algorytmem FCM – maksymalna wartość  $M_{FCM} = n_{FCM} + 2$  zaś minimalna może być równa zero,  $t$  – określa, ile osobników zostanie wygenerowanych metodami nielosowymi:

$$t = \min(\text{round}(0.75 \cdot \mu), E_C), \quad (5.10)$$

gdzie  $round(\bullet)$  – operator zaokrąglania do najbliższej wartości całkowitej,  $t_2$  – ogranicznik liczby generowanych osobników z wykorzystaniem metody FCM (dla podziału na założoną liczbę grup  $n_{FCM}$ ):

$$t_2 = round\left(\frac{t}{2}\right). \quad (5.11)$$

Osobniki wygenerowane na podstawie podziału danych uczących metodą FCM na dwie grupy są dołączane do pokolenia startowego jeżeli zachodzi:

$$M_1 + 2 \leq t_2. \quad (5.12)$$

Poniżej przedstawiono ogólny algorytm generowania pokolenia startowego:

```

t = min(round(0.75 · μ), dim(EC)) ;
t2 = round(0.5 · t) ;
EG ← losowy wybór t rekordów ze zbioru uczącego EC ;
Q(0) = ∅ ;
M1 = min(nFCM, t2) ;
generuj osobniki FCM (M1, t2) ;
MFCM = dim(Q(0)) ;
M2 = t - MFCM ;
generuj osobniki na podstawie danych uczących (M2) ;
M3 = μ - t ;
generuj osobniki losowo (M2) ;

```

gdzie  $E_G$  – podzbiór zbioru  $E_C$ , zawierający  $t$  - losowo wybranych rekordów uczących ze zbioru  $E_C$ , funkcje:

generuj osobniki FCM( $\bullet, \bullet$ ) ,

generuj osobniki na podstawie danych uczących( $\bullet$ ) ,

generuj osobniki losowo( $\bullet$ ) ,

generują osobniki pokolenia startowego, odpowiednio z wykorzystaniem algorytmu FCM, na podstawie danych uczących oraz losowo na ustalonych na podstawie rekordów w podzbiorze  $E_G$  zbioru uczącego  $E_C$  przedziałach zmienności.

Szczegółowe algorytmy przedstawiające sposób generowania tych osobników przedstawiono poniżej.

Algorytm generowania osobników na podstawie podziału metodą FCM:

```

[ $C_{FCM}$ ,  $U_{FCM}$ ] = fcm( $E_C$ ,  $M_1$ ) ;
if bFCMSukces then
  for  $i=1$  to  $M_1$ 
    for  $n=1$  to  $N$ 
       $\sigma_n = 5 \cdot \text{normfit}(E_C(U_{FCM,i,n}))$  ;
       $C_n = C_{FCM,n}$  ;
    end
     $w = 1.5 \cdot (\max(U_{FCM,i,n+1}) - \min(U_{FCM,i,n+1}))$  ;
     $Q^{(0)} \leftarrow Q^{(0)} \cup \text{osobnik}(C_1, \dots, C_N, \sigma_1, \dots, \sigma_N, w)$  ;
  end
  if  $M_1 + 2 \leq t_2$  then
    [ $C_{FCM2}$ ,  $U_{FCM2}$ ,  $f_{FCM2}$ ] = fcm( $E_C$ , 2) ;
    for  $i=1$  to 2
      for  $n=1$  to  $N$ 
         $\sigma_n = 5 \cdot \text{normfit}(E_C(U_{FCM2,i,n}))$  ;
         $C_n = C_{FCM2,n}$  ;
         $w = 1.5 \cdot (\max(E_C(U_{FCM2,i,n})) - \min(E_C(U_{FCM2,i,n})))$  ;
      end
       $Q^{(0)} \leftarrow Q^{(0)} \cup \text{osobnik}(C_1, \dots, C_N, \sigma_1, \dots, \sigma_N, w)$  ;
    end
  end
end

```

gdzie  $N$ -liczba wejść systemu rozmytego,  $n+1$  oznacza wartości związane z wyjściem systemu rozmytego, funkcja  $\text{fcm}(\bullet, \bullet)$  wykonuje operację grupowania na  $M_1$  grup, metodą FCM, na zadanym parametrem zbiorze danych  $E_C$ , funkcja  $\text{normfit}(\bullet)$  zwraca wartość dyspersji funkcji gaussowskiej dla zadanego zbioru danych, wartości 5 oraz 1.5 dotyczące odpowiednio współczynnika rozszerzenia dyspersji w gaussowskich funkcjach przynależności przesłanek oraz współczynnika rozszerzenia nośnika w trójkątnej funkcji przynależności konkluzji zostały dobrane eksperymentalnie i stanowią wyłącznie przybliżenia, które później są optymalizowane przez strategię ewolucyjną. Flaga **bFCMSukces** informuje, czy operacja podziału FCM została wykonana poprawnie. W algorytmie FCM zastosowano następujące parametry:

- Maksymalna liczba iteracji = 500.
- Minimalna wartość poprawy w kolejnych iteracjach =  $1 \cdot 10^{-9}$ .

Macierze  $C_{FCM}$  i  $C_{FCM2}$  przechowują współrzędne odnalezionych wartości centralnych, natomiast  $U_{FCM}$  i  $U_{FCM2}$  przechowują wartości odległości od poszczególnych centrów (które można interpretować jako stopień przynależności danej wejściowej o wskazanym indeksie do danej grupy powstałej dookoła wybranego centrum). Zapis  $C_{FCM,i,n}$ ,  $C_{FCM2,i,n}$  oznacza wartość centralną dla  $i$ -tej grupy i  $n$ -tego wymiaru (odpowiednio  $i$ -tej reguły i  $n$ -tego wejścia systemu rozmytego), natomiast zapis  $E_C(U_{FCM,i,n})$ ,  $E_C(U_{FCM2,i,n})$  należy interpretować jako pobranie wartości  $x_{0,n} \in E_C$  dla wszystkich rekordów, dla których macierz  $U$  wskazuje, iż przynależą do  $i$ -tej grupy. Funkcja  $\text{osobnik}(C_1, \dots, C_N, \sigma_1, \dots, \sigma_N, w)$  tworzy osobnika-regułę z funkcjami przynależności w przesłankach i konkluzji o podanych parametrach.



Algorytm generowania osobników na podstawie danych uczących:

```

for  $i=1$  to  $M_2$ 
  for  $n=1$  to  $N$ 
     $\Delta x = \max( E_G(x_{0,n}(i)) - x_{0,n}^{\leftarrow}, x_{0,n}^{\rightarrow} - E_G(x_{0,n}(i)) )$  ;
     $d_x = \delta[0.0001, 1] \cdot \Delta x$  ;
     $c_n = E_G(x_{0,n}(i))$  ;
     $\sigma_n = \sqrt{\frac{(d_x)^2}{-2 \cdot \ln(G_L)}}$  ;
  end
   $\Delta t_0 = \min( E_G(t_0(i)) - t_0^{\leftarrow}, t_0^{\rightarrow} - E_G(t_0(i)) )$  ;
   $w = 2 \cdot \delta[0.0001, 1] \cdot \Delta t_0$  ;
   $Q^{(0)} \leftarrow Q^{(0)} \cup \text{osobnik}(c_1, \dots, c_N, \sigma_1, \dots, \sigma_N, w)$  ;
end

```

gdzie  $E_G(\bullet)$  – operacja pobrania wskazanej danej ze zbioru  $E_G$ ,  $\delta[\bullet, \bullet]$  – realizacja zmiennej losowej o rozkładzie równomiernym na wskazanym, domkniętym przedziale,  $c_n$ ,  $\sigma_n$  – parametry gaussowskiej funkcji przynależności  $n$ -tej wartości wejściowej w przesłance. Funkcja  $\text{osobnik}(c_1, \dots, c_N, \sigma_1, \dots, \sigma_N, w)$  tworzy osobnika-regułę z funkcjami przynależności w przesłankach i konkluzji o podanych parametrach. Zapis  $Q^{(0)} \leftarrow Q^{(0)} \cup \bullet$  oznacza dodanie do  $Q^{(0)}$  wskazanego osobnika.

Generowanie pozostałych  $M_2$  osobników jest wykonywane w sposób losowy, przy czym wygenerowane osobniki muszą mieć parametry funkcji przynależności przesłanek i konkluzji należące do następujących przedziałów:

$$c_{j,i} \in [x_{0i}^{\leftarrow}, x_{0i}^{\rightarrow}], \quad (5.13)$$

$$\sigma_{j,i} \in \left[ 0.0001 \cdot (x_{0i}^{\leftarrow} - x_{0i}^{\rightarrow}), \sqrt{\frac{(x_{0i}^{\leftarrow} - x_{0i}^{\rightarrow})^2}{-2 \cdot \ln(G_L)}} \right], \quad (5.14)$$

$$w_j \in [0.0001 \cdot (x_{0i}^{\leftarrow} - x_{0i}^{\rightarrow}), (x_{0i}^{\leftarrow} - x_{0i}^{\rightarrow})], \quad (5.15)$$

gdzie  $i=1, 2, \dots, N, j=1, 2, \dots, M_2$ . Powyższe wartości są generowane z rozkładem równomiernym na odpowiadających im przedziałach.

Wymóg dotyczący przynależności parametrów opisujących funkcje przynależności (w szczególności gaussowskie funkcje przynależności w przesłankach) podyktowany jest koniecznością ograniczenia przestrzeni poszukiwań strategii ewolucyjnej oraz zwiększeniem szans na odnalezienie „dobrego” osobnika w sensie przyjętego kryterium jakości, opisanego w dalszej części niniejszego podrozdziału.

### Próbkowanie stochastyczne

Wybór do operacji genetycznych  $\lambda$  rodziców spośród  $\mu$  osobników pokolenia  $Q^{(k)}$  następuje z wykorzystaniem próbkowania stochastycznego.

Poniżej przedstawiono algorytm próbkowania:

```

 $s = \sum_{i=1}^{\mu} \underline{f}(i);$ 
if  $s == 0$  then
     $v_s = \text{ones}(\mu) / \mu;$ 
else
    for  $j = 1$  to  $\mu$ 
        
$$v_s(j) = \frac{\sum_{i=1}^j \underline{f}(i)}{s};$$

    end
end
 $m = \delta[0, 1] \cdot (1/\lambda);$ 
for  $i = 1$  to  $\lambda$ 
     $j = 0;$ 
    while  $(v_s(j) \leq m) \ \& \ (j < \mu)$ 
         $j = j + 1;$ 
    end
     $\mathcal{J}^{(k)}[i] = Q^{(k)}[j];$ 
     $m = m + 1/\lambda;$ 
end

```

gdzie  $\text{ones}(\bullet)$  generuje wektor jednostkowy o zadanym wymiarze i w niniejszym algorytmie spełnia rolę zabezpieczenia na wypadek, gdyby wszystkie elementy pokolenia bazowego miały ocenę równą 0,  $m$  – punkt próbkowania stochastycznego,  $\delta[\bullet, \bullet]$  – realizacja zmiennej losowej o rozkładzie równomiernym, na podanym, zamkniętym przedziale,  $k$  – numer bieżącego pokolenia, wektor  $v_s$  przechowuje znormalizowaną sumę narastającą wartości funkcji przystosowania osobników z pokolenia  $Q^{(k)}$ ,  $\underline{f}$  – wektor wartości przystosowania osobników. Indeksy wartości ocen w wektorze  $\underline{f}$  odpowiadają indeksom osobników w pokoleniu bazowym  $Q^{(k)}$ . Jeśli wszystkie osobniki w pokoleniu mają zerową ocenę przystosowania, osobniki tworzące pokolenie rodzicielskie są wybierane z  $Q^{(k)}$  z równym prawdopodobieństwem.

### Operatory

W strategii ewolucyjnej zastosowano dwa operatory, realizowane w osobnych funkcjach operujących na całych pokoleniach. W pierwszej kolejności pokolenie  $\mathcal{J}^{(k)}$  poddawane jest operacji krzyżowania max-min-arytmetycznego, następnie osobniki powstałego w ten sposób

pokolenia pośredniego  $\mathcal{J}_c^{(k)}$  podlegają periodycznej mutacji nierównomiernej. Liczebność pokoleń  $\mathcal{J}^{(k)}$  oraz  $\mathcal{J}_c^{(k)}$  jest stała i wynosi  $\lambda$ .

Poniżej przedstawiono algorytm wykonania operacji krzyżowania:

```

k = 0;
while k/2 < ⌊λ/2⌋
  iP1 = ⌈δ[0, λ]⌉;
  iP2 = ⌈δ[0, λ]⌉;
  while iP1 == iP2
    iP2 = ⌈δ[0, λ]⌉;
  end
  jP1 =  $\mathcal{J}^{(k)}$ [iP1];
  jP2 =  $\mathcal{J}^{(k)}$ [iP2];
   $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \text{krzyżowanie}(j_{P1}, j_{P2})$ ;
  k = k + 1;
end
if 2·k < λ then
   $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \text{trzeci w kolejności wg funkcji przystosowania osobnik z}$ 
   $\text{ostatniego procesu krzyżowania};$ 
end

```

gdzie  $k$  – numer bieżącego pokolenia,  $i_{P1}, i_{P2}$  – indeksy osobników,  $j_{P1}, j_{P2}$  – osobniki wybrane do krzyżowania, zapis  $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \bullet$  oznacza dodanie do  $\mathcal{J}_c^{(k)}$  wskazanych osobników.

Jak widać z powyższego algorytmu, krzyżowanie przebiega tak długo, aż zostanie wygenerowane nowego pokolenie o liczebności  $\lambda$ , w którym każdy osobnik jest osobnikiem nowym (choć niekoniecznie unikalnym). Takie postępowanie jest podyktowane faktem zastosowania selekcji elitarniej w procesie wyboru kolejnego pokolenia bazowego na podstawie bieżącego pokolenia bazowego i bieżącego pokolenia potomnego, która gwarantuje przetrwanie najlepszego osobnika z prawdopodobieństwem równym 1 i jest odmienne od typowego algorytmu genetycznego, w którym operacja krzyżowania jest wykonywana z zadanyim prawdopodobieństwem (para osobników, które nie podlegały krzyżowaniu przechodzi przez procedurę krzyżującą niezmienną). Szczegółowy opis operacji krzyżowania znajduje się poniżej:

```

Cp1 ←  $\mathcal{J}^{(k)}$ (ip1);
Cp2 ←  $\mathcal{J}^{(k)}$ (ip2);
a = δ[0, 1];

```

$$C_{c1}^{(l)} = a \cdot C_{p1}^{(l)} + (1-a) \cdot C_{p2}^{(l)}; \text{ dla } l=1,2,\dots, \mathcal{L}$$

$$C_{c2}^{(l)} = a \cdot C_{p2}^{(l)} + (1-a) \cdot C_{p1}^{(l)}; \text{ dla } l=1,2,\dots, \mathcal{L}$$

$$C_{c3}^{(l)} = \min(C_{p1}^{(l)}, C_{p2}^{(l)}); \text{ dla } l=1,2,\dots, \mathcal{L}$$

$$C_{c4}^{(l)} = \max(C_{p1}^{(l)}, C_{p2}^{(l)}); \text{ dla } l=1,2,\dots, \mathcal{L}$$

$$\mathcal{J}_4 \leftarrow \text{uczenie\_lokalne}(C_{c1}, C_{c2}, C_{c3}, C_{c4}, E_{NT});$$

$$v_4 \leftarrow \text{ocen}(C_{c1}, C_{c2}, C_{c3}, C_{c4}, E_c);$$

wybierz 2 najlepsze osobniki oceniane względem  $v_4$ ;

gdzie  $C_{p1}$ ,  $C_{p2}$  – chromosomy osobników rodzicielskich,  $C_{p1}^{(l)}$ ,  $C_{p2}^{(l)}$  –  $l$ -ta pozycja w ciągu kodowym chromosomów rodzicielskich,  $\mathcal{J}_4$  – pomocnicze pokolenie osobników potomnych, zawierające cztery osobniki uzyskane w wyniku krzyżowania max-min-arytmetycznego,  $v_4$  – wektor ocen osobników należących do  $\mathcal{J}_4$ . W stosunku do typowego krzyżowania max-min arytmetycznego, wprowadzono losowe dobieranie parametru krzyżowania  $a$ . W celu oceny osobników potomnych wykorzystywana jest funkcja oceniająca z nadrzędnej, głównej pętli strategii ewolucyjnej oraz algorytm lokalnego uczenia parametrów konkluzji reguły. Ponieważ reguła jest potencjalnym kandydatem do ostatecznej bazy wiedzy systemu rozmytego, uczenie przeprowadzane jest z wykorzystaniem całego zbioru uczącego  $E_{NT}$ , a nie lokalnego, niepokrytego zbioru danych uczących.

Powstałe w wyniku krzyżowania pokolenie  $\mathcal{J}_c^{(k)}$  jest poddawane zmodyfikowanej, periodycznej mutacji nierównomierniej, przeprowadzanej wg następującego algorytmu:

```

for i=1 to  $\lambda$ 
   $C_M \leftarrow \mathcal{J}_c^{(k)}[i];$ 
  if  $\delta[0,1] < p_{rM}$  then
     $l = \lceil \delta[0, \mathcal{L}] \rceil;$ 
     $b = \delta\{0,1\};$ 
    if  $l \leq 2 \cdot N$  then
      if  $l \bmod 2 == 0$ 
        if  $b==0$  then

$$C_M^{(l)} = C_M^{(l)} + \Delta(k, \sqrt{\frac{(x_{0,\lceil l/2 \rceil}^{\rightarrow} - x_{0,\lceil l/2 \rceil}^{\leftarrow})^2}{-2 \cdot \ln(G_L)}} - C_M^{(l)});$$

        else

$$C_M^{(l)} = C_M^{(l)} - \Delta(k, C_M^{(l)} - 0.0001);$$

        end
      else

```

```

        if b==0 then
             $C_M^{(j)} = C_M^{(j)} + \Delta(k, x_{0, \lceil l/2 \rceil}^{\rightarrow} - C_M^{(j)})$ ;
        else
             $C_M^{(j)} = C_M^{(j)} - \Delta(k, C_M^{(j)} - x_{0, \lceil l/2 \rceil}^{\leftarrow})$ ;
        end
    end
else
    if b==0 then
         $C_M^{(j)} = C_M^{(j)} + \Delta(k, (t_0^{\rightarrow} - t_0^{\leftarrow}) - C_M^{(j)})$ ;
    else
         $C_M^{(j)} = C_M^{(j)} - \Delta(k, C_M^{(j)} - 0.0001)$ ;
    end
end
end
end
end

```

gdzie  $\delta\{0,1\}$  jest realizacją zmiennej losowej o rozkładzie równomiernym, dwupunktowym na zbiorze  $\{0,1\}$ ,  $C_M$  – chromosom mutowanego  $i$ -tego osobnika ze zbioru pomocniczego  $\mathcal{J}_c^{(k)}$ ,  $C_M^{(j)}$  –  $j$ -ty, mutowany element ciągu kodującego w chromosomie,  $p_{rM}$  – prawdopodobieństwo mutacji stanowiące parametr metody,  $\Delta(k,d)$  – funkcja wyliczająca zmianę wartości parametru reprezentowanego przez osobnika, periodycznie zawężająca zakres zmian wraz ze wzrostem numeru pokolenia:

$$\Delta(l,d) = d \cdot \left( 1 - \delta[0,1] \left( 1 - \frac{\text{mod}(l,10)}{\mathcal{L}} \right)^b \right), \quad (5.16)$$

gdzie  $l$  – numer pokolenia,  $d$  – przedział zmienności wartości,  $b$  – parametr mutacji,  $b > 0$ ,  $\delta[0,1]$  – realizacja zmiennej losowej o rozkładzie równomiernym na przedziale zamkniętym  $[0,1]$ , operator  $\text{mod}(\bullet, \bullet)$  – oznacza operację modulo. Jego zastosowanie znacznie poprawiło zbieżność w ostatniej fazie działania strategii ewolucyjnej, ponieważ zapewniło okresowe wprowadzenie różnorodności do populacji zdominowanej przez osobniki bliskie osobnikowi najlepszemu na wskutek zastosowanej selekcji elitarniej. Wartość 10 znajdująca się w wykładniku funkcji została dobrana na podstawie serii eksperymentów i jest szacunkowa.

### Selekcja elitarna

Nowe pokolenie bazowe  $Q^{(k+1)}$  powstaje poprzez wybór  $\mu$  osobników z połączenia starego pokolenia bazowego oraz pokolenia potomnego:  $Q^{(k)} \cup O^{(k)}$ . Wektor powstały z połączenia ocen osobników:  $\underline{f}$  i  $\underline{f}_O$  jest sortowany rosnąco. Następnie wybiera się  $\mu$  najlepszych osobników z połączenia zbiorów  $Q^{(k)} \cup O^{(k)}$ , wg indeksów wynikających z posortowania odpowiadających im ocen.

### Funkcja oceniająca przystosowanie

Funkcja przystosowania ocenia pojedynczego osobnika-regułę rozmytą z parametryzowaną konkluzją  $R$  w zakresie opisu danych uczących  $E_C$  oraz zależności pomiędzy ocenianą regułą i innymi, wygenerowanymi uprzednio regułami. Ocena jest wykonywana z wykorzystaniem bieżącego, pokrywanego zbioru uczącego  $E_C$  stanowiącego podzbiór zbioru  $E_{NT}$  oraz uwzględnia dotychczas stworzoną bazę wiedzy  $\mathcal{R}$ . Ocena ta składa się z iloczynu algebraicznego pięciu ocen cząstkowych. Wartość  $f$  funkcji  $\text{ocene}(\bullet, \bullet, \bullet)$  oceniającej osobnika-regułę  $R$  jest następująca:

$$f(R, \mathcal{R}, E_C) = \bar{\Psi}_{NT}(R, E_C) \cdot C_p(R, E_C) \cdot C_n(R, E_C) \cdot \mathcal{G}(R, E_C) \cdot \Xi(R, \mathcal{R}), \quad (5.17)$$

gdzie poszczególne oceny cząstkowe składające się na ostateczną wartość funkcji przystosowania  $f$  są zdefiniowane następująco:

- $\bar{\Psi}_{NT}(R, E_C)$  - średnia kompatybilność reguły  $R$  z danymi uczącymi  $E_C$ :

$$\bar{\Psi}_{NT}(R, E_C) = \frac{1}{N_{EC}} \sum_{i=1}^{N_{EC}} R_c(R, E_C[i]), \quad (5.18)$$

gdzie  $R_c(R, E_C[i])$  – stopień kompatybilności reguły  $R$  z  $i$ -tym rekordem ze zbioru uczącego  $E_C$ ,  $N_{EC}$  -  $\text{card}(E_C)$ . Większa wartość funkcji  $\bar{\Psi}_{NT}$  oznacza, że oceniana reguła lepiej opisuje zestaw rekordów ze zbioru uczącego  $E_C$ . Stopień kompatybilności reguły  $R_c(R, E_C[i])$  jest wyznaczany następująco:

$$R_c(R, E_C[i]) = R_c(R, \{x_0(i), t_0(i)\}) = \mu_A(x_0(i)) \star_{T_c} \mu_B(t_0(i)), \quad (5.19)$$

gdzie  $E_C[i] = \{x_0(i), t_0(i)\}$  –  $i$ -ty rekord z podzbioru zbioru uczącego  $E_C$ ,  $\mu_A, \mu_B$  – funkcje przynależności, odpowiednio przesłanek i konkluzji reguły  $R$ ,  $\star_{T_c}$  -  $t$ -norma stosowana w przypadku koniunkcyjnej interpretacji reguł.

- $C_p(R, E_C)$  - średnie pokrycie danych ze zbioru  $E_C$  przez regułę  $R$ :

$$C_p(R, E_C) = \sum_{\{[x_0(i), t_0(i)] \in E_\omega^+(R, E_C)\}} \frac{R_c(R, [x_0(i), t_0(i)])}{n_\omega^+}, \quad (5.20)$$

$$E_\omega^+(R, E_C) = \{[x_0(i), t_0(i)] \in E_C \mid R_c(R, [x_0(i), t_0(i)]) \geq \omega\}, \quad (5.21)$$

gdzie  $E_\omega^+(R, E_C)$  - zbiór rekordów ze zbioru uczącego  $E_C$ , dla których stopień kompatybilności z regułą  $R$  jest większy lub równy od wartości  $\omega$  stanowiącej parametr metody,  $n_\omega^+$  - liczebność zbioru  $E_\omega^+$ .

- $C_n(R, E_C)$  - stopień dopasowania konkluzji reguły  $R$ :

$$C_n(R, E_C) = \begin{cases} 1 & n^- \leq \kappa \cdot n_\omega^+ \\ \frac{1}{n^- - \kappa \cdot n_\omega^+ + \exp(1)} & n^- > \kappa \cdot n_\omega^+ \end{cases} \quad (5.22)$$

gdzie  $\kappa$  – parametr metody,  $\kappa \in [0,1]$ ,  $n^-$  – liczebność zbioru  $E^-(R, E_C)$ , zdefiniowanego następująco:

$$E^-(R, E_C) = \{(x_0(i), t_0(i)) \in E_C \mid R_c(R, [x_0(i), t_0(i)]) = 0 \wedge \mu_{\underline{A}}(x_0(i)) > 0\} \quad (5.23)$$

Dla reguł, dla których zbiór pokrywanych rekordów uczących jest większy o ustalony stopień  $\kappa$  od zbioru rekordów, których ta reguła nie pokrywa, funkcja  $C_n$  przyjmuje wartość 1, natomiast w przeciwnym wypadku przyjmuje tym mniejszą wartość, im większa jest dysproporcja pomiędzy liczbą rekordów pokrywanych i nie pokrywanych przez rozmytą regułę  $R$ .

- $\mathcal{G}(R, E_C)$  - poziom szerokości dyspersji przesłanek i nośnika konkluzji reguły  $R$  względem szerokości przedziałów zmienności argumentów wejściowych i wartości wyjściowych ze zbioru uczącego  $E_C$  :

$$\mathcal{G}(R, E_C) = \frac{\exp\left(1 - \left(\frac{w}{t_0^+ - t_0^-} + \sum_{n=1}^N \frac{2 \cdot \sqrt{-0.5 \cdot \sigma_n^2 \cdot \ln(G_L)}}{x_{0,n}^+ - x_{0,n}^-}\right)\right) - 1}{\exp(1) - 1}, \quad (5.24)$$

$$\begin{aligned} t_0^- &= \min_{\{i \mid [x_0(i), t_0(i)] \in E_C\}} (t_0(i)) \\ t_0^+ &= \max_{\{i \mid [x_0(i), t_0(i)] \in E_C\}} (t_0(i)), \end{aligned} \quad (5.25)$$

$$\begin{aligned} x_{0,n}^- &= \min_{\{i \mid [x_0(i), t_0(i)] \in E_C\}} (x_{0,n}(i)) \\ x_{0,n}^+ &= \max_{\{i \mid [x_0(i), t_0(i)] \in E_C\}} (x_{0,n}(i)), \end{aligned} \quad (5.26)$$

gdzie  $w$ ,  $\sigma_n$  – parametry zbiorów rozmytych reguły  $R$ ,  $G_L$  – parametr metody. Ta ocena cząstkowa zapobiega tworzeniu jednej reguły o szerokich zbiorach w przesłankach i konkluzji, która mogłaby „zdominować” bazę wiedzy systemu i pokryć znaczącą część danych ze zbioru  $E_C$ .

- $\Xi(R, \mathcal{R})$  - ocena podobieństwa reguły  $R$  do reguł zawartych w uprzednio wygenerowanej bazie wiedzy  $\mathcal{R}$ :

$$\Xi(R, \mathcal{R}) = 1 - \max_{\{i \mid R^{(i)} \in \mathcal{R}\}} \left( \mu_{\underline{A}}(\underline{c}_x^{(i)}) \star_{Tc} \mu_B(c_y^{(i)}) \right), \quad (5.27)$$

gdzie  $\underline{c}_x^{(i)}$  – wektor środków funkcji gaussowskich przesłanek  $i$ -tej reguły,  $c_y^{(i)}$  – środek trójkątnej funkcji przynależności konkluzji  $i$ -tej reguły z bazy wiedzy  $\mathcal{R}$ ,  $\star_{T_c}$  -  $t$ -norma stosowana w przypadku koniunkcyjnej interpretacji reguł. Ponieważ zarówno reguła  $R$  jak i reguły  $R^{(i)} \in \mathcal{R}$  są regułami systemu z parametryzowanymi konkluzjami, istotny jest sposób wyznaczania środków trójkątnych funkcji przynależności konkluzji. W tym celu reguły są uczone lokalnie za pomocą zbioru uczącego  $E_{NT}$ , a następnie jako argumenty wejściowe do wyliczenia położenia środka ciężkości konkluzji przyjmuje się wektor wartości jednostkowych o wymiarze  $N$ . Ta ocena cząstkowa odpowiedzialna jest za niedopuszczenie do wygenerowania przez strategię ewolucyjną na poziomie iteracyjnego algorytmu pokrywania jednakowych reguł. Jest to jeden z elementów wspomagających redukcję bazy wiedzy, wprowadzony już na etapie generowania reguł, celem niedopuszczenia do nadmiernego rozrostu liczby reguł i tym samym zafałszowania faktycznych zdolności pokrywania danych ze zbioru uczącego na poziomie algorytmu pokrywającego.

### Podsumowanie

Algorytm generowania reguł w wyniku swojego działania zwraca do algorytmu pokrywającego najlepszą odnalezioną w procesie ewolucyjnym regułę, opisującą podzbiór  $E_C$  zbioru uczącego  $E_{NT}$ . W części jawnej reguła zawiera wyłącznie parametry funkcji przynależności przesłanek i szerokość nośnika konkluzji, nie zawiera natomiast parametrów kombinacji liniowej, opisującej położenie środka ciężkości trójkątnej funkcji przynależności w konkluzji. Parametry te są zawarte w części ukrytej osobnika i każdorazowo uczone, w miarę potrzeb wynikających z kolejnych kroków algorytmu. Tam gdzie było to możliwe (gdy nie ulegały modyfikacji parametry opisujące funkcje przynależności), ze względów optymalizacyjnych wykorzystywano wyliczone wcześniej parametry konkluzji.

## 5.2 Redukcja bazy reguł

Uzyskana w pierwszym etapie trójetapowej metody uczenia baza wiedzy może zawierać reguły redundantne lub takie, których obecność jest niepożądana, gdyż negatywnie wpływają na jakość systemu rozmytego. Najczęściej też zawiera ona zbyt wiele reguł, co może być przyczyną efektu „przeuczenia” systemu. Z tego powodu zaproponowano mechanizm redukcji bazy reguł, opierający się na zmodyfikowanej postaci prostego algorytmu genetycznego. W algorytmie tym zastosowano zapamiętywanie i dołączanie do kolejnego pokolenia najlepszego spośród dotychczas wygenerowanych osobników, o ile w kolejnym pokoleniu nie pojawił się osobnik lepszy od wszystkich dotychczas wygenerowanych. W tym etapie trójetapowej metody uczenia, optymalizacji podlegał wyłącznie skład bazy reguł, natomiast parametry rozmytych zbiorów przynależności w przesłankach i konkluzji nie były w żaden sposób modyfikowane. W algorytmie wykorzystano kodowanie binarne, krzyżowanie dwupunktowe oraz klasyczną mutację stosowaną w prostym algorytmie genetycznym. Funkcja oceny osobników uwzględnia zarówno skład bazy reguł pod względem jakości tworzonego systemu, jak i poziom pokrycia danych ze zbioru uczącego. Szczegółowy opis elementów tworzących metodę redukcji bazy reguł znajduje się w dalszej części niniejszego podrozdziału.



Alternatywnie, dla przypadków, w których wygenerowana baza reguł składała się z mniej niż 10 reguł rozmytych (wartość wyliczona empirycznie, odpowiadająca akceptowalnemu czasowi obliczeń), stosowano przeszukiwanie wszystkich możliwych kombinacji reguł metodą prostego wyszukiwania. Należy jednak zwrócić uwagę na fakt, iż złożoność obliczeniowa tego podejścia jest wyjątkowo niekorzystna (rzędu  $2^i$ , gdzie  $i$ -liczba reguł), dlatego też niezbędne stało się wykorzystanie do optymalizacji bazy reguł algorytmu genetycznego dla przypadków, w których baza wiedzy składa się z większej liczby reguł warunkowych **if-then**.

Część pracy związanej z optymalizacją bazy reguł została wykonana w pierwszym etapie trójetapowej metody uczenia, ponieważ składowa  $\Xi$  funkcji oceny osobników wpływa pozytywnie na generowanie reguł o większej różnorodności w sensie funkcji przynależności w przesłankach i konkluzji.

Zastosowany algorytm redukcji bazy reguł zaprezentowano poniżej:

```

 $Q^{(0)} \leftarrow \text{inicjalizuj}(M)$  ;
 $k = 0$ ;
 $\underline{f} \leftarrow \text{ocień}(Q^{(0)})$  ;
 $q_B \leftarrow \text{najlepszy}(Q^{(0)})$  ;
while  $k < K$ 
     $\mathcal{J}^{(k)} = \text{preselekcja\_proporcjonalna}(Q^{(k)})$  ;
     $\mathcal{J}_c^{(k)} = \text{krzyżowanie\_dwupunktowe}(\mathcal{J}^{(k)})$  ;
     $O^{(k)} = \text{mutacja}(\mathcal{J}_c^{(k)})$  ;
     $\underline{f}_O \leftarrow \text{ocień}(O^{(k)})$  ;
    if  $\text{ocień}(q_B) > \max(\underline{f}_O)$  then
        wymień w  $O^{(k)}$  najgorszego osobnika na  $q_B$  ;
    else
         $q_B \leftarrow \text{best}(Q^{(k)})$  ;
    end
     $Q^{(k+1)} = O^{(k)}$  ;
     $k = k + 1$  ;
end

```

gdzie  $k$  – numer pokolenia,  $Q^{(k)}$  –  $k$ -te pokolenie bazowe,  $\mathcal{J}^{(k)}$  –  $k$ -te pokolenie rodziców (tymczasowe),  $\mathcal{J}_c^{(k)}$  –  $k$ -te pokolenie tymczasowe, powstałe po operacji krzyżowania,  $O^{(k)}$  –  $k$ -te pokolenie potomne,  $K$  – maksymalna liczba pokoleń (warunek zatrzymania algorytmu genetycznego). Funkcja  $\text{inicjalizuj}(M)$  generuje pokolenie startowe  $M$  osobników, funkcja  $\text{ocień}(O^{(k)})$  ocenia pokolenie, funkcja  $\text{najlepszy}(\bullet)$  wybiera najlepszego osobnika z populacji, funkcja  $\text{preselekcja\_proporcjonalna}(Q^{(k)})$  wybiera klasyczną metodą proporcjonalną (ze zwracaniem) pokolenie rodziców, stanowiące podstawę dla dalszych operacji genetycznych. Podczas całej pracy algorytmu genetycznego jest zapamiętywany najlepszy, dotychczas wygenerowany osobnik  $q_B$ . Jeżeli po obliczeniu wartości funkcji przystosowania dla pokolenia

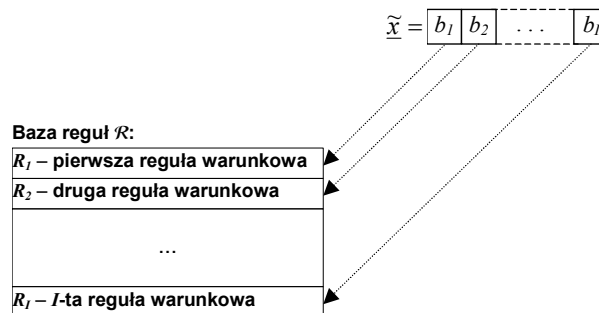
$O^{(k)}$  okaże się, że wszystkie osobniki mają ocenę przystosowania niższą od oceny zapamiętanego, najlepszego dotychczas osobnika  $q_B$ , w miejsce najgorszego osobnika w pokoleniu  $O^{(k)}$  wstawia się osobnika  $q_B$ . W przeciwnym wypadku osobnik z pokolenia  $O^{(k)}$  o odpowiadającej mu największej wartości funkcji przystosowania zostaje zapamiętany w miejsce  $q_B$ , jako najlepszy dotychczas odnaleziony.

### Kodowanie osobników

W algorytmie zastosowano binarne kodowanie osobników. Długość łańcuchów kodujących jest stała i równa liczebności reguł uzyskanych w pierwszym etapie trójetapowej metody uczenia. Każdy osobnik reprezentuje system rozmyty (analogicznie do podejścia Pittsburgh) i przechowuje informacje o składzie bazy reguł, przy czym pojedynczy  $l$ -ty gen informuje jedynie o tym, czy reguła znajduje się w bazie wiedzy (wartość na  $l$ -tej pozycji równa 1) czy też nie wchodzi w jej skład (wartość na  $l$ -tej pozycji równa 0).

Parametry funkcji przynależności przesłanek i konkluzji nie są przechowywane w chromosomie. Zakłada się przy tym ich stałą wartość, równą wartościom wygenerowanym w pierwszym etapie trójetapowej metody uczenia.

Na poniższym rysunku przedstawiono schematycznie sposób kodowania w etapie redukcji bazy reguł:



Rysunek 5.4: Kodowanie reguł w drugim etapie trójetapowej metody uczenia.

Formalnie wektor zawierający chromosom ma następującą postać:

$$\tilde{x} = [b_1, \dots, b_i, \dots, b_I]^T, \quad (5.28)$$

gdzie  $I = \text{card}(\mathcal{R})$ ,  $b_i \in \{0,1\}$ .

Reguły w bazie reguł  $\mathcal{R}$  uzyskanej w pierwszym etapie trójetapowej metody muszą być ponumerowane (numeracja musi być niezmienna), aby loci osobnika jednoznacznie identyfikowały reguły, które tworzą reprezentowany system rozmyty.

Dla ułatwienia działania operatorów (w szczególności operatora mutacji), pokolenie osobników jest przechowywane w postaci macierzy o wymiarach  $M \times I$ :

$$Q^{(k)} = \begin{bmatrix} b_{11} & \cdots & b_{1I} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MI} \end{bmatrix} \begin{matrix} \text{- osobnik 1} \\ \vdots \\ \text{- osobnik M} \end{matrix}, \quad (5.29)$$

gdzie  $M$  – liczba osobników,  $I$  – liczba reguł w bazie reguł  $\mathcal{R}$ .

### Inicjalizacja pokolenia startowego

Pokolenie startowe jest generowane dwojako. Pierwsze  $(M-1)$  osobników pokolenia powstaje w sposób losowy, natomiast pozostały jeden osobnik reprezentuje pełny system rozmyty, zawierający wszystkie reguły z bazy reguł  $\mathcal{R}$  uzyskanej w pierwszym etapie trójetapowej metody uczenia. Wszystkie pozycje ciągu kodowego tego osobnika zawierają wartość 1. Do wygenerowania ciągów kodowych osobników wykorzystano realizację zmiennej losowej o rozkładzie równomiernym, dwupunktowym na zbiorze  $\{0,1\}$ . Powstałe w ten sposób pokolenie tworzy macierz o postaci zaprezentowanej powyżej.

### Preselekcja osobników rodzicielskich

Do preselekcji osobników rodzicielskich wykorzystano najczęściej stosowaną metodę proporcjonalną, w której prawdopodobieństwo wyboru osobnika do pokolenia rodziców  $\mathcal{J}$  jest proporcjonalne do jego oceny. Spośród  $M$  osobników populacji  $Q$  wybieranych jest  $M$  osobników, ze zwracaniem. W ten sposób powstaje populacja  $\mathcal{J}$ .

Szczegółowy algorytm preselekcji przedstawiono poniżej:

```

s =  $\sum_{i=1}^M f(i)$  ;
if s==0 then
    vs=ones(M)/M ;
else
    for j = 1 to M
         $v_s(j) = \frac{\sum_{i=1}^j f(i)}{s}$  ;
    end
end
m =  $\delta(0,1)$  ;

```

```

for  $i = 1$  to  $M$ 
   $j = 0$  ;
  while  $(v_s(j) \leq m) \ \& \ (j < M)$ 
     $j = j + 1$  ;
  end
   $\mathcal{J}^{(k)}[i] \leftarrow Q^{(k)}[i]$  ;
   $m = \delta[0,1]$  ;
end

```

gdzie  $\underline{f}$  - wektor ocen osobników pokolenia  $Q^{(k)}$ ,  $v_s$  – wektor zawierający znormalizowaną sumę narastającą wartości ocen osobników,  $\delta[0,1]$  – realizacja zmiennej losowej o rozkładzie równomiernym na przedziale  $[0,1]$ .

### Operatory

Populacja rodzicielska  $\mathcal{J}^{(k)}$  podlega dwóm operacjom genetycznym. W pierwszej kolejności jest wykonywana operacja krzyżowania dwupunktowego, następnie na tak uzyskanym pokoleniu pośrednim  $\mathcal{J}_c^{(k)}$  jest przeprowadzana operacja mutacji.

Do pojedynczej operacji krzyżowania losowo wybiera się dwa osobniki rodzicielskie z puli  $\mathcal{J}^{(k)}$ , bez zwracania. Wybór następuje z jednakowym prawdopodobieństwem – na tym etapie nie jest uwzględniana ranga osobników. Wybrana para osobników podlega krzyżowaniu dwupunktowemu z prawdopodobieństwem  $p_{rc}$ , które stanowi parametr metody redukcji bazy reguł. Osobniki potomne (lub jeśli para nie podlegała krzyżowaniu, osobniki rodzicielskie) są dołączane do pokolenia pośredniego  $\mathcal{J}_c^{(k)}$ . Jeśli liczba osobników jest nieparzysta, pozostały osobnik z pokolenia  $\mathcal{J}^{(k)}$  jest dołączany do pokolenia  $\mathcal{J}_c^{(k)}$  bez zmian. Punkty rozcięcia chromosomu są losowane z rozkładem równomiernym wielopunktowym na zbiorze  $\{1,2,\dots,\mathcal{L}\}$ :

$$\begin{aligned} \underline{\tilde{x}}_\alpha &= [c_{\alpha,1}, \dots, c_{\alpha,\mathcal{L}}]^T & \Rightarrow & \quad \underline{\tilde{x}}_1 = [c_{\alpha,1}, \dots, c_{\alpha,\zeta}, c_{\beta,\zeta+1}, \dots, c_{\beta,\gamma}, c_{\alpha,\gamma+1}, \dots, c_{\alpha,\mathcal{L}}]^T \\ \underline{\tilde{x}}_\beta &= [c_{\beta,1}, \dots, c_{\beta,\mathcal{L}}]^T & & \quad \underline{\tilde{x}}_2 = [c_{\beta,1}, \dots, c_{\beta,\zeta}, c_{\alpha,\zeta+1}, \dots, c_{\alpha,\gamma}, c_{\beta,\gamma+1}, \dots, c_{\beta,\mathcal{L}}]^T, \end{aligned} \quad (5.30)$$

gdzie  $\underline{\tilde{x}}_\alpha$ ,  $\underline{\tilde{x}}_\beta$  - osobniki rodzicielskie,  $\underline{\tilde{x}}_1$ ,  $\underline{\tilde{x}}_2$  - osobniki potomne,  $c_l \in \{0,1\}$  - pozycja ciągu kodowego,  $\zeta, \gamma$  - miejsca krzyżowania i  $l=1,2,\dots,\mathcal{L}$ ,  $\zeta \in \{1,2,\dots,\mathcal{L}\}$ ,  $\gamma \in \{1,2,\dots,\mathcal{L}\}$ .

Algorytm operacji krzyżowania przedstawiono poniżej:

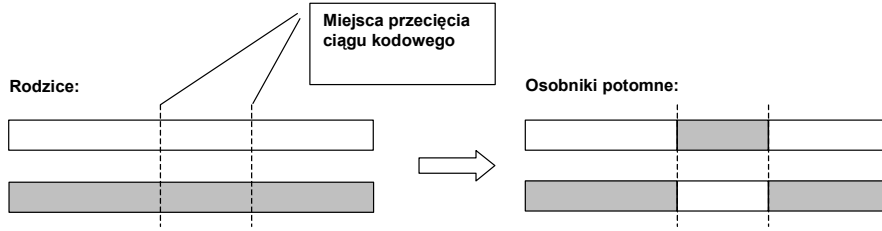
```

VL=ones(M);
Jc(k)=∅;
while  $\sum_{i=1}^M V_L(i) > 1$ 
    iP1 =  $\left\lceil \delta \left( 0, \sum_{i=1}^M V_L(i) \right) \right\rceil$ ;
    jP1 ← J(k)[index(iP1)] ;
    VL[index(iP1)] = 0 ;
    iP2 =  $\left\lceil \delta \left( 0, \sum_{i=1}^M V_L(i) \right) \right\rceil$ ;
    jP2 ← J(k)[index(iP2)] ;
    VL[index(iP2)] = 0 ;
    if δ(0,1) < Prc then
        Jc(k) ← Jc(k) ∪ krzyżowanie dwupunktowe(jP1, jP2) ;
    else
        Jc(k) ← Jc(k) ∪ [jP1, jP2] ;
    end
end
if  $\sum_{i=1}^M V_L(i) = 1$  then
    Jc(k) ← Jc(k) ∪ pozostały osobnik(J(k)) ;
end

```

gdzie  $V_L$  – wektor wyznaczający, które osobniki z pokolenia nie zostały jeszcze wybrane do krzyżowania (wartość dla tej pozycji wynosi 1), a które zostały już wybrane (wartość dla tej pozycji wynosi 0),  $i_{P1}$ ,  $i_{P2}$  – wskaźniki osobników krzyżowanych,  $j_{P1}$ ,  $j_{P2}$  – osobniki (wektory binarne) podlegające krzyżowaniu,  $P_{rc}$  – prawdopodobieństwo krzyżowania, stanowiące parametr metody, funkcja **krzyżowanie dwupunktowe**(•,•) wykonuje operację krzyżowania dwupunktowego, tak jak to zaprezentowano na rysunku poniżej, zwracając dwa nowe osobniki powstałe ze wskazanych osobników rodzicielskich. Zapis  $J_c^{(k)} \leftarrow J_c^{(k)} \cup \bullet$  oznacza operację dodania do populacji  $J_c^{(k)}$  wskazanych osobników.

Schematycznie operację krzyżowania dwupunktowego zaprezentowano na poniższym rysunku:



Rysunek 5.5: Operacja krzyżowania zastosowana w redukcji bazy reguł.

Pokolenie  $\mathcal{J}_c^{(k)}$ , powstałe w wyniku przeprowadzenia operacji krzyżowania na pokoleniu  $\mathcal{J}^{(k)}$ , poddawane jest operacji mutacji. Ze względu na macierzową postać przechowywania pokolenia  $\mathcal{J}_c^{(k)}$ , operacja ta została uproszczona i zrealizowana w następujący sposób:

$$\underline{Q}^{(k)} = \underline{\mathcal{J}}_c^{(k)} \oplus \underline{M}_m, \quad (5.31)$$

gdzie  $\oplus$  - operator XOR wykonywany element po elemencie,  $\underline{Q}^{(k)}$  oraz  $\underline{\mathcal{J}}_c^{(k)}$  – macierze binarne pokoleń o organizacji wierszowej (Rysunek 2.1),  $\underline{M}_m$  – macierz mutująca, uzyskana w następujący sposób:

$$\underline{M}_m = \begin{bmatrix} \delta(0,1) < p_{rM} & \cdots & \delta(0,1) < p_{rM} \\ \vdots & \ddots & \vdots \\ \delta(0,1) < p_{rM} & \cdots & \delta(0,1) < p_{rM} \end{bmatrix}, \quad (5.32)$$

gdzie  $(\delta(0,1) < p_{rM}) \in \{0,1\}$  - przyjmuje wartość równą 1, jeżeli wylosowana wartość jest mniejsza od prawdopodobieństwa mutacji  $p_{rM}$  oraz 0 w przeciwnym wypadku.

### Selekcja

Zastosowano standardową selekcję stosowaną w algorytmach genetycznych, zmodyfikowaną o element elitarny. Osobnik o najwyższej wartości funkcji przystosowania jest zachowywany i dołączany do kolejnego pokolenia bazowego (w miejsce osobnika o najniższej wartości funkcji przystosowania) w przypadku, gdy w wyniku przeprowadzenia operacji genetycznych wszystkie osobniki mają ocenę niższą niż wartość funkcji przystosowania osobnika zachowanego. Przeprowadzone doświadczenia wskazały pozytywny wpływ tego mechanizmu na zbieżność opisywanego algorytmu genetycznego.

### Funkcja oceniająca przystosowanie

Funkcja oceniająca przystosowanie osobnika uwzględnia zarówno jakość generowanego systemu jak i zapewnia kompletność ocen bazy wiedzy, dlatego też ma postać złożoną:

$$f(\tilde{x}, E_{NT}) = \frac{1}{\sqrt{f_J(\tilde{x}, E_{NT})}}, \quad (5.33)$$

gdzie  $f_J$  – funkcja kary, o następującej postaci:

$$f_J(\tilde{x}, E_{NT}) = \begin{cases} Js_{RMSE}(\tilde{x}, E_{NT}) & \text{jeżeli } J_R(\tilde{x}, E_{NT}) \geq \tau \\ \sum_{n_T=1}^{N_T} (t_0(n_T))^2 & \text{jeżeli } J_R(\tilde{x}, E_{NT}) < \tau \end{cases}, \quad (5.34)$$

gdzie  $J_R$  - wskaźnik kompletności pokrycia, zdefiniowany jako średnie pokrycie przypadające na pojedynczą regułę, tworzącą system rozmyty reprezentowany przez osobnika:

$$J_R(\tilde{x}, E_{NT}) = \frac{\sum_{n_T=1}^{N_T} \sum_{i=1}^{I_{\tilde{x}}} R_c(R^{(i)}, E_{NT}[n_T])}{N_T \cdot I_{\tilde{x}}}, \quad (5.35)$$

gdzie  $I_{\tilde{x}}$  - liczebność bazy reguł reprezentowanej przez osobnika  $\tilde{x}$ ,  $R_c(R^{(i)}, E_{NT}[n_T])$  - kompatybilność  $i$ -tej reguły z  $n_T$ -tym elementem danych ze zbioru uczącego, wyliczana w sposób opisany w pierwszym etapie trójetapowej metody uczenia. Wskaźnik jakości  $Js_{RMSE}$  bazujący na błędzie średniokwadratowym (ang. *Root Mean Square Error*) jest wyliczany następująco:

$$Js_{RMSE}(\tilde{x}, E_{NT}) = \frac{1}{2 \cdot N_T} \sum_{n_T=1}^{N_T} (t_0(n_T) - y_0(n_T, E_{NT}(n_T)))^2, \quad (5.36)$$

gdzie  $y_0$  - wartość wyjściowa, generowana przez system rozmyty reprezentowany przez osobnika o chromosomie  $\tilde{x}$ .

### Podsumowanie

Jak wspomniano w części wstępnej podrozdziału, dla baz reguł otrzymanych w pierwszym etapie trójetapowej metody uczenia, których wielkość nie przekraczała 10 reguł stosowano wyszukiwanie wyczerpujące, ze względu na mniejszy nakład obliczeniowy związany z uruchomieniem obliczeń i ostatecznie krótszy czas optymalizacji. Jednak dla przestrzeni poszukiwań składającej się z kilkunastu i więcej reguł, czas optymalizacji metodą genetyczną był znacznie krótszy od czasu optymalizacji metodą wyszukiwania wyczerpującego. Różnica ta oczywiście wzrasta wraz ze wzrostem liczby reguł w bazie  $\mathcal{R}$ .

## 5.3 Strojenie bazy danych systemu rozmytego

Otrzymana w drugim etapie baza danych zredukowanej bazy wiedzy  $\mathcal{R}_{II}$  poddawana była w etapie trzecim lokalnemu dostrajaniu. Do strojenia wykorzystano strategię ewolucyjną ( $\mu+\lambda$ ) oraz osobniki o kodowaniu zmiennoprzecinkowym, w którym pojedynczy osobnik koduje całą bazę wiedzy (zgodnie z podejściem Pittsburgh). W tym etapie liczba reguł w bazie wiedzy  $\mathcal{R}_{II}$  jest stała, a modyfikacji podlega jedynie baza danych, zawierająca parametry funkcji przynależności przesłanek i konkluzji rozmytych reguł warunkowych **if-then**. Podobnie jak w poprzednich etapach trójetapowej metody uczenia, parametry konkluzji nie podlegają procesowi ewolucyjnemu i stanowią część *niejawną* kodowania osobników. Ponieważ pojedynczy osobnik reprezentuje jeden system rozmyty, tam gdzie było to niezbędne

wykorzystywano uczenie parametrów konkluzji metodą globalną. Jako operatory genetyczne zastosowano podobnie jak w pierwszym etapie metody trójetapowej krzyżowanie max-min-arytmetyczne oraz periodyczną mutację nierównomierną. Wybór osobników rodzicielskich przeprowadzono z wykorzystaniem próbkowania stochastycznego. Kolejne pokolenie bazowe powstawało z połączenia pokolenia bazowego i potomnego, z wykorzystaniem selekcji elitarniej. Poniżej przedstawiono algorytm strategii ewolucyjnej lokalnego dostrajania parametrów funkcji przynależności w przesłankach i konkluzjach reguł:

```

wylicz  $\underline{x}_0^{\leftarrow}, \underline{x}_0^{\rightarrow}$  oraz  $t_0^{\leftarrow}, t_0^{\rightarrow}$ ;
 $Q^{(0)} \leftarrow$  inicjalizuj( $\mu, N_T$ );
 $k = 0$ ;
while  $k < K$ 
    if  $k == 0$ 
         $\underline{f} \leftarrow$  oceń( $Q^{(k)}, E_{NT}$ );
    end
    [ $\mathcal{J}^{(k)}, \underline{f}_{\mathcal{J}}$ ]  $\leftarrow$  próbkowanie stochastyczne( $Q^{(k)}, \lambda$ ); wybór  $\lambda$  osobników
    [ $\mathcal{J}_c^{(k)}, \underline{f}_{\mathcal{J}_c}$ ]  $\leftarrow$  krzyżowanie( $\mathcal{J}^{(k)}, \underline{f}_{\mathcal{J}}$ );
    [ $\mathcal{O}^{(k)}, \underline{f}_{\mathcal{O}}$ ]  $\leftarrow$  mutacja( $\mathcal{J}_c^{(k)}, \underline{f}_{\mathcal{J}_c}$ );
     $Q^{(k+1)} \leftarrow$  selekcja elitarna  $\mu$  osobników z  $Q^{(k)} \cup \mathcal{O}^{(k)}$ ;
     $\underline{f} \leftarrow$  przepisanie ocen wybranych osobników z  $\underline{f}$  i  $\underline{f}_{\mathcal{O}}$  zgodnie z  $Q^{(k+1)}$ ;
     $k = k + 1$ ;
end

```

gdzie  $k$ -licznik pokoleń,  $K$ -maksymalna liczba pokoleń (warunek zatrzymania strategii ewolucyjnej),  $\underline{f}_{\mathcal{J}}, \underline{f}_{\mathcal{J}_c}, \underline{f}_{\mathcal{O}}$  - wektory ocen pokoleń odpowiednio  $\mathcal{J}^{(k)}, \mathcal{J}_c^{(k)}, \mathcal{O}^{(k)}$ .

Powyższy algorytm jest bardzo podobny do zastosowanego w pierwszym etapie trójetapowej metody uczenia, przy czym odmiennie niż to miało miejsce w pierwszym etapie, ocena poszczególnych osobników jest dokonywana bezpośrednio po operacjach preselekcji, krzyżowania i mutacji, odpowiednio wewnątrz funkcji **próbkowanie stochastyczne**( $\bullet, \bullet$ ), **krzyżowanie**( $\bullet, \bullet$ ), **mutacja**( $\bullet, \bullet$ ). Wartości są wyliczane, o ile zachodzi taka potrzeba (zmianie uległ którykolwiek z parametrów bazy danych systemu), ze względu na znaczny nakład obliczeniowy związany z wykonaniem operacji wyliczenia wskaźnika jakości dla całego systemu rozmytego przy założonym zbiorze danych  $E_{NT}$ . Zapis [ $\bullet, \bullet$ ]  $\leftarrow \bullet$  oznacza że funkcja zwraca wiele wartości (więcej niż jedną).

### Kodowanie osobników

W trzecim etapie trójetapowej metody uczenia zastosowano kodowanie zmiennoprzecinkowe, z chromosomami o stałej długości, zawierającymi wszystkie parametry bazy danych reprezentowanego systemu rozmytego. Ponieważ jedna reguła rozmyta jest kodowana za pomocą  $2 \cdot N + 1$  elementów, pełen chromosom składa się z  $I_{II} \cdot (2 \cdot N + 1)$  wartości, gdzie  $I_{II}$  oznacza liczbę reguł w  $\mathcal{R}_{II}$ . Utworzenie systemu rozmytego z chromosomu wymaga

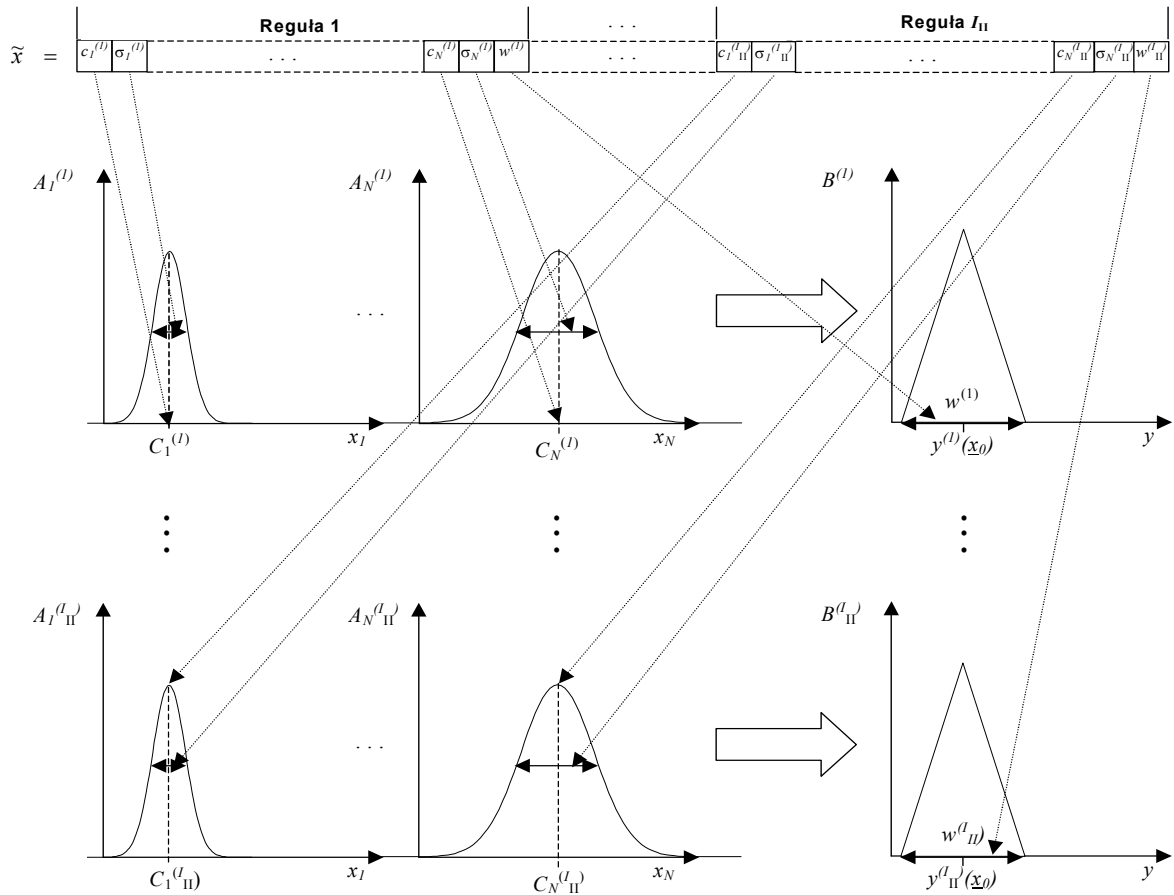


znajomości (w funkcji dekodującej chromosom) numeru reguły  $i_{II}$  i numeru  $n$  wejścia, na podstawie locus  $l$  w chromosomie:

$$i_{II} = \left\lceil \frac{l}{2 \cdot N + 1} \right\rceil, \quad (5.37)$$

$$n = \left\lceil \frac{l \bmod (2 \cdot N + 1)}{2} \right\rceil \text{ dla } (l \bmod (2 \cdot N + 1)) \neq 0, \quad (5.38)$$

gdzie  $i_{II}$  – numer reguły w  $\mathcal{R}_{II}$ ,  $N$  – liczba wejść systemu. Jeżeli  $(l \bmod (2 \cdot N + 1)) = 0$ , parametr określa szerokość nośnika w konkluzji reguły, natomiast dla  $(l \bmod (2 \cdot N + 1)) \neq 0$ , parametr opisuje funkcję przynależności przesłanki, i jest to odpowiednio dla  $(l \bmod (2 \cdot N + 1) \bmod 2) = 1$  środek krzywej gaussowskiej, a dla  $(l \bmod (2 \cdot N + 1) \bmod 2) = 0$  dyspersja tej krzywej w przesłance rozmytej reguły warunkowej **if-then**. Schematycznie kodowanie zastosowane w tym etapie trójetapowej metody uczenia przedstawiono na poniższym rysunku:



Rysunek 5.6: Kodowanie bazy danych w trzecim etapie trójetapowej metody uczenia.

### Inicjalizacja pokolenia startowego

Pierwszy osobnik pokolenia startowego jest osobnikiem powstałym z połączenia bazy reguł  $\mathcal{R}_{II}$  i odpowiadających tym regułom parametrów funkcji przynależności przesłanek i konkluzji z bazy danych  $\mathcal{R}$ , uzyskanej w pierwszym etapie trójetapowej metody uczenia. W przypadku, gdy trzeci etap trójetapowej metody jest uruchamiany wielokrotnie, osobnik ten posiada bazę reguł  $\mathcal{R}_{II}$ , natomiast parametry opisujące gaussowskie funkcje przynależności w przesłankach oraz szerokość nośnika w konkluzji pochodzą od najlepszego osobnika uzyskanego w poprzednich uruchomieniach trzeciego etapu uczenia. Pozostałe  $\mu-1$  osobników pokolenia startowego jest tworzone losowo, z rozkładem równomiernym na odpowiednich przedziałach zmienności parametrów funkcji przynależności przesłanek i konkluzji. Pełny algorytm generowania pokolenia startowego został zamieszczony poniżej:

```

 $[\underline{c}_0^{\leftarrow}, \underline{c}_0^{\rightarrow}, \underline{\sigma}_0^{\leftarrow}, \underline{\sigma}_0^{\rightarrow}, \underline{w}_0^{\leftarrow}, \underline{w}_0^{\rightarrow}] \leftarrow \text{wylicz wartości}(\mathcal{R}, \mathcal{R}_{II}) ;$ 
 $[\underline{\sigma}_0^{\leftarrow}, \underline{w}_0^{\leftarrow}] \leftarrow \text{skoryguj}(\underline{\sigma}_0^{\leftarrow}, \underline{w}_0^{\leftarrow}) ;$ 
 $\tilde{\underline{x}}_0^{\leftarrow}, \tilde{\underline{x}}_0^{\rightarrow} \leftarrow \text{wyznacz wektory}(\underline{c}_0^{\leftarrow}, \underline{c}_0^{\rightarrow}, \underline{\sigma}_0^{\leftarrow}, \underline{\sigma}_0^{\rightarrow}, \underline{w}_0^{\leftarrow}, \underline{w}_0^{\rightarrow}) ;$ 
 $\tilde{\underline{x}}_0^d = \tilde{\underline{x}}_0^{\rightarrow} - \tilde{\underline{x}}_0^{\leftarrow} ;$ 
 $Q^{(0)} = \emptyset ;$ 
 $Q^{(0)} = Q^{(0)} \cup \text{osobnik}(\mathcal{R}, \mathcal{R}_{II}) ;$ 
for  $m = 2$  to  $\mu$ 
     $\underline{r}_r \leftarrow \text{wektor losowy};$ 
     $\tilde{\underline{x}} = \tilde{\underline{x}}_0^{\leftarrow} + \underline{r}_r \circ \tilde{\underline{x}}_0^d ;$ 
     $Q^{(0)} = Q^{(0)} \cup \text{osobnik}(\tilde{\underline{x}}) ;$ 
end
 $Q^{(0)} \leftarrow \text{uczenie globalne}(Q^{(0)}, E_{NT}) ;$ 

```

gdzie  $\underline{c}_0^{\leftarrow}, \underline{c}_0^{\rightarrow}$  - wektory przedziałów zmienności środków funkcji gaussowskich w przesłankach, o postaci:

$$\begin{aligned} \underline{c}_0^{\leftarrow} &= [c_{0,1}^{\leftarrow}, \dots, c_{0,N-1_{II}}^{\leftarrow}]^T \\ \underline{c}_0^{\rightarrow} &= [c_{0,1}^{\rightarrow}, \dots, c_{0,N-1_{II}}^{\rightarrow}]^T, \end{aligned} \quad (5.39)$$

$$\begin{aligned} c_{0,i}^{\leftarrow} &= c_{0,i} - \sqrt{\frac{(\sigma_{0,i})^2}{-2 \cdot \ln(G_L)}} \\ c_{0,i}^{\rightarrow} &= c_{0,i} + \sqrt{\frac{(\sigma_{0,i})^2}{-2 \cdot \ln(G_L)}}, \end{aligned} \quad (5.40)$$

dla  $i \in \{1, 2, \dots, N \cdot I_{II}\}$ ,  $\underline{\sigma}_0^{\leftarrow}, \underline{\sigma}_0^{\rightarrow}$  - wektory przedziałów zmienności dyspersji funkcji gaussowskich w przesłankach, o następującej postaci:

$$\begin{aligned} \underline{\sigma}_0^{\leftarrow} &= [\sigma_{0,1}^{\leftarrow}, \dots, \sigma_{0,N \cdot I_{II}}^{\leftarrow}]^T \\ \underline{\sigma}_0^{\rightarrow} &= [\sigma_{0,1}^{\rightarrow}, \dots, \sigma_{0,N \cdot I_{II}}^{\rightarrow}]^T, \end{aligned} \quad (5.41)$$

$$\begin{aligned} \sigma_{0,i}^{\leftarrow} &= 0.5 \cdot \sigma_{0,i} \\ \sigma_{0,i}^{\rightarrow} &= 1.5 \cdot \sigma_{0,i}, \end{aligned} \quad (5.42)$$

dla  $i \in \{1, 2, \dots, N \cdot I_{II}\}$ ,  $\underline{w}_0^{\leftarrow}, \underline{w}_0^{\rightarrow}$  - wektory przedziałów zmienności długości nośników w konkluzjach reguł, o następującej postaci:

$$\begin{aligned} \underline{w}_0^{\leftarrow} &= [w_{0,1}^{\leftarrow}, \dots, w_{0,I_{II}}^{\leftarrow}]^T \\ \underline{w}_0^{\rightarrow} &= [w_{0,1}^{\rightarrow}, \dots, w_{0,I_{II}}^{\rightarrow}]^T, \end{aligned} \quad (5.43)$$

$$\begin{aligned} w_{0,i}^{\leftarrow} &= 0.25 \cdot w_{0,i} \\ w_{0,i}^{\rightarrow} &= 1.25 \cdot w_{0,i}, \end{aligned} \quad (5.44)$$

dla  $i \in \{1, 2, \dots, I_{II}\}$ ,  $\tilde{\underline{x}}_0^{\leftarrow}, \tilde{\underline{x}}_0^{\rightarrow}$  - złożone wektory przedziałów zmienności parametrów funkcji przynależności przesłanek i konkluzji dla całej bazy danych systemu rozmytego, w kolejności ich występowania w chromosomie. Wektory te mają następującą postać:

$$\begin{aligned} \tilde{\underline{x}}_0^{\leftarrow} &= [c_{0,1}^{\leftarrow}, \sigma_{0,1}^{\leftarrow}, \dots, c_{0,N \cdot 1}^{\leftarrow}, \sigma_{0,N \cdot 1}^{\leftarrow}, w_{0,1}^{\leftarrow}, c_{0,2}^{\leftarrow}, \dots, w_{0,I_{II}-1}^{\leftarrow}, c_{0,1 \cdot I_{II}}^{\leftarrow}, \sigma_{0,1 \cdot I_{II}}^{\leftarrow}, \dots, c_{0,N \cdot I_{II}}^{\leftarrow}, \sigma_{0,N \cdot I_{II}}^{\leftarrow}, w_{0,I_{II}}^{\leftarrow}]^T \\ \tilde{\underline{x}}_0^{\rightarrow} &= [c_{0,1}^{\rightarrow}, \sigma_{0,1}^{\rightarrow}, \dots, c_{0,N \cdot 1}^{\rightarrow}, \sigma_{0,N \cdot 1}^{\rightarrow}, w_{0,1}^{\rightarrow}, c_{0,2}^{\rightarrow}, \dots, w_{0,I_{II}-1}^{\rightarrow}, c_{0,1 \cdot I_{II}}^{\rightarrow}, \sigma_{0,1 \cdot I_{II}}^{\rightarrow}, \dots, c_{0,N \cdot I_{II}}^{\rightarrow}, \sigma_{0,N \cdot I_{II}}^{\rightarrow}, w_{0,I_{II}}^{\rightarrow}]^T, \end{aligned} \quad (5.45)$$

gdzie długość każdego z wektorów jest równa  $(2 \cdot N + 1) \cdot I_{II}$ ,  $\underline{r}_r$  - wektor wartości realizacji zmiennej losowej o rozkładzie równomiernym na przedziale  $[0, 1]$ :

$$\underline{r}_r = [\delta(0,1), \dots, \delta(0,1)], \quad (5.46)$$

gdzie długość wektora  $\underline{r}_r$  jest równa  $(2 \cdot N + 1) \cdot I_{II}$ ,  $\circ$  - operator iloczynu tablicowego na argumentach.

Funkcja **skoryguj** nie dopuszcza do powstania dyspersji w przesłance lub nośnika w konkluzji, o zerowej lub ujemnej długości. Funkcja **osobnik**( $\mathcal{R}, \mathcal{R}_{II}$ ) generuje osobnika o bazie wiedzy powstałej w wyniku działania pierwszego i drugiego etapu metody uczenia. Funkcja **wektor losowy** generuje wektor wartości losowych na przedziale  $[0, 1]$  o długości  $(2 \cdot N + 1) \cdot I_{II}$ .

## Próbkowanie

Podobnie jak w pierwszym etapie trójetapowej metody, wybór  $\lambda$  rodziców do operacji genetycznych spośród  $\mu$  kandydatów został zaimplementowany z wykorzystaniem próbkowania stochastycznego, z algorytmem identycznym jak opisany w pierwszym etapie metody.

## Operatory

W strategii ewolucyjnej zastosowano dwa operatory, realizowane w osobnych funkcjach operujących na całych pokoleniach. W pierwszej kolejności, na pokoleniu  $\mathcal{J}^{(k)}$  wykonywana jest operacja krzyżowania max-min-arytmetycznego, następnie osobniki powstałego w ten sposób pokolenia pośredniego  $\mathcal{J}_c^{(k)}$  są poddawane periodycznej mutacji nierównomiernej. Liczebność pokoleń  $\mathcal{J}^{(k)}$  oraz  $\mathcal{J}_c^{(k)}$  jest stała i wynosi  $\lambda$ . Poniżej przedstawiono algorytm wykonania operacji krzyżowania na pokoleniu  $\mathcal{J}^{(k)}$ :

```

 $V_L = \text{ones}(\lambda)$  ;
 $\mathcal{J}_c^{(k)} = \emptyset$  ;
 $\underline{f}_{\mathcal{J}_c}^{(k)} = \emptyset$  ;
while  $\sum_{i=1}^{\lambda} V_L(i) > 1$ 
     $i_{P1} = \left\lceil \delta \left( 0, \sum_{i=1}^{\lambda} V_L(i) \right) \right\rceil$  ;
     $j_{P1} \leftarrow \mathcal{J}^{(k)}[\text{index}(i_{P1})]$  ;
     $V_L[\text{index}(i_{P1})] = 0$  ;
     $i_{P2} = \left\lceil \delta \left( 0, \sum_{i=1}^{\lambda} V_L(i) \right) \right\rceil$  ;
     $j_{P2} \leftarrow \mathcal{J}^{(k)}[\text{index}(i_{P2})]$  ;
     $V_L[\text{index}(i_{P2})] = 0$  ;
    if  $\delta(0, \dots, 1) < P_{rc}$  then
         $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \text{dwa najlepsze osobniki z krzyżowania } (j_{P1}, j_{P2})$  ;
         $\underline{f}_{\mathcal{J}_c}^{(k)} \leftarrow \underline{f}_{\mathcal{J}_c}^{(k)} \cup \text{ocen}\acute{e}(j_{P1}, j_{P2})$  ;
    else
         $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup [j_{P1}, j_{P2}]$  ;
         $\underline{f}_{\mathcal{J}_c}^{(k)} \leftarrow \underline{f}_{\mathcal{J}_c}^{(k)} \cup [\underline{f}_{\mathcal{J}}^{(k)}(j_{P1}), \underline{f}_{\mathcal{J}}^{(k)}(j_{P2})]$  ;
    end
end
if  $\sum_{i=1}^{\lambda} V_L(i) = 1$  then
     $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \text{pozostały osobnik}(\mathcal{J}^{(k)})$  ;
     $\underline{f}_{\mathcal{J}_c}^{(k)} \leftarrow \underline{f}_{\mathcal{J}_c}^{(k)} \cup \underline{f}_{\mathcal{J}}^{(k)}(\text{powyższego osobnika})$  ;
end

```

gdzie  $V_L$  – wektor wyznaczający, które osobniki z pokolenia nie zostały jeszcze wybrane do krzyżowania (wartość dla tej pozycji wynosi 1), a które zostały już wybrane (wartość dla tej pozycji wynosi 0),  $i_{P1}$ ,  $i_{P2}$  – wskaźniki osobników krzyżowanych,  $j_{P1}$ ,  $j_{P2}$  – osobniki (wektory

binarne) podlegające krzyżowaniu,  $P_{rc}$  – prawdopodobieństwo krzyżowania, stanowiące parametr metody,  $\underline{f}_{\mathcal{J}}^{(k)}$  oraz  $\underline{f}_{\mathcal{J}_c}^{(k)}$  – wektory ocen osobników odpowiednio  $\mathcal{J}^{(k)}$  i  $\mathcal{J}_c^{(k)}$ . Zapis  $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \bullet$  oznacza operację dodania do populacji  $\mathcal{J}_c^{(k)}$  wskazanych osobników.  $\delta(0, \bullet)$  - realizacja zmiennej losowej o rozkładzie równomiernym na wskazanym przedziale.

Do pojedynczej operacji krzyżowania losowo wybiera się dwa osobniki rodzicielskie z puli  $\mathcal{J}^{(k)}$ , bez zwracania. Wybór następuje z jednakowym prawdopodobieństwem – na tym etapie nie jest uwzględniana ranga osobników. Wybrana para osobników podlega krzyżowaniu max-min-arytmetycznemu z prawdopodobieństwem  $p_{rc}$ , które stanowi parametr metody. Osobniki potomne (lub jeśli para nie podlegała krzyżowaniu, osobniki rodzicielskie) są dołączane do pokolenia pośredniego  $\mathcal{J}_c^{(k)}$ . Jeśli liczba osobników  $\lambda$  jest nieparzysta, pozostały osobnik z pokolenia  $\mathcal{J}^{(k)}$  jest dołączany do pokolenia  $\mathcal{J}_c^{(k)}$  bez zmian, wraz z równoległym przepisaniem z  $\underline{f}_{\mathcal{J}}$  do  $\underline{f}_{\mathcal{J}_c}$  jego oceny. W procesie krzyżowania max-min-arytmetycznego, z dwóch osobników rodzicielskich powstają cztery osobniki potomne, z których wybierane są dwa, o największej wartości funkcji przystosowania. Parametry konkluzji są każdorazowo ustalane z wykorzystaniem uczenia globalnego na zbiorze  $E_{NT}$ . W przypadku, gdy operacja krzyżowania nie jest wykonywana, oceny osobników rodzicielskich są przepisywane z  $\underline{f}_{\mathcal{J}}$  do  $\underline{f}_{\mathcal{J}_c}$  bez ponownego przeliczania wartości funkcji przystosowania.

Operacja krzyżowania dwóch osobników jest przeprowadzana następująco:

```

 $C_{p1} \leftarrow \mathcal{J}^{(k)}(i_{p1}) ;$ 
 $C_{p2} \leftarrow \mathcal{J}^{(k)}(i_{p2}) ;$ 
 $a = \delta(0, 1) ;$ 
 $C_{c1}^{(l)} = a \cdot C_{p1}^{(l)} + (1-a) \cdot C_{p2}^{(l)} ; \text{ dla } l=1, 2, \dots, \mathcal{L}$ 
 $C_{c2}^{(l)} = a \cdot C_{p2}^{(l)} + (1-a) \cdot C_{p1}^{(l)} ; \text{ dla } l=1, 2, \dots, \mathcal{L}$ 
 $C_{c3}^{(l)} = \min(C_{p1}^{(l)}, C_{p2}^{(l)}) ; \text{ dla } l=1, 2, \dots, \mathcal{L}$ 
 $C_{c4}^{(l)} = \max(C_{p1}^{(l)}, C_{p2}^{(l)}) ; \text{ dla } l=1, 2, \dots, \mathcal{L}$ 
 $\mathcal{J}_4 \leftarrow \text{uczenie\_globalne}(C_{c1}, C_{c2}, C_{c3}, C_{c4}, E_{NT}) ;$ 
 $v_4 \leftarrow \text{ocen}(C_{c1}, C_{c2}, C_{c3}, C_{c4}, E_{NT}) ;$ 
wybierz 2 najlepsze osobniki (oceny wg  $v_4$ ) ;

```

gdzie  $C_{p1}$ ,  $C_{p2}$  – chromosomy osobników rodzicielskich,  $C_{p1}^{(l)}$ ,  $C_{p2}^{(l)}$  –  $l$ -ta pozycja w ciągu kodowym chromosomów rodzicielskich,  $\mathcal{J}_4$  - pomocnicze pokolenie osobników potomnych, zawierające cztery osobniki uzyskane w wyniku krzyżowania max-min-arytmetycznego,  $v_4$  – wektor ocen osobników należących do  $\mathcal{J}_4$ . Powyższy algorytm jest podobny do algorytmu zaprezentowanego w podrozdziale 5.1.2, przy czym odmienna jest funkcja oceniająca, a w miejsce uczenia lokalnego parametrów konkluzji reguł zastosowano uczenie globalne.

Kolejną wykonywaną operacją jest periodyczna mutacja nierównomierna, realizowana zgodnie z zamieszczonym poniżej algorytmem:

```

for i=1 to  $\lambda$ 
   $C_M \leftarrow \mathcal{J}_c^{(k)}[i]$  ;
  if  $\delta(0, \dots, 1) < p_{rM}$  then
     $l = \lceil \delta(0, \mathcal{L}) \rceil$  ;
     $b = \delta\{0, 1\}$  ;
    if  $b=0$  then
       $C_M^{(l)} = C_M^{(l)} + \Delta(k, \tilde{x}_{0,l}^d)$  ;
    else
       $C_M^{(l)} = C_M^{(l)} - \Delta(k, \tilde{x}_{0,l}^d)$  ;
    end
    uczenie_globalne ( $C_M$ ) ;
     $f_{\mathcal{J}_c}^{(k)} \leftarrow f_{\mathcal{J}_c}^{(k)} \cup \text{ocień} (C_M)$  ;
  else
     $f_{\mathcal{J}_c}^{(k)} \leftarrow f_{\mathcal{J}_c}^{(k)} \cup f_{\mathcal{J}}^{(k)}(C_M)$  ;
  end
   $\mathcal{J}_c^{(k)} \leftarrow \mathcal{J}_c^{(k)} \cup \text{osobnik}(C_M)$  ;
end
 $O^{(k)} \leftarrow \mathcal{J}^{(k)}$  ;

```

gdzie  $p_{rM}$  – prawdopodobieństwo mutacji, stanowiące parametr metody,  $C_M^{(l)}$  –  $l$ -ty gen chromosomu, reprezentujący parametr funkcji przynależności przesłanki lub konkluzji,  $\tilde{x}_{0,l}^d$  – przedział zmienności dla  $l$ -tego genu, wyliczony na etapie generowania pokolenia startowego, funkcja **osobnik**(•) tworzy nowego osobnika na podstawie argumentu, funkcja  $\Delta(k, \tilde{x}_{0,l}^d)$  wylicza zmianę wartości parametru reprezentowanego przez osobnika, periodycznie zawężając zakres zmian wraz ze wzrostem numeru pokolenia:

$$\Delta(k, d) = d \cdot \left( 1 - \delta[0,1] \left( 1 - \frac{\text{mod}(l,10)}{\mathcal{L}} \right)^b \right), \quad (5.47)$$

gdzie  $k$  – numer pokolenia,  $d$  – przedział zmienności wartości,  $b$  – parametr mutacji,  $b > 0$ ,  $\delta[0,1]$  – realizacja zmiennej losowej o rozkładzie równomiernym na przedziale  $[0,1]$ ,  $\text{mod}(\bullet, \bullet)$  – operacja dzielenia modulo. Jego zastosowanie znacznie poprawiło zbieżność w ostatniej fazie działania strategii ewolucyjnej, ponieważ zapewniło okresowe wprowadzenie różnorodności do populacji zdominowanej przez osobniki bliskie osobnikowi najlepszemu na wskutek zastosowanej selekcji elitarniej. Wartość 10 znajdująca się w wykładniku funkcji została dobrana szacunkowo, na podstawie serii eksperymentów.

### Selekcja elitarna

Nowe pokolenie bazowe  $Q^{(k+1)}$  powstaje poprzez wybór  $\mu$  osobników z połączenia starego pokolenia bazowego oraz pokolenia potomnego:  $Q^{(k)} \cup O^{(k)}$ . Wektor powstały z połączenia ocen

osobników:  $\underline{f}$  i  $\underline{f}_O$  jest sortowany rosnąco. Następnie wybiera się  $\mu$  najlepszych osobników z połączonych zbiorów  $Q^{(k)} \cup O^{(k)}$ , wg indeksów wynikających z posortowanych ocen.

### Funkcja oceniająca przystosowanie

Funkcja oceniająca przystosowanie osobnika jest miarą jakości systemu rozmytego reprezentowanego przez osobnika:

$$f(\tilde{x}, E_{NT}) = \frac{1}{\sqrt[4]{f_J(\tilde{x}, E_{NT})}}, \quad (5.48)$$

gdzie  $f_J$  – funkcja kary, zdefiniowana następująco:

$$f_J(\tilde{x}, E_{NT}) = \frac{1}{2 \cdot N_T} \sum_{n_T=1}^{N_T} (t_0(n_T) - y_0(n_T, E_{NT}(n_T)))^2, \quad (5.49)$$

gdzie  $y_0(n_T, E_{NT}(n_T))$  - wartość wyjściowa systemu rozmytego o bazie wiedzy reprezentowanej przez osobnika  $\tilde{x}$ , dla  $n_T$ -tego rekordu ze zbioru uczącego  $E_{NT}$ .

### Podsumowanie

W ramach przeprowadzonych eksperymentów (przy założonych parametrach metody) strategia ewolucyjna wykorzystywana w trzecim etapie trójetapowej metody uczenia była uruchamiana dwu lub trzykrotnie, przy czym w drugim i trzecim uruchomieniu, w miejsce parametrów funkcji przynależności przesłanek i nośnika konkluzji osobnika dokoła którego tworzone jest pokolenie startowe wprowadzono parametry najlepszego osobnika uzyskanego w poprzednim (odpowiednio pierwszym lub drugim) uruchomieniu strategii. W ten sposób kolejne uruchomienia strategii ewolucyjnej powodowały kontynuację procesu lokalnego dostrajania. Przeprowadzone eksperymenty wskazują, iż trzecie uruchomienie strategii wprowadza jedynie nieznaczną poprawkę w jakości systemu (mierzonej funkcją przystosowania) i jedynie w początkowym okresie działania strategii, dla niskiej wartości licznika pokoleń. Ostatecznie wynikiem zarówno trzeciego etapu, jak i całej trójetapowej metody generowania systemu rozmytego z parametryzowanymi konkluzjami jest baza wiedzy osobnika o największej wartości funkcji przystosowania, wybranego z ostatniego pokolenia  $Q^{(k+1)}$  ostatniego uruchomienia strategii ewolucyjnej.





## 6

# Wyniki przeprowadzonych badań

Do przeprowadzenia badań wykorzystano środowisko MATLAB w wersji 6.5 (R13). W celu sprawdzenia słuszności postawionej tezy stworzono w tym środowisku dwa moduły narzędziowe (ang. *toolbox*). Pierwszym modułem implementuje system rozmyty z parametryzowanymi konkluzjami o gaussowskich funkcjach przynależności w przesłankach i trójkątnych funkcjach przynależności w konkluzji, zgodnie z założeniami przedstawionymi w podrozdziale 2.3.1. Drugi moduł zawiera całość związaną z obliczeniami ewolucyjnymi oraz algorytm pokrywania danych i implementuje trójetapową metodę uczenia, zgodnie z założeniami przedstawionymi w rozdziale 5. W celu dywersyfikacji obliczeń pomiędzy różne jednostki obliczeniowe założono, iż każdy z etapów trójetapowej metody uczenia może być wykonywany niezależnie (oczywiście w naturalnym porządku). Pozwoliło to na niezależne i równoległe wykonywanie poszczególnych etapów dla wielu zestawów danych i różnych parametrów obliczeń oraz bardziej elastyczne wykorzystywanie zasobów sprzętowych. Moduł obliczeń ewolucyjnych bezpośrednio wykorzystuje funkcjonalność modułu implementującego system rozmyty. Implementacja obu modułów narzędziowych została wykonana z wykorzystaniem elementów programowania obiektowego, dostępnych w środowisku MATLAB [117] oraz w wersji prostej, funkcjonalnie równoważnej wersji obiektowej, zoptymalizowanej dla masowych obliczeń.

## 6.1 Implementacja środowiska testowego

W kolejnych podrozdziałach przedstawiono szczegóły implementacyjne związane z dwoma modułami, odpowiednio modułem systemu rozmytego z parametryzowanymi konkluzjami oraz modułem trójetapowej metody uczenia na podstawie danych numerycznych. Korzystając z notacji UML [18], [44], [67], [124] zaprezentowano poszczególne klasy dla wersji systemu wykorzystującej elementy programowania obiektowego, dostępne w środowisku MATLAB. Klasy te tworzą środowisko testowe, wykorzystywane w celu sprawdzenia słuszności postawionej w niniejszej rozprawie tezy.

### 6.1.1 Moduł systemu rozmytego z parametryzowanymi konkluzjami

Implementacja modułu narzędziowego została przeprowadzona trójpoziomowo.

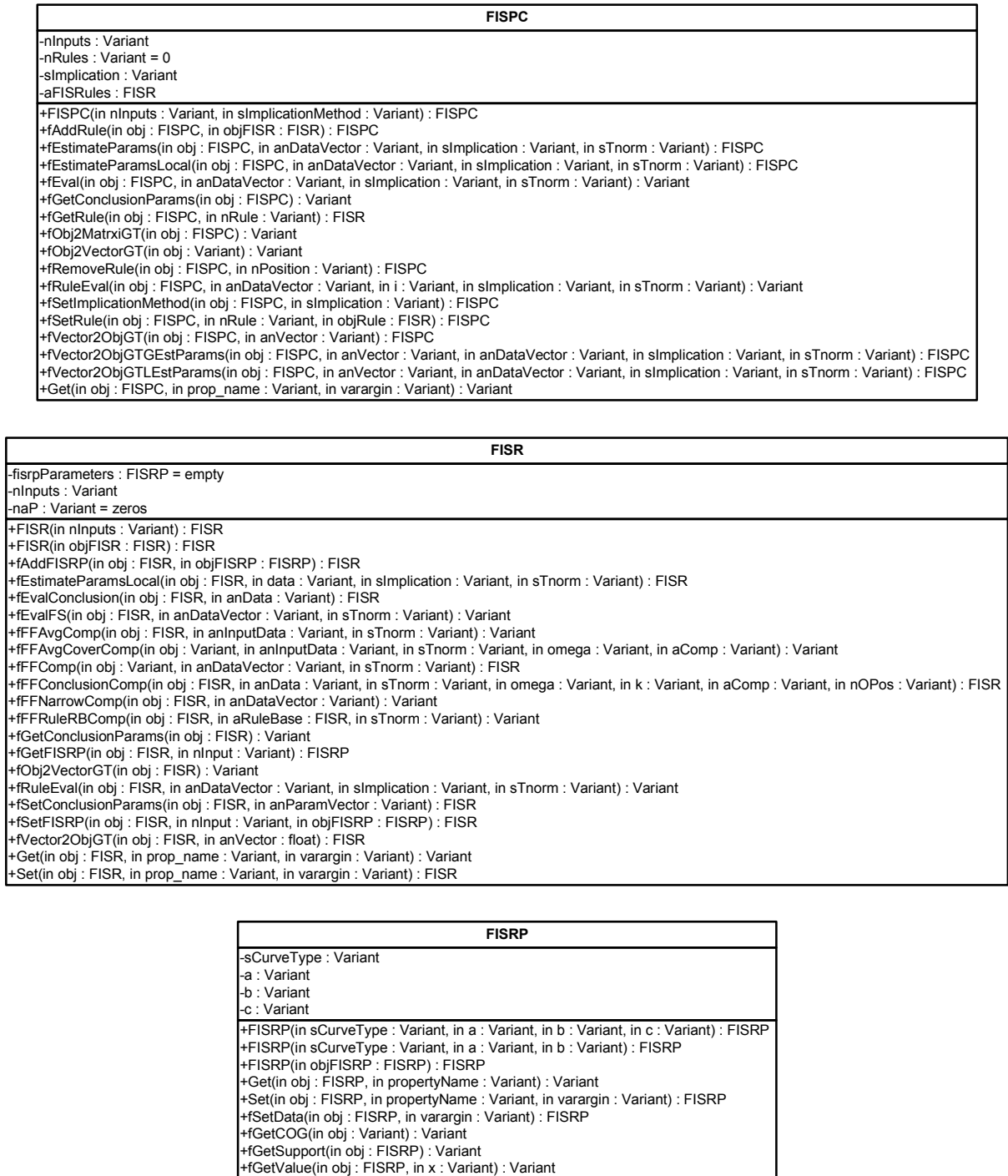
Na najniższym poziomie znajduje się klasa FISRP implementująca funkcjonalność pojedynczego zbioru rozmytego. Obiekty tej klasy przechowują typ i parametry funkcji

przynależności, niezależnie od tego, czy jest to funkcja gaussowska w przesłance, czy trójkątna w konkluzji. Udostępniają m.in. funkcjonalność wyliczenia wartości funkcji przynależności zbioru rozmytego dla zadanego argumentu.

Na wyższym poziomie znajdują się klasa FISR, implementująca pojedynczą regułę rozmytą **if-then** z parametryzowaną konkluzją, o logicznej interpretacji. W obiekcie tej klasy można wykorzystać jedną z następujących implikacji: Łukasiewicza, Fodora, Reichenbacha, Kleene-Dienes, Zadeha, Goguena, Gödela lub Reschera oraz jedną z  $t$ -norm w wielowartościowej przesłance: minimum lub iloczyn algebraiczny. Pojedyncza reguła przechowuje kolekcję obiektów-parametrów FISRP opisujących parametry funkcji przynależności przesłanek oraz funkcji przynależności konkluzji, jak również zawiera wektor parametrów konkluzji, wykorzystywany do ustalenia położenia jej środka ciężkości. Funkcjonalność reguły obejmuje m.in. ustalenie tych parametrów metodą uczenia lokalnego z wykorzystaniem ważonego estymatora najmniejszych kwadratów, wg założeń opisanych w podrozdziale 2.3.1 oraz zawiera funkcje wykorzystywane przez moduł obliczeń ewolucyjnych (w pierwszym etapie trójetapowej metody uczenia), takie jak np. wyliczenie kompatybilności reguły i wskazanego zbioru danych. Dodatkową funkcjonalność ułatwiającą wykorzystywanie klasy przez moduł obliczeń ewolucyjnych stanowią funkcje konwersji obiektu do postaci wektora parametrów funkcji przynależności (Rysunek 5.2) oraz w kierunku odwrotnym, z postaci wektora do formy obiektu klasy FISR. Oczywiście stworzenie obiektu na podstawie parametrów funkcji przynależności wymaga wyliczenia parametrów opisujących położenie konkluzji metodą uczenia lokalnego.

Na najwyższym poziomie znajduje się klasa FISPC, implementująca funkcjonalność pełnego systemu rozmytego z parametryzowanymi konkluzjami o modelu MISO. Klasa ta zawiera kolekcję obiektów-reguł FISR i udostępnia zarówno podstawowe funkcje, takie jak wyliczenie wartości wyjściowej dla zadanych argumentów wejściowych, jak i ustalenie parametrów opisujących położenie środków ciężkości konkluzji za pomocą uczenia globalnego, z wykorzystaniem metody najmniejszych kwadratów oraz dodatkowe wsparcie dla modułu obliczeń ewolucyjnych, wykorzystywane przez drugi i trzeci etap trójetapowej metody uczenia. Również w przypadku tej klasy zaimplementowano funkcje konwersji obiektu do postaci wektora (Rysunek 5.6) oraz dodatkowo do postaci macierzy, ułatwiającej interpretację poszczególnych elementów opisujących funkcje przynależności. Dostępne są również funkcje wykonujące operację odwrotną, analogicznie jak to ma miejsce w przypadku klasy FISR, przy czym do wyznaczenia parametrów opisujących położenie konkluzji wykorzystano metodę uczenia globalnego.

Na poniższym rysunku przedstawiono diagram klas UML modułu systemu rozmytego z parametryzowanymi konkluzjami:



Rysunek 6.1: Diagram klas UML modułu implementującego system rozmyty z parametryzowanymi konkluzjami.

### 6.1.2 Implementacja modułu obliczeń ewolucyjnych

Moduł obliczeń ewolucyjnych składa się z trzech niezależnych klas, z których każda odpowiada funkcjonalnie jednemu etapowi trójetapowej metody uczenia.

Klasa **EA2FISCCover** implementuje algorytm iteracyjnego pokrywania bazy wiedzy, przechowując wszystkie parametry związane z obliczeniami ewolucyjnymi i metodą pokrywającą, jak również przechowuje pełny uczący zbiór danych, bieżący niepokryty zbiór danych uczących, dotychczas zidentyfikowaną bazę wiedzy w formie kolekcji obiektów FISR. Elementy związane ze strategią ewolucyjną odnajdującą kolejną regułę, takie jak pokolenia bazowe, potomne i tymczasowe oraz odpowiadające im tablice ocen stanowią zmienne lokalne głównej metody odnajdującej kolejną regułę. Poszczególne elementy strategii ewolucyjnej, takie jak generowanie pokolenia startowego, krzyżowanie i mutacja są zaimplementowane w oddzielnych metodach klasy. Po odnalezieniu reguły, jej parametry funkcji przynależności przesłanek i szerokość nośnika w konkluzji oraz aktualna migawka (ang. *snapshot*) zbioru niepokrytych rekordów są zapisywane na dysku we wskazanym parametrze folderze. Dzięki takiemu rozwiązaniu istnieje możliwość przerwania i późniejszej kontynuacji obliczeń, jak również pozwala to na ocenę bieżącego zaawansowania algorytmu pokrywania. Dodatkową, cenną informację niosą czasy generowań poszczególnych reguł. Główna metoda implementująca pętlę strategii ewolucyjnej odpowiada również za zbieranie i wyświetlanie graficznej prezentacji procesu ewolucji, który zawiera informacje o aktualnie wykonywanym etapie obliczeń (ocena, krzyżowanie, mutacja), wartość funkcji przystosowania osobnika najlepszego oraz średnią wartość funkcji przystosowania osobników w pokoleniu bazowym, jak również informuje o liczbie odnalezionych uprzednio reguł.

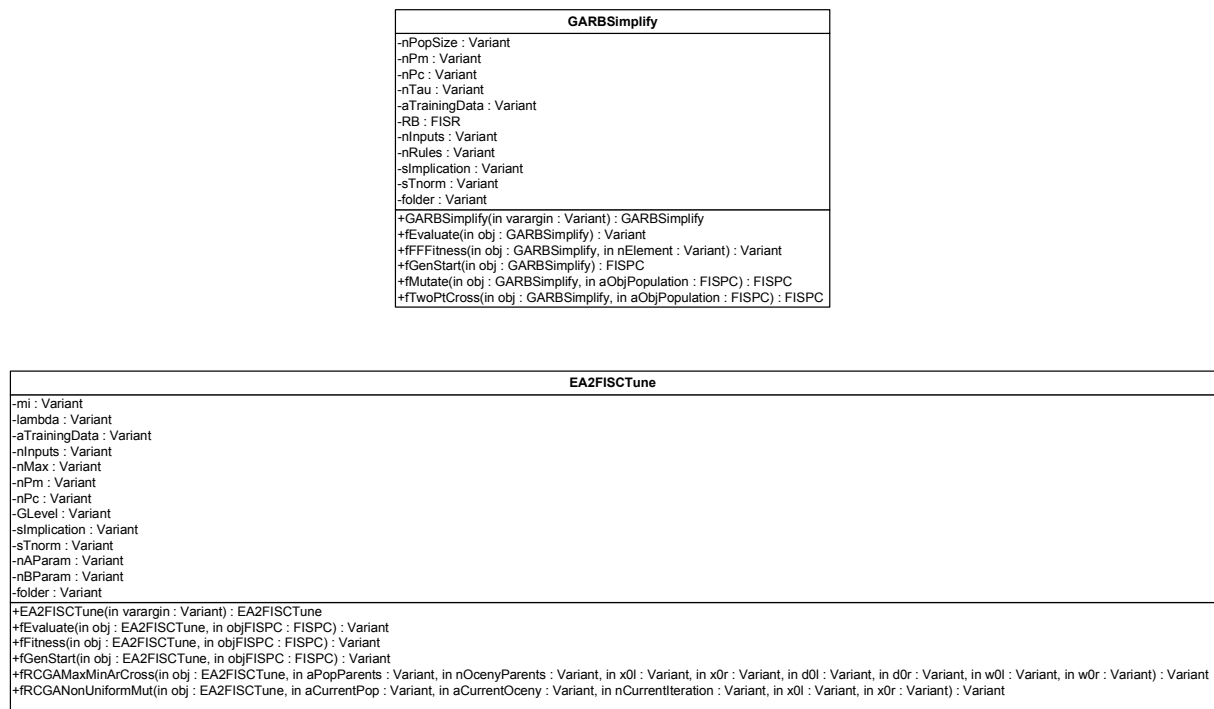
Klasa **GARBSimplify** implementuje algorytm genetyczny odnajdujący optymalną bazę reguł, co odpowiada drugiemu krokowi trójetapowej metody uczenia. Obiekt przechowuje jedynie parametry sterujące algorytmem genetycznym oraz wartości inicjujące, w szczególności zbiór uczący i folder wyników. Wszystkie elementy związane z obliczeniami genetycznymi są przechowywane w postaci zmiennych lokalnych głównej metody implementującej pętlę algorytmu genetycznego. Poszczególne elementy algorytmu genetycznego, takie jak generowanie pokolenia startowego, krzyżowanie i mutacja są zaimplementowane w oddzielnych metodach klasy. Algorytm korzysta bezpośrednio z obiektów klasy FISPC celem utworzenia na etapie obliczeń genetycznych systemów rozmytych o zadanej bazie reguł. Ostateczny wynik jest zapisywany w pliku w określonym parametrze folderze. Żadne wyniki pośrednie nie są kolekcjonowane w postaci plików dyskowych, natomiast podobnie do klasy **EA2FISCCover**, generowany jest wykres, informujący o najlepszym bieżącym osobniku, średnim przystosowaniu populacji bazowej oraz o bieżącym etapie obliczeń. Za te operacje jest odpowiedzialna metoda implementująca główną pętlę algorytmu genetycznego.

Klasa **EA2FISCTune** implementuje operacje lokalnego dostrajania parametrów systemu rozmytego klasy FISPC, w sposób zgodny z opisem przedstawionym w trzecim etapie trójetapowej metody uczenia. Klasa ta, podobnie do **EA2FISCCover**, przechowuje w swoich atrybutach wyłącznie parametry pracy sterujące procesem dostrajania, natomiast elementy strategii ewolucyjnej (w szczególności kolekcje osobników tworzących pokolenia oraz oceny) stanowią zmienne lokalne metody implementującej główną pętlę strategii. Operacje związane ze strategią ewolucyjną, takie jak generowanie pokolenia startowego, mutacja i krzyżowanie są

zawarte w osobnych metodach. Najlepszy osobnik po każdym 10 krokach strategii ewolucyjnej jest zapisywany na dysku we wskazanym folderze. Po zakończeniu obliczeń jest w nim również zapisywany osobnik najlepszy w całym toku ewolucji. Pozwala to na monitorowanie bieżących postępów w pracy strategii ewolucyjnej, jak również daje możliwość wybrania osobnika wygenerowanego w jednym z pośrednich korków strategii. Jest to szczególnie użyteczne w przypadku systemów, które wykazują tendencję do „przeuczania się”. Plik zawiera wektor parametrów funkcji przynależności poszczególnych reguł, ułożone jedno za drugim. Format pojedynczej reguły obejmuje parametry funkcji przynależności przesłanek oraz szerokość nośnika w konkluzji, tak jak to ma miejsce w przypadku plików generowanych przez klasę **EA2FISCCover**, natomiast nie obejmuje parametrów kombinacji liniowej, wykorzystywanych do ustalenia położenia środka ciężkości konkluzji, które to parametry są każdorazowo douczane, gdy jest to niezbędne. W tym celu wykorzystywana jest zaimplementowana w obiekcie **FISPC** metoda uczenia globalnego. Główna metoda implementująca pętlę strategii ewolucyjnej odpowiada również za zbieranie i wyświetlanie graficznej prezentacji procesu ewolucji, który zawiera informacje o aktualnie wykonywanym etapie obliczeń (ocena, krzyżowanie, mutacja), wartości funkcji przystosowania osobnika najlepszego oraz średniej wartości funkcji przystosowania osobników w pokoleniu bazowym, w kolejnych iteracjach algorytmu.

Poniżej przedstawiono diagram klas UML prezentujący szczegóły implementacji klas realizujących funkcjonalność trójetapowej metody uczenia:

EA2FISCCover
-mi : Variant = 61 -lambda : Variant = 61 -aTrainingData : Variant -aT : Variant -nInputs : Variant -nMax : Variant = 50 -kParam : Variant = 0.1 -omegaParam : Variant = 0.05 -minCVPParam : Variant = 1.5 -crossoverParam : Variant = 0.35 -bParams : Variant -GL : Variant -Pc : Variant = 0.6 -Pm : Variant = 0.1 -RB : FISR -folder : Variant
+EA2FISCCover(in varargin : Variant) : EA2FISCCover +FEvaluate(in obj : EA2FISCCover, in sImplicationMethod : Variant, in sTnorm : Variant) : FISPC +FFFFitness(in obj : EA2FISCCover, in rule : FISR, in anCurrentRB : FISR, in anDataVector : Variant, in sTnorm : Variant, in sImplication : Variant) : Variant +FFindNextRule(in obj : EA2FISCCover, in sTnorm : Variant, in sImplication : Variant) : Variant +FGenStartRules(in obj : EA2FISCCover) : Variant +FRBCoverage(in obj : EA2FISCCover, in aRule : FISR, in anData : Variant, in sTnorm : Variant, in sImplication : Variant) : Variant +RCGAMaxMinArCross(in obj : EA2FISCCover, in aObjPopulation : EA2FISCCover, in nAParam : Variant, in RB : FISR, in sTnorm : Variant, in omega : Variant, in sImplication : Variant) : Variant +RCGANonUniformMut(in obj : EA2FISCCover, in aObjPopulation : Variant, in nBParam : Variant, in nIteration : Variant, in nMax : Variant, in anLeftDI : Variant, in anRightDI : Variant) : Variant



Rysunek 6.2: Diagram klas UML modułu implementującego poszczególne etapy trójetapowej metody uczenia.

### 6.1.3 Implementacja - podsumowanie

Implementacja z wykorzystaniem elementów programowania obiektowego pozwala na przejrzyste zaprezentowanie funkcji poszczególnych modułów, znacznie też upraszcza prace programistyczne. Środowisko MATLAB nie oferuje zaawansowanych metod programowania obiektowego, jak ma to miejsce w przypadku języków C++ czy Java, ale elementy pozwalające na definiowanie klas i ich metod okazały się wystarczające w rozważanym zastosowaniu. Natomiast niewątpliwym negatywnym aspektem przemawiającym na korzyść rozwiązań nie wykorzystujących elementów programowania obiektowego w środowisku MATLAB jest znaczne spowolnienie obliczeń wersji obiektowej, w stosunku do wersji operującej na wielowymiarowych macierzach. Wersja operująca na macierzach została opracowana zgodnie z równaniami znajdującymi się w rozdziałach 2.3.1 i 5, oraz zgodnie z algorytmami przedstawionymi w rozdziale 5. Seria testów eksperymentalnych dowiodła ponad dwudziestokrotną różnicę na niekorzyść rozwiązania obiektowego dla pierwszego etapu trójetapowej metody pokrywania oraz ponad pięćdziesięcioczkrotną na niekorzyść rozwiązania obiektowego w etapie trzecim. Wartości te były mierzone dla systemu o dzieściu wejściach oraz zbiorze uczącym o 290 rekordach. Dodatkowym problemem podczas wykonywanych obliczeń okazał się brak wsparcia ze strony środowiska MATLAB dla wieloprocesorowej platformy sprzętowej klasy PC, jak również dla procesorów dwupotokowych HT (ang. Hyper Threading). Szczegółowe czasy obliczeń w zależności od wykorzystywanej

platformy sprzętowej oraz rozważanego zagadnienia znajdują się w końcowej części niniejszego rozdziału.

## 6.2 Przeprowadzone badania

W celu zbadania prawdziwości postawionej w rozprawie tezy przeprowadzono badania, obejmujące zagadnienia identyfikacji pieca gazowego Boxa-Jenkinsa, predykcji liczby plam na słońcu i predykcji chaotycznych szeregów czasowych Mackeya-Glassa. W kolejnych podrozdziałach zaprezentowano uzyskane wyniki oraz podano zestawy parametrów sterujących przebiegiem trójetapowej metody uczenia. Jako kryterium jakości przyjęto wskaźnik błędu średniokwadratowego  $J_{RMSE}$  lub błędu średniego  $J_{AVE}$ , zdefiniowane następująco:

$$J_{RMSE} = \sqrt{\frac{1}{N_T} \sum_{n_T=1}^{N_T} (t_0(n_T) - y_0(n_T, E_{NT}(n_T)))^2}, \quad (6.1)$$

$$J_{AVE} = \frac{1}{N_T \cdot (t_0^{\rightarrow} - t_0^{\leftarrow})} \sum_{n_T=1}^{N_T} abs(t_0(n_T) - y_0(n_T, E_{NT}(n_T))), \quad (6.2)$$

w zależności od postaci wyników porównywanych metod zawartych w literaturze.

### 6.2.1 Identyfikacja pieca gazowego (Box-Jenkins)

Przedstawiony w niniejszej rozprawie model systemu ewolucyjno-rozmytego z parametryzowanymi konkluzjami można zastosować w zagadnieniu identyfikacji systemów. Jako klasyczny i dobrze udokumentowany przykład wybrano zagadnienie identyfikacji przemysłowego pieca gazowego. W tym celu wykorzystano bazę danych G. E. P. Boxa i G. M. Jenkinsa [20], zawierającą 290 rekordów danych o postaci:

$$[x_n, y_n] = [y(n-1), \dots, y(n-4), x(n-1), \dots, x(n-6), y(n)]. \quad (6.3)$$

Sygnałem wejściowym  $x(n)$  jest prędkość przepływu metanu, dostarczanego do pieca wraz z powietrzem. Sygnałem wyjściowym  $y(n)$  jest procentowe stężenie dwutlenku węgla w mieszaninie gazów spalinowych. Okres próbkowania obu wielkości wynosi 9s.

Identyfikację przeprowadzono dla implikacji Łukasiewicza, Fodora, Reichenbacha, Kleene-Dienes, Zadeha, Goguena, Gödela i Reschera oraz dla operatorów  $t$ -normy minimum i iloczynu algebraicznego.

Badania objęły dwa zakresy danych. W pierwszym przypadku badano zdolność aproksymacji danych pomiarowych skomplikowanego procesu przemysłowego. Dla tego przypadku zastosowano uczenie systemu z wykorzystaniem całej bazy danych Boxa\_Jenkinsa. W drugim przypadku badano zdolności uogólniania systemu. W tym celu zbiór uczący został utworzony z pierwszych stu rekordów bazy danych Boxa-Jenkinsa. Pozostałe 190 rekordów stanowiło zbiór testowy. Podział ten jest zgodny z podziałem stosowanym w publikacjach [111], [112] i pozwala na porównanie uzyskanych wyników z wynikami uzyskiwanymi innymi metodami.

W obu przypadkach jako kryterium jakościowe wykorzystywano wartość błędu średniokwadratowego ( $J_{RMSE}$ ) dla wskazanego zbioru danych.

### Wyniki badania jakości aproksymacji danych pomiarowych

Badanie zdolności do aproksymacji danych pomiarowych i precyzyjnego odwzorowania zachowania rzeczywistych, skomplikowanych procesów przemysłowych zostało przeprowadzone z wykorzystaniem całej bazy danych Boxa-Jenkinsa, zawierającej łącznie 290 rekordów. Należy zaznaczyć, że test ten stanowi optymistyczne oszacowanie jakości identyfikacji pieca gazowego, bowiem nie uwzględnia on uogólniania wiedzy na przypadki wcześniej systemowi nieznanne.

W celu wygenerowania systemu rozmytego zastosowano trójetapową metodę uczenia, w której ostatni, trzeci etap uczenia był wykonywany trzykrotnie, przy czym w drugim i trzecim uruchomieniu procesu dostrajania (trzeciego etapu trójetapowej metody uczenia) wykorzystywano najlepszy system rozmyty uzyskany w poprzedzającym uruchomieniu procesu dostrajania. Najlepszy system rozmyty poddawany ocenie stanowił osobnik o największej funkcji przystosowania, uzyskany w trzeciej iteracji trzeciego etapu trójetapowej metody uczenia (w procesie dostrajania parametrów).

Poniżej zestawiono zbiór parametrów sterujących wykonaniem poszczególnych etapów.

Tabela 6.1: Parametry sterujące trójetapowej metody uczenia dla zagadnienia identyfikacji pieca gazowego. Uczenie dla całej bazy danych Boxa-Jenkinsa.

Etap	Parametr	Symbol	Wartość
1.	Liczebność pokolenia bazowego.	$\mu$	60
	Liczebność pokolenia potomnego .	$\lambda$	20
	Liczba pokoleń.	$\mathcal{L}$	50
	Prawdopodobieństwo krzyżowania.	$P_c$	1
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.8
	Współczynnik rekordów pokrywanych przez regułę do rekordów niepokrywanych (zastosowany w funkcji oceny osobnika).	$\kappa$	0.05
	Minimalny stopień kompatybilności danych z regułą (zastosowany w funkcji oceny osobnika)..	$\omega$	0.1
	Minimalny stopień pokrycia rekordu przez reguły. Różne wartości parametru pozwoliły na uzyskanie różnej liczebności reguł (zastosowany na poziomie algorytmu pokrywającego).	$\varepsilon$	0.9 1.0 1.2 1.5 1.8
	Parametr krzyżowania uśredniającego.	$a$	0.35
	Parametr periodycznej mutacji nierównomiernej.	$b$	5
	Poziom funkcji gaussowskich dla wyliczenia szerokości nośnika w funkcjach przynależności przesłanek reguł.	$G_L$	0.25
2.	Liczebność pokolenia bazowego.	$M$	50
	Liczba pokoleń.	$K$	150
	Prawdopodobieństwo mutacji (dla genu)	$P_m$	0.005
	Prawdopodobieństwo krzyżowania.	$P_c$	0.75



	Minimalny wskaźnik pokrycia danych przez bazę reguł (zastosowany w funkcji oceny osobnika).	$\tau$	0.25
3.	Liczebność pokolenia bazowego.	$\mu$	100
	Liczebność pokolenia potomnego.	$\lambda$	20
	Liczba pokoleń dla pojedynczego uruchomienia strategii ewolucyjnej.	$\mathcal{L}$	150
	Prawdopodobieństwo krzyżowania (dla pary osobników)	$P_c$	0.6
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.6
	Poziom funkcji gaussowskich dla wyliczenia nośnika.	$G_L$	0.9
	Parametr krzyżowania uśredniającego.	$a$	0.5
	Parametr periodycznej mutacji nierównomiernej.	$b$	5

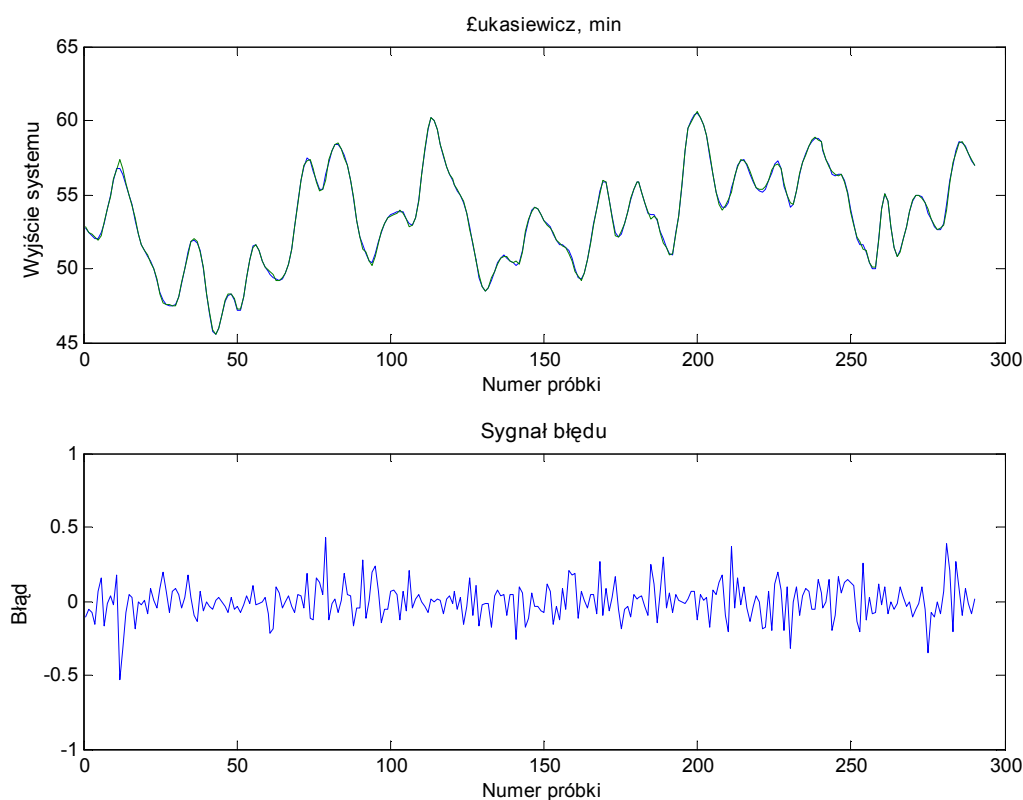
W procesie ustalania parametrów konkluzji metodą uczenia globalnego dla źle uwarunkowanych macierzy ( $D^T D$ ) (równanie 2.115) zastosowano na ich przekątnej głównej korektę równą  $1.15 \cdot 10^{-8}$ . Najniższy wskaźnik błędu średniokwadratowego dla przypadku aproksymacji danych uczących uzyskano dla systemu rozmytego z parametryzowanymi konkluzjami o bazie wiedzy składającej się z 9 rozmytych reguł warunkowych **if-then**, przy zastosowaniu logicznej interpretacji reguł i implikacji Łukasiewicza oraz  $t$ -normy minimum. Szczegółowe zestawienie parametrów funkcji przynależności przesłanek, szerokości nośnika konkluzji oraz parametrów opisujących kombinację liniową określającą położenie środków ciężkości konkluzji w regułach znajduje się w Dodatku A niniejszej rozprawy. W poniższej tabeli zestawiono średni i najlepszy rezultat oraz wyniki dla innych metod.

Tabela 6.2: Porównanie uzyskanych rezultatów dla zagadnienia identyfikacji pieca gazowego. Uczenie przeprowadzono dla całej bazy danych Boxa-Jenkinsa.

Autor	Liczba reguł	$J_{RMSE}$
Tong	19	0.6848
Xu-Lu	25	0.5727
Pedrycz	81	0.5656
Box-Jenkins	-	0.4494
Sugeno-Yasukawa	6	0.4358
Chen i wsp.	3	0.2678
Lin-Cunningham	4	0.2664
Sugeno-Tanaka	2	0.2607
Wang-Langari	2	0.2569
Zikidis-Vasilakos	6	0.2530
Kim-Park-Ji	2	0.2345
Kim-Park	2	0.2190
ANBLIR (implikacja Gödela)	2	0.1892
ANNBFIS	3	0.1791
Czogała-Łęski	6	0.1445
ANBLIR (implikacja Fodora)	6	0.1353
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (średnio)</b>	8 / 9	0.1280
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma min., implikacja Łukasiewicza)</b>	9	<b>0.1193</b>

Wyniki wykorzystane w powyższym zestawieniu zostały zaczerpnięte z literatury: [20], [26], [48], [99], [108], [111], [132], [164], [171], [177], [183], [193]. Średnia wartość błędu średniokwadratowego uzyskiwanego w eksperymentach (liczona jako średnia arytmetyczna dla przypadków stanowiących iloczyn kartezjański operatorów  $t$ -normy i implikacji) wynosi 0.1280, natomiast najlepsza wartość wynosi 0.1193. W obliczeniach pominięto implikację Zadeha, ze względu na występowanie źle uwarunkowanej macierzy ( $\underline{D}^T \underline{D}$ ) (równanie 2.115) w procesie globalnego uczenia parametrów kombinacji liniowej, opisujących położenie środków ciężkości konkluzji rozmytych reguł warunkowych tworzących system i związaną z tym faktem niską jakość systemów rozmytych z parametryzowaną konkluzją, opartych na tej implikacji. Z przeprowadzonych badań wynika, że zastosowanie globalnych metod optymalizacji w postaci obliczeń ewolucyjnych poprawia jakość automatycznie generowanych tą metodą systemów w zadaniu identyfikacji systemów. Uzyskane wyniki klasyfikują rozpatrywany w niniejszej rozprawie system powyżej systemów uzyskiwanych z wykorzystaniem innych technik obliczeń miękkich, m.in. systemów neuronowych i neuronowo-rozmytych.

Na poniższym rysunku przedstawiono sygnały wejściowe oraz sygnał błędu najlepszego uzyskanego rezultatu:



Rysunek 6.3: Identyfikacja pieca gazowego dla całego zbioru danych uczących, implikacja Łukasiewicza,  $t$ -norma minimum. Górny wykres prezentuje sygnał wyjściowy (linia niebieska) oraz sygnał aproksymowany przez system (linia zielona), dolny wykres sygnał błędu.

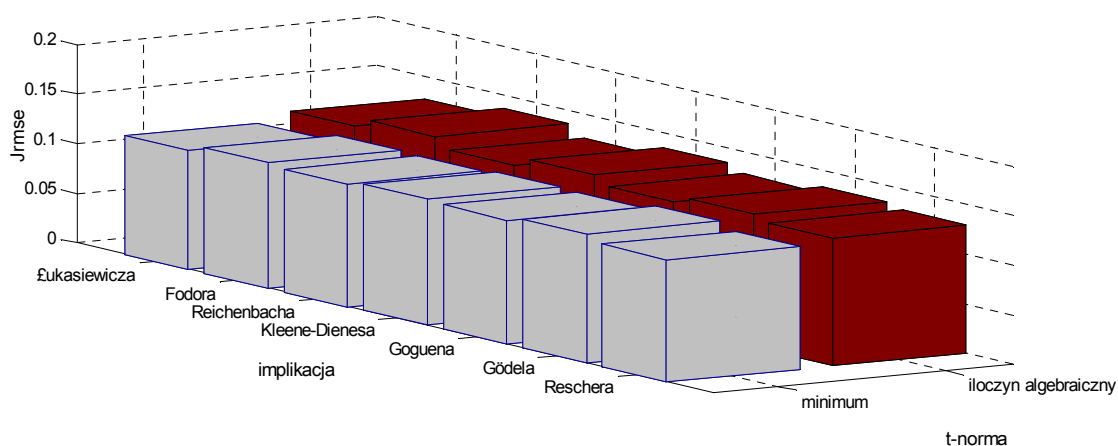
Oceniane systemy rozmyte uzyskano w procesie wielokrotnego uruchomienia trójetapowej metody uczenia, które w zależności od przyjętego parametru pokrycia danych  $\varepsilon$  wygenerowały 6, 7, 8 i 9 reguł warunkowych. Najlepsze rezultaty uzyskano dla 8 i 9 reguł, co potwierdza zaobserwowaną dla opisywanego zagadnienia zależność wzrostu jakości aproksymacji wraz ze wzrostem liczby reguł rozmytych tworzących bazę wiedzy systemu rozmytego.

W poniższej tabeli zestawiono osiągnięte rezultaty w podziale na implikacje i operatory  $t$ -normy dla systemu rozmytego z parametryzowanymi konkluzjami o odpowiednio ośmiu i dziewięciu regułach warunkowych if-then.

Tabela 6.3: Porównanie uzyskanych rezultatów w podziale na implikacje i operatory  $t$ -normy, dla zagadnienia identyfikacji pieca gazowego. Uczenie przeprowadzono dla całej bazy danych Boxa-Jenkinsa.

Implikacja	liczba reguł / $t$ -norma	$J_{RMSE}$
Łukasiewicza	9 / minimum	<b>0.1193</b>
	9 / iloczyn algebraiczny	0.1279
Fodora	9 / minimum	0.1261
	8 / iloczyn algebraiczny	0.1369
Reichenbacha	9 / minimum	0.1239
	9 / iloczyn algebraiczny	0.1262
Kleene-Dienes	9 / minimum	0.1270
	9 / iloczyn algebraiczny	0.1358
Goguena	9 / minimum	0.1251
	9 / iloczyn algebraiczny	0.1276
Gödela	9 / minimum	0.1302
	9 / iloczyn algebraiczny	0.1342
Reschera	9 / minimum	<i>0.1235</i>
	9 / iloczyn algebraiczny	0.1288

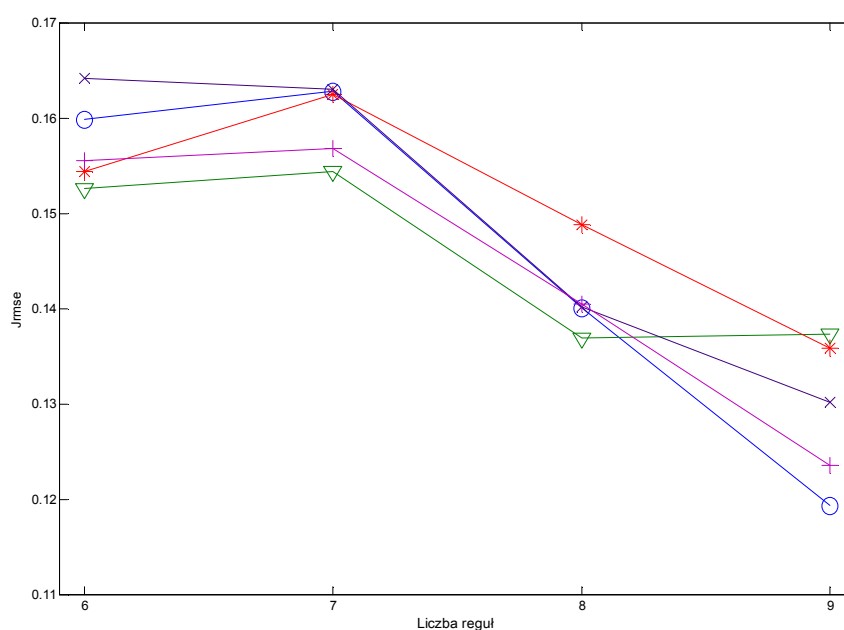
Z powyższej tabeli wynika, że lepsze rezultaty są osiągnięte dla systemów rozmytych z parametryzowaną konkluzją, które wykorzystują jako  $t$ -normę operator minimum. Najlepszą wartość (dla implikacji Łukasiewicza) oznaczono czcionką pogrubioną, kolejną (dla implikacji Reschera) wyróżniono kursywą. Powyższe zestawienie zaprezentowano poniżej w formie graficznej.



Rysunek 6.4: Graficzna prezentacja aproksymacji danych pomiarowych, dla zagadnienia identyfikacji pieca gazowego, w podziale na implikacje i  $t$ -normy.

Porównując osiągnięte wyniki jakości systemów dla różnych par implikacja -  $t$ -norma można zauważyć, że różnice pomiędzy systemem najgorszym i najlepszym nie przekraczają 0.02 wartości bezwzględnej  $J_{\text{RMSE}}$ , zatem skuteczność trójetapowej metody uczenia w poszukiwaniu optimum globalnego (systemu, który najlepiej aproksymuje dane pomiarowe) przy braku wcześniejszych wskazań w zakresie preferowanych dla rozpatrywanego zagadnienia operatorów  $t$ -normy oraz implikacji nie przekreśla szans na uzyskanie tą metodą dobrych wyników.

W metodzie trójetapowej liczba wygenerowanych reguł może być niedeterministycznie sterowana za pomocą parametru pokrycia danych  $\varepsilon$ , co przy dużej liczbie uruchomień pozwala na uzyskanie żądanej liczby reguł z dużym prawdopodobieństwem. Poniżej graficznie zaprezentowano zależność błędu średniokwadratowego od liczby reguł budujących bazę wiedzy systemu rozmytego, dla wybranych implikacji i operatora  $t$ -normy:

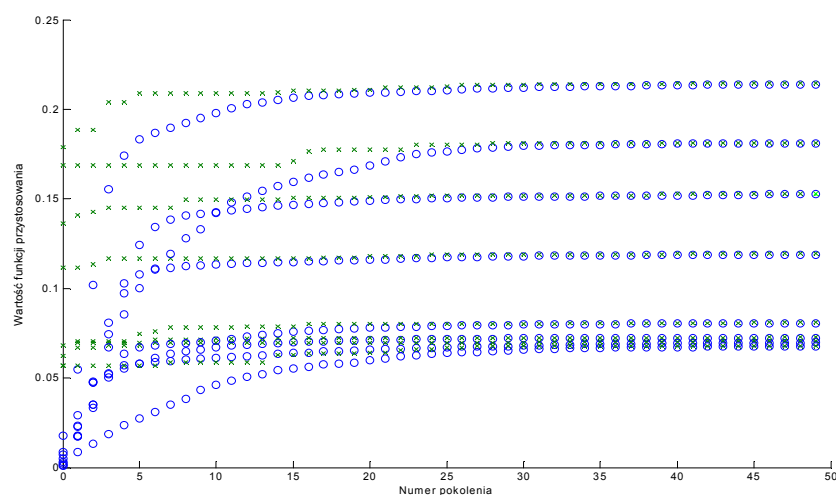


Rysunek 6.5: Wpływ liczby reguł na zdolność aproksymacji danych przez system.

Oznaczenia: x - Godel / minimum, o - Łukasiewicz / minimum,

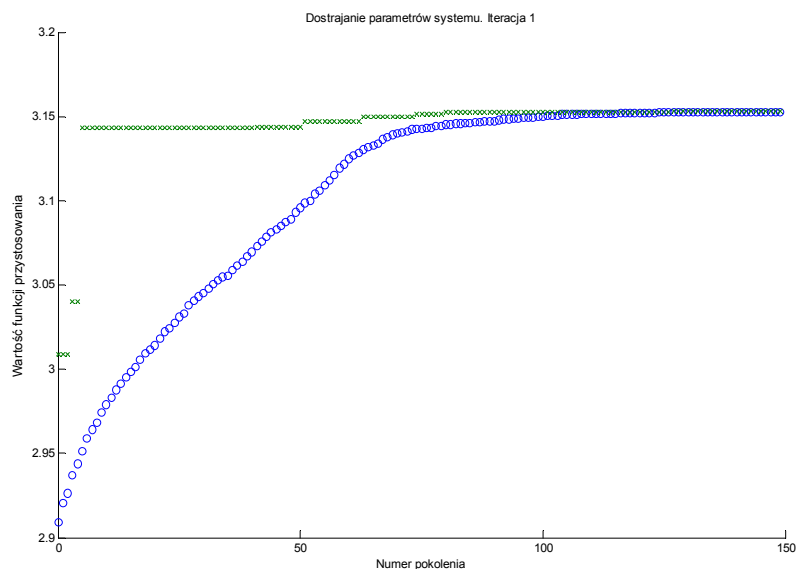
+ - Rescher / minimum, \* - Kleene-Dienes / iloczyn, ∇ - Fodor / iloczyn.

Na poniższym rysunku przedstawiono przebieg procesu ewolucyjnego, na etapie pokrywania danych uczących w wyniku którego uzyskano bazę wiedzy o 8 regułach rozmytych.

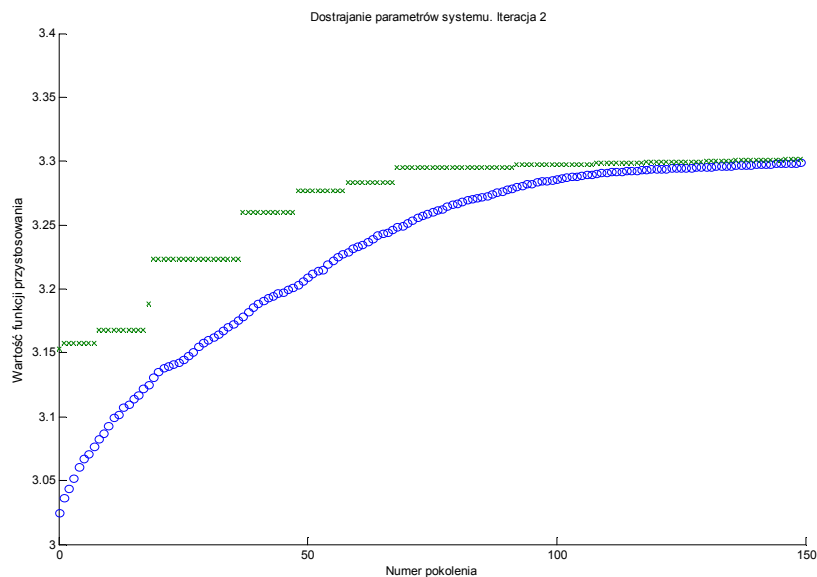


Rysunek 6.6: Proces pokrywania danych uczących, 8 reguł. Oznaczenia: x - wartość funkcji przystosowania najlepszego osobnika, o - średnia wartość funkcji przystosowania pokolenia.

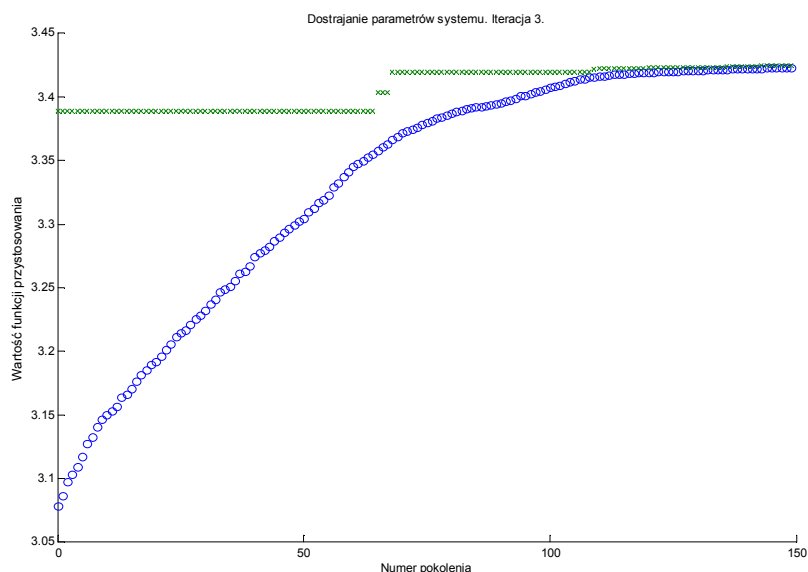
Na poniższych rysunkach przedstawiono przebieg procesu dostrajania parametrów systemu (trzeci etap trójetapowej metody uczenia), który doprowadził do odnalezienia najlepszego systemu, opisywanego w tym podrozdziale.



Rysunek 6.7: Pierwsza iteracja dostrajania parametrów systemu. Oznaczenia: x - wartość funkcji przystosowania najlepszego osobnika, o - średnia wartość funkcji przystosowania pokolenia.



Rysunek 6.8: Druga iteracja dostrajania parametrów systemu. Oznaczenia: x - wartość funkcji przystosowania najlepszego osobnika, o - średnia wartość funkcji przystosowania pokolenia.



Rysunek 6.9: Trzecia iteracja dostrajania parametrów systemu. Oznaczenia: x - wartość funkcji przystosowania najlepszego osobnika, o - średnia wartość funkcji przystosowania pokolenia.

Jak widać z powyższych wykresów, z powodu zastosowania selekcji elitarniej następuje stopniowe zanikanie różnorodności pokolenia bazowego. Wartości określające liczbę pokoleń w poszczególnych etapach trójetapowej metody uczenia, stanowiące warunek zatrzymania strategii ewolucyjnych były dobierane eksperymentalnie. Dobrano takie wartości, dla których z powodu znikomej różnorodności pokolenia nie następuje już poprawa w funkcji przystosowania osobnika najlepszego. W procesie iteracyjnego pokrywania, pierwsza reguła uzyskuje zawsze największe wartości funkcji przystosowania, ponieważ nie podlega ona ocenie kompatybilności z pozostałymi regułami. W procesie dostrajania parametrów można zauważyć, że wartość funkcji przystosowania najlepszego osobnika w pierwszym uruchomieniu procesu dostrajania stanowi wartość bazową dla najlepszego osobnika w uruchomieniu drugim, jednakże dla uruchomienia drugiego i trzeciego już taka zależność nie zachodzi. Jest to związane z elementem losowości, występującym podczas generowania pokolenia startowego, co powoduje, że już w wyniku losowania parametrów osobników tworzących pokolenie startowe jest możliwe odnalezienie osobnika lepszego.

### Wyniki badania zdolności uogólniania

Badanie zdolności uogólniania systemu przeprowadzono dla zbioru uczącego składającego się z pierwszych 100 rekordów bazy danych pomiarowych Boxa-Jenkinsa. Zdolność uogólniania mierzono na pozostałych 190 rekordach danych, tworzących zbiór testowy. Jako miarę jakości przyjęto błąd średniokwadratowy dla danych testujących. W tabelach podano również błąd średniokwadratowy dla danych uczących oraz łączny błąd średniokwadratowy dla całego zbioru danych Boxa-Jenkinsa. Przyjęty przez autora podział na dane uczące i dane testowe jest zgodny z najczęściej spotykanym w literaturze, dlatego też istnieje możliwość porównania osiągniętych wyników z wynikami prezentowanymi w innych publikacjach, m.in. w [111].

W celu wygenerowania systemu rozmytego zastosowano trójetapową metodę uczenia, w której ostatni, trzeci etap uczenia był wykonywany dwukrotnie, przy czym w drugim uruchomieniu procesu dostrajania parametrów (trzeciego etapu metody trójetapowej) wykorzystywano najlepszy system, uzyskany w pierwszym uruchomieniu. Najlepszy system rozmyty poddawany ocenie stanowił osobnik o największej funkcji przystosowania, uzyskany w drugiej iteracji trzeciego etapu trójetapowej metody uczenia (w procesie dostrajania parametrów).

Poniżej zestawiono zbiór parametrów sterujących wykonaniem poszczególnych etapów.

Tabela 6.4: Parametry sterujące trójetapowej metody uczenia dla zagadnienia identyfikacji pieca gazowego. Uczenie dla zbioru 100 pierwszych rekordów bazy danych Boxa-Jenkinsa.

Etap	Parametr	Symbol	Wartość
1.	Liczebność pokolenia bazowego.	$\mu$	60
	Liczebność pokolenia potomnego .	$\lambda$	20
	Liczba pokoleń.	$\mathcal{L}$	50
	Prawdopodobieństwo krzyżowania.	$P_c$	1
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.8
	Współczynnik rekordów pokrywanych przez regułę do rekordów niepokrywanych (zastosowany w funkcji oceny osobnika).	$\kappa$	0.05
	Minimalny stopień kompatybilności danych z regułą (zastosowany w funkcji oceny osobnika)..	$\omega$	0.1
	Minimalny stopień pokrycia rekordu przez reguły. Różne wartości parametru pozwoliły na uzyskanie różnej liczebności reguł (zastosowany na poziomie algorytmu pokrywającego).	$\varepsilon$	1.0 1.2 1.5 1.8
	Parametr krzyżowania uśredniającego.	$a$	0.35
	Parametr periodycznej mutacji nierównomiernej.	$b$	10
	Poziom funkcji gaussowskich dla wyliczenia szerokości nośnika w funkcjach przynależności przesłanek reguł.	$G_L$	0.25
2.	Liczebność pokolenia bazowego.	$M$	50
	Liczba pokoleń.	$K$	150
	Prawdopodobieństwo mutacji (dla genu)	$P_m$	0.005
	Prawdopodobieństwo krzyżowania.	$P_c$	0.75
	Minimalny wskaźnik pokrycia danych przez bazę reguł (zastosowany w funkcji oceny osobnika).	$\tau$	0.25
3.	Liczebność pokolenia bazowego.	$\mu$	100
	Liczebność pokolenia potomnego .	$\lambda$	20
	Liczba pokoleń dla pojedynczego uruchomienia strategii ewolucyjnej.	$\mathcal{L}$	150
	Prawdopodobieństwo krzyżowania (dla pary osobników)	$P_c$	0.6
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.6
	Poziom funkcji gaussowskich dla wyliczenia nośnika.	$G_L$	0.9
	Parametr krzyżowania uśredniającego.	$a$	0.5
	Parametr periodycznej mutacji nierównomiernej.	$b$	5



W procesie ustalania parametrów konkluzji metodą uczenia globalnego dla źle uwarunkowanych macierzy ( $D^T D$ ) (równanie 2.115) zastosowano na ich przekątnej głównej korektę równą  $1.15 \cdot 10^{-2}$ .

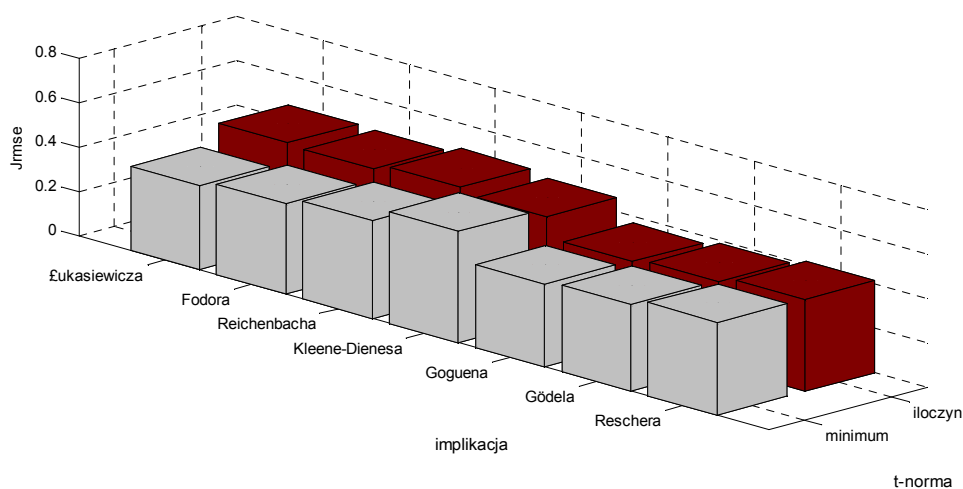
W procesie ewolucyjnym uzyskano systemy rozmyte z parametryzowanymi konkluzjami o bazie wiedzy składającej się odpowiednio z 7, 8 i 9 rozmytych reguł warunkowych **if-then**. Największą zdolność uogólniania uzyskano dla systemu składającego się z 9 rozmytych reguł warunkowych **if-then**, przy zastosowaniu logicznej interpretacji reguł z implikacją Goguena oraz  $t$ -normą iloczyn. Szczegółowe zestawienie parametrów funkcji przynależności przesłanek, szerokości nośnika konkluzji oraz parametrów opisujących kombinację liniową określającą położenie środków ciężkości konkluzji w regułach znajduje się w Dodatku A niniejszej rozprawy.

W poniższej tabeli zestawiono najlepsze uzyskane wyniki zdolności uogólniania systemu, w zależności od stosowanej implikacji i operatora  $t$ -normy. Dodatkowo zestawienie zostało uzupełnione o wyniki błędu średniokwadratowego uzyskane dla uczącego zbioru danych.

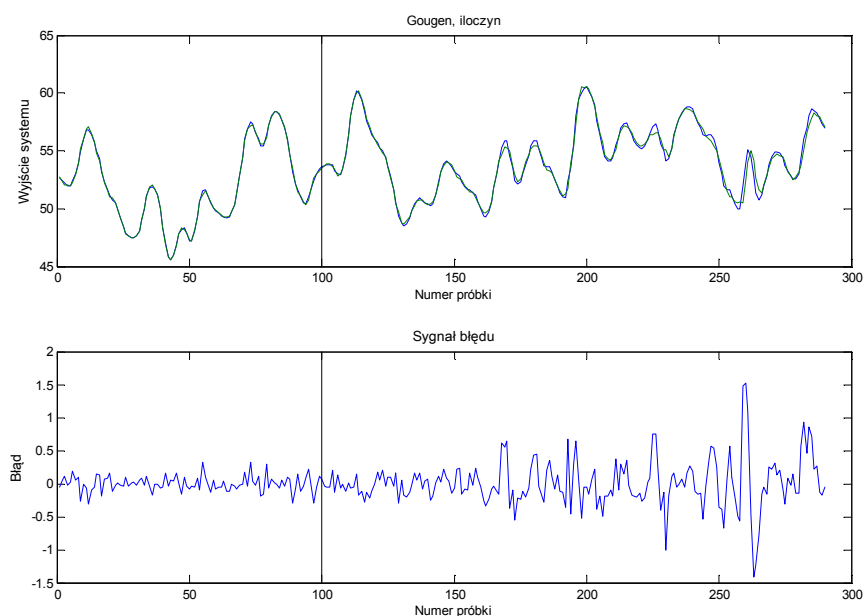
Tabela 6.5: Porównanie uzyskanych wyników uogólniania dla zagadnienia identyfikacji pieca gazowego. Wyniki dla zbioru uczącego i testującego.

Implikacja	liczba reguł / $t$ -norma	$J_{RMSE}$	
		uczący	testowy
Łukasiewicza	7 / minimum	0.1047	0.3810
	7 / iloczyn	0.0970	0.4658
Fodora	8 / minimum	0.1010	0.4078
	9 / iloczyn	0.0886	0.4567
Reichenbacha	7 / minimum	0.0935	0.4423
	7 / iloczyn	0.0869	0.4839
Kleene-Dienes	7 / minimum	0.0885	0.5002
	9 / iloczyn	0.0759	0.4617
Zadeha	8 / minimum	0.0803	0.4565
	9 / iloczyn	0.0707	12.2800
Goguena	7 / minimum	0.1126	0.3733
	7 / iloczyn	0.1173	<b>0.3666</b>
Gödela	9 / minimum	0.1044	0.3890
	8 / iloczyn	0.1082	0.3858
Reschera	9 / minimum	0.0888	0.4202
	9 / iloczyn	0.1029	0.4156

Analizując wyniki zawarte w powyższej tabeli można zauważyć, że system, który dobrze aproksymuje dane uczące gorzej uogólnia wiedzę na dane testowe, na wskutek tzw. przeuczenia, jak to ma miejsce np. dla par (implikacja -  $t$ -norma) Łukasiewicz - iloczyn, Fodor - iloczyn czy Kleene-Dienes - iloczyn. Natomiast systemy, które lepiej uogólniają wiedzę na zbiór testowy, wykazują mniejszą dokładność w aproksymacji danych uczących. Zjawisko to jest zgodne ze statystyczną teorią uczenia. Na poniższym rysunku zaprezentowano graficznie uzyskane wyniki zdolności uogólniania systemu. Pominęto implikację Zadeha.



Rysunek 6.10: Graficzna prezentacja zdolności uogólniania dla zagadnienia identyfikacji pieca gazowego, w podziale na implikacje i  $t$ -normy.



Rysunek 6.11: Identyfikacja pieca gazowego, implikacja Goguena,  $t$ -norma iloczyn. Górny wykres prezentuje sygnał wyjściowy (linia niebieska) oraz odpowiedź systemu (linia zielona), dolny sygnał błędu. Pionowa linia prezentuje podział na zbiór uczący i testowy.

Poniżej przedstawiono szczegółowe wyniki uogólniania systemu, w zależności od liczby reguł tworzących bazę wiedzy systemu rozmytego.

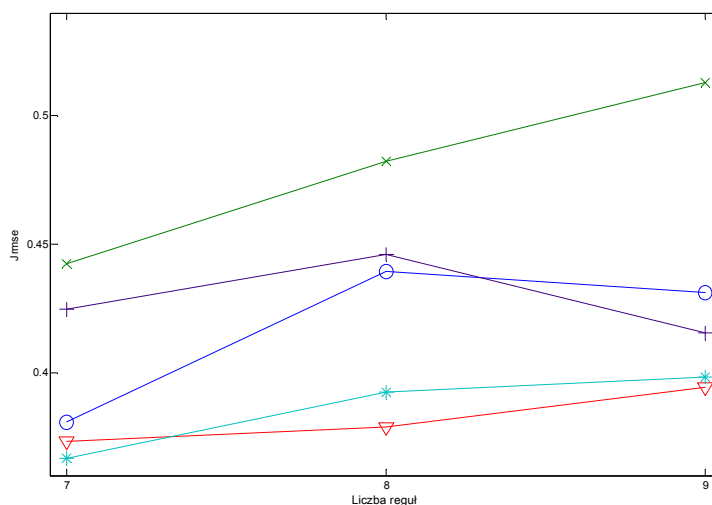
Tabela 6.6: Porównanie uzyskanych wyników uogólniania dla zagadnienia identyfikacji pieca gazowego w podziale na liczbę reguł. Wyniki dla zbioru testującego.

Implikacja	<i>t</i> -norma	Liczba reguł		
		7	8	9
Łukasiewicza	minimum	0.3810	0.4393	0.4312
	iloczyn	0.4658	0.4923	0.5588
Fodora	minimum	0.4853	0.4078	0.4880
	Iloczyn	0.4890	0.5251	0.4567
Reichenbacha	minimum	0.4423	0.4823	0.5130
	iloczyn	0.4839	0.5002	0.5396
Kleene-Dienes	minimum	0.5002	0.9569	0.5234
	iloczyn	1.8216	0.5193	0.4617
Zadeha	minimum	0.4940	0.4565	0.5916
	iloczyn	16.1518	14.3443	12.2800
Goguena	minimum	0.3733	<b>0.3788</b>	0.3944
	iloczyn	<b>0.3666</b>	0.3922	0.3985
Gödela	minimum	0.3931	0.4270	<b>0.3890</b>
	iloczyn	0.4183	0.3858	0.3896
Reschera	minimum	0.4802	1.2732	0.4202
	iloczyn	0.4245	0.4459	0.4156

Najlepsze uzyskane wyniki oznaczono czcionką pogrubioną, kolejne kursywą. Wyniki uzyskane dla implikacji Zadeha i *t*-normy iloczynu, na etapie wyznaczania parametrów konkluzji cechują się źle uwarunkowaną macierzą ( $\underline{D}^T \underline{D}$ ) (równanie 2.115), co powoduje drastyczny wzrost błędu średniokwadratowego i w praktyce wyklucza zastosowanie takiego połączenia implikacji i operatora *t*-normy. Na uwagę zasługuje fakt, że kolejny w sensie zdolności uogólniania system również wykorzystuje implikację Goguena, jak również należy zauważyć, iż różnice w jakości systemu, wynikające ze zmiany operatora *t*-normy mają dla tej implikacji nieznaczny wpływ.

Wyniki zestawione w powyższej tabeli wykazują, że dla implikacji Goguena, która tworzy system o najwyższej zdolności uogólniania występuje zjawisko spadku tej zdolności wraz ze wzrostem liczby reguł.

Na poniższym rysunku przedstawiono graficznie zależność zdolności uogólniania od liczby reguł, dla wybranych implikacji.



Rysunek 6.12: Wpływ liczby reguł na zdolność uogólniania systemu, dla wybranych implikacji i operatorów  $t$ -normy. Oznaczenia: x - Reichenbach / minimum, o - Łukasiewicz / iloczyn, + - Rescher / iloczyn, \* - Goguen / iloczyn, ∇ - Goguen / minimum. Zbiór danych testowych.

Poniżej zestawiono osiągnięte wyniki z wynikami pochodzącymi z literatury [111], [112].

Tabela 6.7: Porównanie uzyskanych wyników uogólniania dla zagadnienia identyfikacji pieca gazowego z innymi metodami. Wyniki dla zbioru testującego.

Metoda	Liczba reguł	$J_{RMSE}$ testowy
LS (ang. <i>Least Squares</i> )	6	0.5579
IQP (ang. <i>Iterative Quadratic Programming</i> )	6	0.5537
$\epsilon$ LSSLI (ang. <i><math>\epsilon</math>-Insensitive Learning by Solving a System of Linear Inequalities</i> )	6	0.4956
ANBLIR	8	0.4020
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma minimum, implikacja Gödela)</b>	9	0.3890
ANBLIR	7	0.3870
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma minimum, implikacja Goguena)</b>	8	0.3788
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma iloczyn, implikacja Goguena)</b>	7	<b>0.3666</b>
IQP <sup>1</sup>	2	0.3442

<sup>1</sup> Rezultat uzyskany dla uczenia lokalnego.

Z powyższej tabeli wynika, że przy podobnej liczbie reguł w bazie wiedzy, zaproponowana w niniejszej rozprawie metoda uzyskuje lepsze wyniki uogólniania, niż systemy neuronowo-rozmyte i niektóre metody oparte na  $\varepsilon$ -nieczułym modelowaniu rozmytym.

### Podsumowanie

Uzyskane wyniki wykazują, że zaproponowana w niniejszej rozprawie metoda automatycznego generowania systemów rozmytych z parametryzowanymi konkluzjami na podstawie numerycznych danych uczących jest metodą dającą bardzo dobre rezultaty w zastosowaniach identyfikacji systemów. Przedstawione w niniejszym podrozdziale wyniki potwierdzają, iż dobre rezultaty mogą być osiągnięte zarówno w zakresie aproksymacji danych pomiarowych jak i w przypadku systemów, które uogólniają wiedzę. Dodatkowo dla zagadnienia aproksymacji danych pomiarowych metoda osiągała bardzo dobre rezultaty niezależnie od zastosowanej implikacji czy przyjętego operatora  $t$ -normy. W praktyce oznacza to, że badana w niniejszej rozprawie trójetapowa metoda automatycznego generowania systemów rozmytych z parametryzowanymi konkluzjami na podstawie danych numerycznych jest metodą uniwersalną, co oznacza, że do uzyskania tą metodą wysokiej jakości systemów odwzorowujących rzeczywistość nie jest potrzebna dogłębna znajomość badanego zjawiska.

### 6.2.2 Przewidywanie liczby plam na słońcu (Sunspots)

Drugim badanym w ramach niniejszej rozprawy zagadnieniem była predykcja średniej rocznej liczby plam na słońcu. W tym celu wykorzystano bazę danych pochodzącą z pracy [178]. Dane pomiarowe zawierają poziom aktywności słońca w okresie od 1700 do 1979r. z okresem rocznym. Na potrzeby uczenia systemu dane przygotowano w następującej postaci:

$$\begin{aligned} & [\underline{x}_n^T, y_n], \\ & \underline{x}_n = [x(n-1), x(n-2), \dots, x(n-12)]^T, \\ & y_n = x(n). \end{aligned} \quad (6.4)$$

Zbiór uczący został utworzony z pierwszych 100 rekordów, natomiast pozostałe 168 rekordów utworzyło zbiór testujący.

Identyfikację przeprowadzono dla implikacji Łukasiewicza, Fodora, Reichenbacha, Kleene-Dienes, Zadeha, Goguena, Gödela i Reschera oraz dla operatorów  $t$ -normy minimum i iloczynu algebraicznego.

### Wyniki badania zdolności uogólniania

Wyniki badania zdolności uogólniania przeprowadzono dla zbioru uczącego składającego się z pierwszych 100 rekordów bazy danych pomiarowych bazy danych Sunspots [178]. Zdolność uogólniania mierzono na pozostałych 168 rekordach danych, tworzących zbiór testowy. Jako miarę jakości przyjęto błąd średniokwadratowy dla zbioru danych testujących. Przyjęty przez autora podział na dane uczące i dane testowe jest zgodny z najczęściej spotykanym w literaturze, dlatego też istnieje możliwość porównania osiągniętych wyników z wynikami prezentowanymi w innych publikacjach, m.in. w [112].

W celu wygenerowania systemu rozmytego zastosowano trójetapową metodę uczenia, w której ostatni, trzeci etap uczenia był wykonywany dwukrotnie, przy czym w drugim uruchomieniu procesu dostrajania parametrów (trzeciego etapu metody trójetapowej) jako bazę do obliczeń wykorzystywano najlepszy system, uzyskany w pierwszym uruchomieniu. Najlepszy system rozmyty poddawany ocenie stanowił osobnik o największej funkcji przystosowania, uzyskany w drugiej iteracji trzeciego etapu trójetapowej metody uczenia (w procesie dostrajania parametrów).

Poniżej zestawiono zbiór parametrów sterujących wykonaniem poszczególnych etapów.

Tabela 6.8: Parametry sterujące trójetapowej metody uczenia dla zagadnienia przewidywania ilości plam na słońcu. Uczenie dla zbioru 100 pierwszych rekordów bazy danych Sunspots.

Etap	Parametr	Symbol	Wartość
1.	Liczebność pokolenia bazowego.	$\mu$	60
	Liczebność pokolenia potomnego .	$\lambda$	20
	Liczba pokoleń.	$\mathcal{L}$	50
	Prawdopodobieństwo krzyżowania.	$P_c$	1
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.8
	Współczynnik rekordów pokrywanych przez regułę do rekordów niepokrywanych (zastosowany w funkcji oceny osobnika).	$\kappa$	0.05
	Minimalny stopień kompatybilności danych z regułą (zastosowany w funkcji oceny osobnika)..	$\omega$	0.1
	Minimalny stopień pokrycia rekordu przez reguły. Różne wartości parametru pozwoliły na uzyskanie różnej liczebności reguł (zastosowany na poziomie algorytmu pokrywającego).	$\varepsilon$	0.8 1.0 1.2 1.5
	Parametr krzyżowania uśredniającego.	$a$	0.35
	Parametr periodycznej mutacji nierównomiernej.	$b$	10
	Poziom funkcji gaussowskich dla wyliczenia szerokości nośnika w funkcjach przynależności przesłanek reguł.	$G_L$	0.25
2.	Liczebność pokolenia bazowego.	$M$	50
	Liczba pokoleń.	$K$	150
	Prawdopodobieństwo mutacji (dla genu)	$P_m$	0.005
	Prawdopodobieństwo krzyżowania.	$P_c$	0.75
	Minimalny wskaźnik pokrycia danych przez bazę reguł (zastosowany w funkcji oceny osobnika).	$\tau$	0.25
3.	Liczebność pokolenia bazowego.	$\mu$	100
	Liczebność pokolenia potomnego .	$\lambda$	20
	Liczba pokoleń dla pojedynczego uruchomienia strategii ewolucyjnej.	$\mathcal{L}$	150
	Prawdopodobieństwo krzyżowania (dla pary osobników)	$P_c$	0.6
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.6
	Poziom funkcji gaussowskich dla wyliczenia nośnika.	$G_L$	0.9
	Parametr krzyżowania uśredniającego.	$a$	0.5
	Parametr periodycznej mutacji nierównomiernej.	$b$	5

W procesie ustalania parametrów konkluzji metodą uczenia globalnego dla źle uwarunkowanych macierzy ( $D^T D$ ) (równanie 2.115) zastosowano na ich przekątnej głównej korektę równą  $1.15 \cdot 10^{-2}$ .

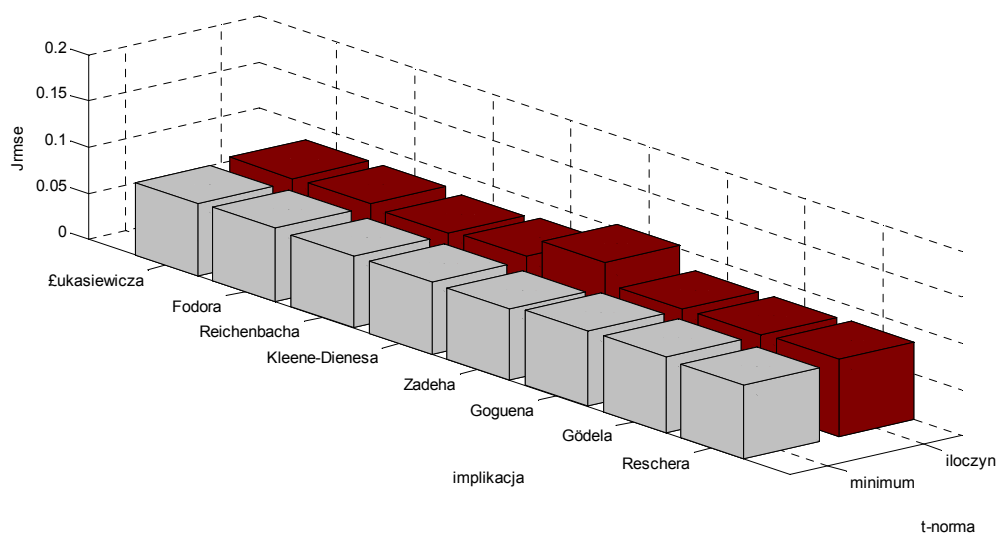
W procesie ewolucyjnym uzyskano systemy rozmyte z parametryzowanymi konkluzjami o bazie wiedzy składającej się z 3 do 6 oraz 8 reguł. Największą zdolność uogólniania uzyskano dla systemu składającego się z 4 rozmytych reguł warunkowych **if-then**, przy zastosowaniu logicznej interpretacji reguł z implikacją Zadeha, dla operatora  $t$ -normy minimum. W poniższej tabeli zestawiono najlepsze uzyskane wyniki zdolności uogólniania systemu, w zależności od zastosowanej implikacji i operatora  $t$ -normy. Dodatkowo zestawienie zostało uzupełnione o wyniki błędu średniokwadratowego uzyskane dla uczącego zbioru danych.

Tabela 6.9: Porównanie uzyskanych wyników uogólniania dla zagadnienia przewidywania ilości plam na słońcu (Sunspots). Wyniki dla zbioru uczącego i testującego.

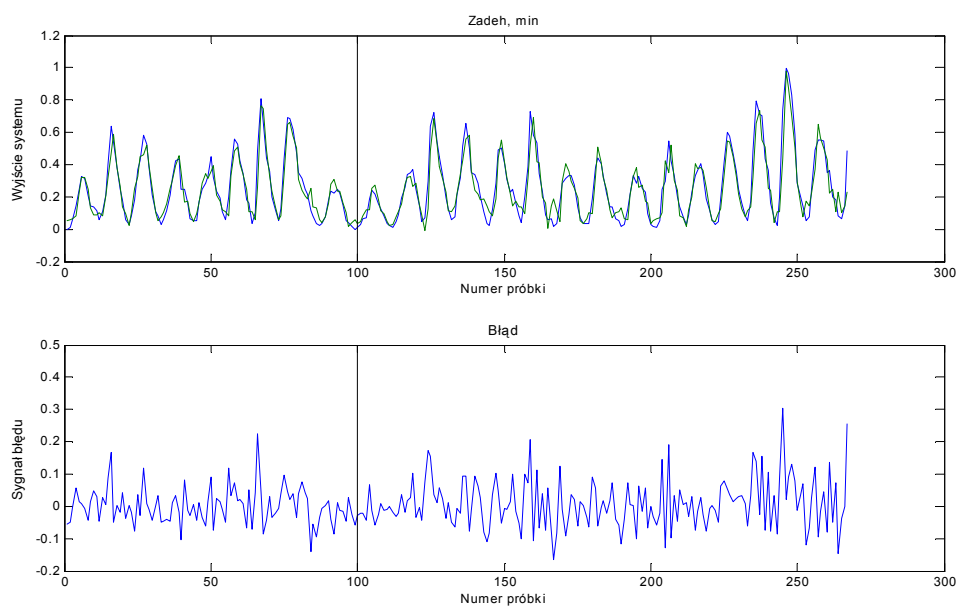
Implikacja	liczba reguł / $t$ -norma	$J_{RMSE}$	
		uczący	testowy
Łukasiewicza	6 / minimum	0.0631	0.0784
	6 / iloczyn	0.0522	0.0823
Fodora	5 / minimum	0.0660	0.0801
	5 / iloczyn	0.0617	0.0832
Reichenbacha	6 / minimum	0.0608	0.0778
	6 / iloczyn	0.0535	0.0793
Kleene-Dienes	6 / minimum	0.0562	0.0787
	3 / iloczyn	0.0660	0.0826
Zadeha	4 / minimum	0.0558	<b>0.0770</b>
	6 / iloczyn	0.0441	0.1044
Goguena	8 / minimum	0.0630	0.0820
	6 / iloczyn	0.0698	0.0824
Gödela	6 / minimum	0.0659	0.0814
	6 / iloczyn	0.0685	0.0831
Reschera	5 / minimum	0.0605	0.0796
	6 / iloczyn	0.0654	0.0839

Analizując powyższe zestawienie można zauważyć, że większość dobrych wyników przypada dla systemów o bazie wiedzy składającej się z 6 reguł warunkowych, jednak system o największej zdolności uogólniania uzyskano dla bazy wiedzy składającej się z 4 reguł warunkowych. Na uwagę zwraca fakt, iż dla 6 par implikacja –  $t$ -norma wskaźnik błędu średniokwadratowego dla zbioru testującego przyjmuje wartości poniżej 0.08.

Na poniższym rysunku graficznie zaprezentowano uzyskane rezultaty.



Rysunek 6.13: Graficzna prezentacja zdolności uogólniania dla zagadnienia przewidywania ilości plam na słońcu, w podziale na implikacje i t-normy.



Rysunek 6.14: Przewidywanie liczby plam na słońcu, implikacja Zadeha,  $t$ -norma minimum. Górny wykres prezentuje sygnał wyjściowy (linia niebieska) oraz odpowiedź systemu (linia zielona), dolny sygnał błęd. Pionowa linia prezentuje podział na zbiór uczący i testowy.

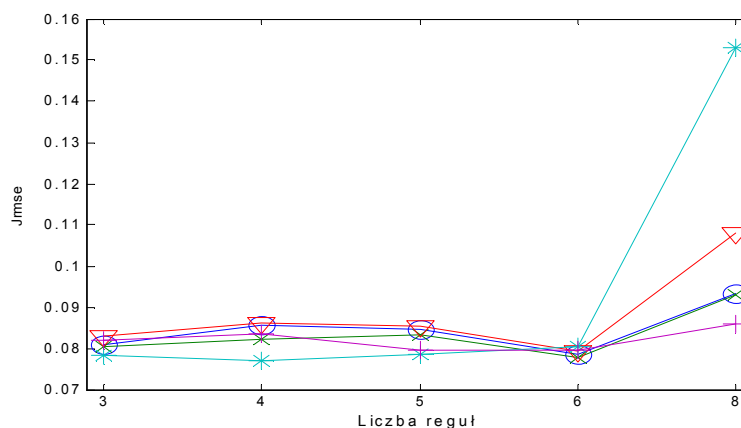
Poniżej przedstawiono szczegółowe wyniki uogólniania wiedzy, w zależności od liczby reguł tworzących bazę wiedzy systemu rozmytego.



Tabela 6.10: Porównanie uzyskanych wyników uogólniania dla zagadnienia przewidywania ilości plam na słońcu (Sunspots). Wyniki dla zbioru testującego.

Implikacja	<i>t</i> -norma	Liczba reguł				
		3	4	5	6	8
Łukasiewicza	minimum	0.0810	0.0856	0.0846	0.0784	0.0932
	iloczyn	0.0856	0.0860	0.0878	0.0823	0.1029
Fodora	minimum	0.0811	0.0815	0.0801	0.0801	0.0915
	iloczyn	0.0865	0.0858	0.0832	0.0861	0.0913
Reichenbacha	minimum	0.0805	0.0822	0.0832	0.0778	0.0931
	iloczyn	0.0829	0.0861	0.0855	0.0793	0.1080
Kleene-Dienes	minimum	0.0798	0.0841	0.0790	0.0787	0.1028
	iloczyn	0.0826	0.0844	0.0866	0.0847	0.1097
Zadeha	minimum	0.0784	<b>0.0770</b>	0.0786	0.0805	0.1531
	iloczyn	0.1114	0.1454	0.1214	0.1044	0.1806
Goguena	minimum	0.0845	0.0850	0.0827	0.0829	0.0820
	iloczyn	0.0835	0.0843	0.0836	0.0824	0.0852
Gödela	minimum	0.0834	0.0856	0.0839	0.0814	0.0819
	iloczyn	0.0856	0.0852	0.0840	0.0831	0.0839
Reschera	minimum	0.0819	0.0835	0.0796	0.0796	0.0859
	iloczyn	0.0840	0.0886	0.0847	0.0839	0.0910

Porównując wartości uzyskanych wyników widać, że podobnie do wyników oceny zdolności uogólniania systemu dla zagadnienia identyfikacji pieca gazowego, dla implikacji Zadeha i operatora iloczynu jako *t*-normy uzyskano najgorsze wyniki, znacznie odbiegające od pozostałych wyników. Natomiast najlepsza uzyskana zdolność uogólniania uzyskano dla implikacji Zadeha i *t*-normy minimum. Na poniższym rysunku zaprezentowano graficznie zależność jakości systemu od liczby reguł tworzących jego bazę wiedzy, dla wybranych implikacji i operatorów *t*-normy.



Rysunek 6.15: Wpływ liczby reguł na zdolność uogólniania systemu, dla wybranych implikacji i operatorów *t*-normy. Oznaczenia: x - Reichenbach / minimum, o - Łukasiewicz / minimum, + - Rescher / minimum, \* - Zadeh / minimum, ∇ - Kleene-Dienes / iloczyn. Zbiór danych testowych.

Z powyższego wykresu oraz danych numerycznych wynika, że w zagadnieniu przewidywania ilości plam na słońcu nie rysuje się wyraźna zależność pomiędzy jakością systemu, a ilością reguł dla baz wiedzy o niskiej liczebności reguł. Natomiast wzrost wartości błędu średniokwadratowego dla ośmiu reguł rozmytych następuje dla wszystkich par implikacja –  $t$ -norma, co powoduje drastyczne zmniejszenie zdolności.

Poniżej zestawiono osiągnięte wyniki z wynikami pochodzącymi z literatury [112] dla porównywalnej liczby reguł.

Tabela 6.11: Porównanie uzyskanych wyników uogólniania dla zagadnienia przewidywania ilości plam na słońcu (Sunspots). Wyniki dla zbioru testującego.

Metoda	Liczba reguł	$J_{RMSE}$ testowy
LS <sup>2</sup>	3	0.08416
IQP	3	0.08032
$\varepsilon$ LSSL <sup>2</sup>	4	0.07945
IQP <sup>2</sup>	5	0.07803
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma minimum, implikacja Zadeha)</b>	4	<b>0.07695</b>

Z powyższej tabeli wynika, że przy podobnej komplikacji bazy wiedzy, zaproponowana w niniejszej rozprawie metoda uzyskuje lepsze wyniki uogólniania, niż systemy uzyskane metodami opartymi na  $\varepsilon$ -nieczułym uczeniu.

### Podsumowanie

Uzyskane wyniki potwierdzają wysoką zdolność opisywanej trójetapowej metody uczenia do tworzenia systemów rozmytych, które cechują się dobrą zdolnością uogólniania posiadanej wiedzy, podobnie jak to miało miejsce w poprzednim eksperymencie identyfikacji pieca gazowego. Metoda uzyskuje zadowalające wyniki dla dowolnej kombinacji implikacji i operatora  $t$ -normy za wyjątkiem pary Zadech / iloczyn algebraiczny, podobnie jak to ma miejsce poprzednim eksperymencie. Rezultaty potwierdzają uniwersalność systemu w przedstawionych zastosowaniach.

### 6.2.3 Predykcja chaotycznych szeregów czasowych (Mackey–Glass)

Przedstawione w niniejszym podrozdziale wyniki dotyczą predykcji szeregu czasowego opisującego zmianę gęstości białych krwinek, o zachowaniu zgodnym z matematycznym opisem wyrażonym w formie równania Mackey’a-Glassa [159]:

$$\frac{\partial x(t)}{\partial t} = \frac{a \cdot x(t - \tau)}{1 + [x(t - \tau)]^{10}} - b \cdot x(t), \quad (6.5)$$

<sup>2</sup> Rezultat uzyskany dla uczenia lokalnego.

gdzie  $x(t)$  jest gęstością białych krwinek,  $a$ ,  $b$ ,  $\tau$  są stałymi. W celu umożliwienia porównania wyników wykorzystano standardową bazę danych (tzw. bazę R. J. Janga, opisaną w [96]) dostępną w sieci Internet, obejmującą 1000 rekordów danych. Zastosowano podział na część uczącą (pierwsze 500 rekordów) oraz część testującą (pozostałe 500 rekordów) zgodny z podziałami stosowanymi w innych pracach, m.in. w [29], [48], [96], [111], [113].

Przyjęto następującą postać danych wejściowych:

$$\begin{aligned} & [\underline{x}_n^T(k), t_n(k)], \\ & \underline{x}_n(k) = [x(k), x(k-6), x(k-12), x(k-18)]^T, \\ & y_n(k) = x(k+6), \end{aligned} \quad (6.6)$$

zatem rozwiązywany problem stanowi zagadnienie predykcji 6.-krokowej.

Jako wskaźnik jakości przyjęto (podobnie jak to miało miejsce w poprzednich przykładach) wartość błędu średniokwadratowego odpowiedzi systemu w porównaniu z wartościami wzorcowymi.

W celu wygenerowania systemu rozmytego zastosowano trójetapową metodę uczenia, w której ostatni, trzeci etap uczenia był wykonywany dwukrotnie, przy czym w drugim uruchomieniu procesu dostrajania parametrów (trzeciego etapu metody trójetapowej) jako bazę obliczeń przyjęto najlepszy system, uzyskany w uruchomieniu poprzednim. Najlepszy system rozmyty poddawany ocenie stanowił osobnik o najwyższej wartości przystosowania, uzyskany w drugiej iteracji trzeciego etapu trójetapowej metody uczenia (w procesie dostrajania parametrów).

Poniżej zestawiono zbiór parametrów sterujących wykonaniem poszczególnych etapów.

Tabela 6.12: Parametry sterujące trójetapowej metody uczenia dla zagadnienia predykcji chaotycznych szeregów czasowych (Mackey'a-Glassa). Uczenie dla zbioru 500 pierwszych rekordów.

Etap	Parametr	Symbol	Wartość
1.	Liczebność pokolenia bazowego.	$\mu$	60
	Liczebność pokolenia potomnego.	$\lambda$	20
	Liczba pokoleń.	$\mathcal{L}$	50
	Prawdopodobieństwo krzyżowania.	$P_c$	1
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.8
	Współczynnik rekordów pokrywanych przez regułę do rekordów niepokrywanych (zastosowany w funkcji oceny osobnika).	$\kappa$	0.05
	Minimalny stopień kompatybilności danych z regułą (zastosowany w funkcji oceny osobnika)..	$\omega$	0.1
	Minimalny stopień pokrycia rekordu przez reguły. Różne wartości parametru pozwoliły na uzyskanie różnej liczebności reguł (zastosowany na poziomie algorytmu pokrywającego).	$\varepsilon$	0.8
			0.9
			1.0
			1.1
			1.3
			1.5
	Parametr krzyżowania uśredniającego.	$a$	0.35

	Parametr periodycznej mutacji nierównomiernej.	$b$	10
	Poziom funkcji gaussowskich dla wyliczenia szerokości nośnika w funkcjach przynależności przesłanek reguł.	$G_L$	0.25
2.	Liczebność pokolenia bazowego.	$M$	50
	Liczba pokoleń.	$K$	150
	Prawdopodobieństwo mutacji (dla genu)	$P_m$	0.005
	Prawdopodobieństwo krzyżowania.	$P_c$	0.75
	Minimalny wskaźnik pokrycia danych przez bazę reguł (zastosowany w funkcji oceny osobnika).	$\tau$	0.25
3.	Liczebność pokolenia bazowego.	$\mu$	100
	Liczebność pokolenia potomnego.	$\lambda$	20
	Liczba pokoleń dla pojedynczego uruchomienia strategii ewolucyjnej.	$\mathcal{L}$	150
	Prawdopodobieństwo krzyżowania (dla pary osobników)	$P_c$	0.6
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.6
	Poziom funkcji gaussowskich dla wyliczenia nośnika.	$G_L$	0.9
	Parametr krzyżowania uśredniającego.	$a$	0.25
	Parametr periodycznej mutacji nierównomiernej.	$b$	5

W procesie ustalania parametrów konkluzji metodą uczenia globalnego dla źle uwarunkowanych macierzy ( $D^T D$ ) (równanie 2.115) zastosowano na ich przekątnej głównej korektę równą  $1.15 \cdot 10^{-2}$ .

W procesie ewolucyjnym uzyskano systemy rozmyte z parametryzowanymi konkluzjami o bazie wiedzy składającej się z 4 do 10 rozmytych reguł warunkowych **if-then** (w zależności od przyjętego parametru  $\varepsilon$ ). Najlepszą zdolność uogólniania uzyskano dla systemu wykorzystującego logiczną interpretację reguł z implikacją Kleene-Dienes,  $t$ -normą iloczyn oraz bazą wiedzy składającą się z 4 reguł.

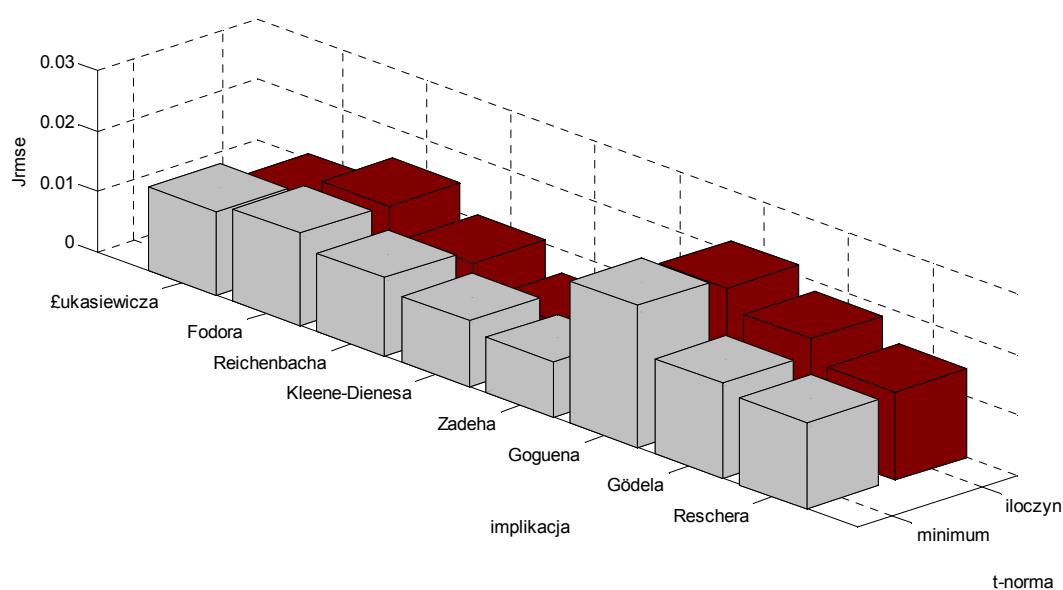
W poniższej tabeli zestawiono najlepsze uzyskane wyniki zdolności uogólniania, w zależności od stosowanej implikacji i operatora  $t$ -normy. Dodatkowo zestawienie zostało uzupełnione o wyniki błędu średniokwadratowego uzyskane dla uczącego zbioru danych.

Tabela 6.13: Porównanie uzyskanych wyników uogólniania dla zagadnienia predykcji szeregów czasowych (Mackey’a-Glassa). Wyniki dla zbioru uczącego i testującego.

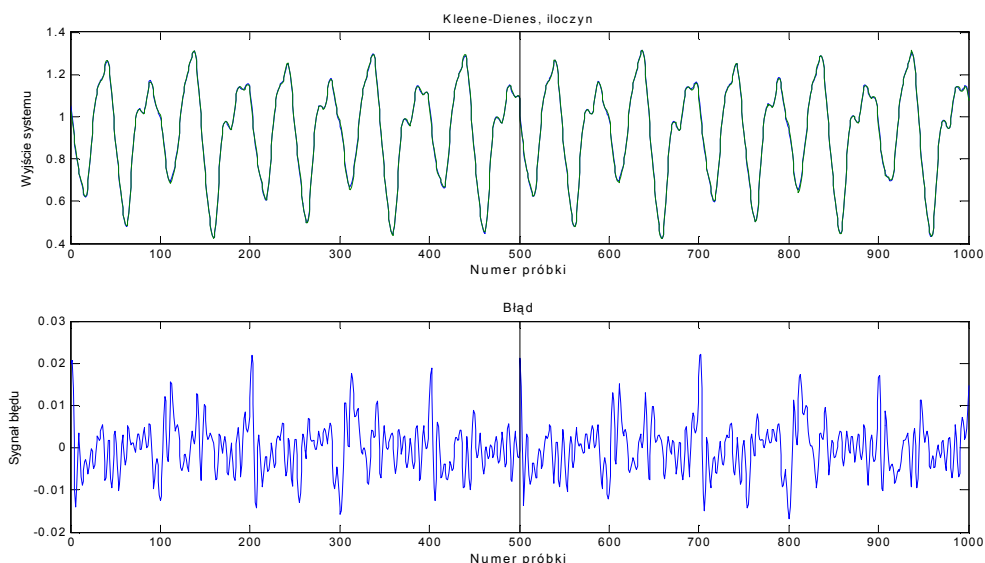
Implikacja	liczba reguł / $t$ -norma	$J_{RMSE}$	
		uczący	testowy
Łukasiewicza	9 / minimum	0.0136	0.0137
	7 / iloczyn	0.0106	0.0107
Fodora	7 / minimum	0.0152	0.0153
	9 / iloczyn	0.0147	0.0150
Reichenbacha	7 / minimum	0.0129	0.0130
	10 / iloczyn	0.0106	0.0106
Kleene-Dienes	7 / minimum	0.0106	0.0110
	4 / iloczyn	0.0059	<b>0.0060</b>
Zadeha	6 / minimum	0.0090	0.0092
	5 / iloczyn	0.0064	0.0067
Goguena	7 / minimum	0.0234	0.0236
	7 / iloczyn	0.0213	0.0216

Gödel	9 / minimum	0.0156	0.0158
	7 / iloczyn	0.0179	0.0183
Reschera	7 / minimum	0.0141	0.0144
	7 / iloczyn	0.0144	0.0146

Analizując wyniki zawarte w powyższej tabeli można zauważyć, że systemy, które dobrze aproksymują dane uczące, równie dobrze uogólniają swoją wiedzę. W powyższym zestawieniu nie występują wyjątkowo złe połączenia implikacji i  $t$ -norm, jak to miało miejsce np. w zagadnieniu identyfikacji pieca gazowego, chociaż różnica pomiędzy systemem najlepszym (Kleene-Dienes, iloczyn), a najgorszym (Goguena, minimum) jest blisko czterokrotna (w sensie przyjętego kryterium jakości, dla danych testujących). Dwa najlepsze wyniki należą do systemów wykorzystujących implikację Kleene-Dienes i Zadeha.



Rysunek 6.16: Graficzna prezentacja zdolności uogólniania dla zagadnienia predykcji szeregów czasowych, w podziale na implikacje i  $t$ -normy.



Rysunek 6.17: Predykcja szeregu czasowego, implikacja Kleene-Dienes,  $t$ -norma iloczyn. Górny wykres prezentuje sygnał wyjściowy (linia niebieska) oraz odpowiedź systemu (linia zielona), dolny sygnał błęd. Pionowa linia prezentuje podział na zbiór uczący i testowy.

Poniżej przedstawiono szczegółowe wyniki, w podziale na liczbę reguł.

Tabela 6.14: Porównanie uzyskanych wyników  $J_{RMSE}$  zdolności uogólniania dla zagadnienia predykcji szeregów czasowych (Mackey'a-Glassa). Wyniki dla zbioru testującego.

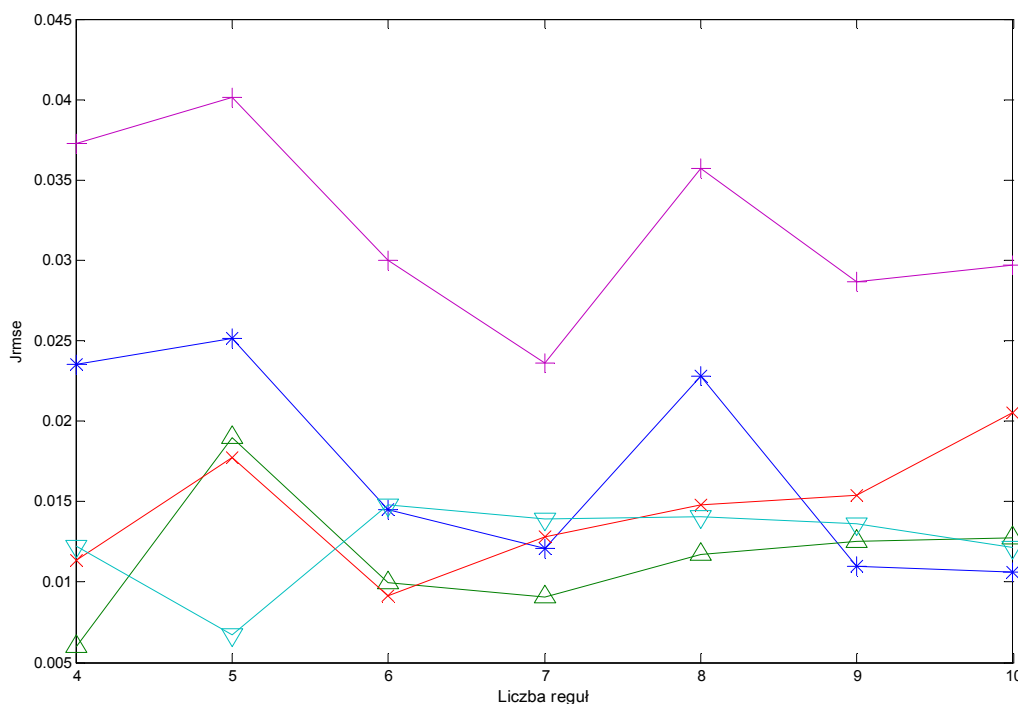
Implikacja	$t$ -norma	Liczba reguł						
		4	5	6	7	8	9	10
Łukasiewicza	minimum	0.0249	0.0284	0.0263	0.0146	0.0268	0.0137	0.0163
	iloczyn	0.0235	0.0244	0.0189	<i>0.0107</i>	0.0294	0.0132	0.0169
Fodora	minimum	0.0209	0.0287	0.0281	0.0153	0.0318	0.0155	0.0228
	iloczyn	0.0216	0.0342	0.0332	0.0172	0.0228	0.0150	0.0257
Reichenbacha	minimum	0.0336	0.0217	0.0350	0.0130	0.0294	0.0135	0.0247
	iloczyn	0.0235	0.0251	0.0145	0.0121	0.0228	<b>0.0110</b>	<b>0.0106</b>
Kleene-Dienes	minimum	0.0314	0.0178	0.0196	0.0110	0.0174	0.0138	0.0200
	iloczyn	<b>0.0060</b>	0.0190	<i>0.0100</i>	<b>0.0091</b>	<b>0.0117</b>	<i>0.0125</i>	0.0128
Zadeha	minimum	<i>0.0113</i>	<i>0.0177</i>	<b>0.0092</b>	0.0128	0.0148	0.0154	0.0206
	iloczyn	0.0123	<b>0.0067</b>	0.0148	0.0140	<i>0.0140</i>	0.0136	<i>0.0122</i>
Goguena	minimum	0.0373	0.0402	0.0300	0.0236	0.0357	0.0287	0.0297
	iloczyn	0.0365	0.0308	0.0407	0.0216	0.0342	0.0256	0.0289
Gödel	minimum	0.0224	0.0411	0.0342	0.0177	0.0354	0.0158	0.0222
	iloczyn	0.0409	0.0356	0.0421	0.0183	0.0315	0.0246	0.0192
Reschera	minimum	0.0212	0.0196	0.0194	0.0144	0.0175	0.0183	0.0245
	iloczyn	0.0281	0.0283	0.0279	0.0146	0.0151	0.0194	0.0200

W powyższym zestawieniu najlepsze wyniki zaznaczono pogrubieniem, kolejne wyróżniono kursywą.

Na uwagę zasługuje fakt, że systemy wykorzystujące implikację Kleene-Dienes oraz  $t$ -normę iloczyn uzyskują bardzo dobre wyniki uogólniania (w tym najlepszy wynik w eksperymencie) dla baz wiedzy składających się z nie więcej niż 8 reguł oraz wyniki niewiele odbiegające od najlepszych dla baz wiedzy składających się z 9 lub 10 rozmytych reguł warunkowych. Porównywalnie dobre wyniki dla baz wiedzy składających się z 4 do 8 reguł warunkowych można uzyskać dla implikacji Zadeha, a dla baz wiedzy składających się z 9 i 10 reguł przy zastosowaniu implikacji Reichenbacha i  $t$ -normy iloczyn.

Z powyższego zestawienia wynika również, iż większa liczba reguł bazy wiedzy systemu rozmytego powoduje zmniejszenie różnic w jakości systemów rozmytych z parametryzowaną konkluzją, w zależności od wykorzystywanej implikacji i  $t$ -normy, w rozważanym zagadnieniu.

Na poniższym rysunku zestawiono w formie graficznej zależność zdolności uogólniania od liczby reguł tworzących bazę wiedzy systemu, dla wybranych par implikacja –  $t$ -norma.



Rysunek 6.18: Wpływ liczby reguł na zdolność uogólniania systemu, dla wybranych implikacji i operatorów  $t$ -normy. Oznaczenia:  $\Delta$  - Kleene-Dienes / iloczyn,  $x$  - Zadeh / minimum,  $\nabla$  - Zadeh / iloczyn,  $*$  - Reichenbach / iloczyn,  $+$  - Goguen / minimum. Zbiór danych testowych.

Największą niezależność uzyskanych wyników od liczby reguł w rozpatrywanym zakresie prezentuje system wykorzystujący implikację Kleene-Dienes i  $t$ -normę iloczyn, oraz system wykorzystujący implikację Zadeha i  $t$ -normę iloczyn. Jak widać z powyższego wykresu, wraz ze wzrostem liczby reguł nieznacznie spada zdolność systemu do uogólniania wiedzy.

Poniżej zestawiono osiągnięte wyniki z wynikami uzyskanymi za pomocą systemu neuronowo-rozmytego metodą ANBLIR [111] oraz ANFIS wraz z algorytmem Optimized Subtractive Clustering [98] dla porównywalnej liczby reguł. Zestawienie obejmuje najlepsze uzyskane wyniki.

Tabela 6.15: Porównanie uzyskanych wyników  $J_{RMSE}$  zdolności uogólniania dla zagadnienia predykcji szeregów czasowych (Mackey’a-Glassa) z wynikami uzyskanymi metodą ANBLIR.

Metoda	Liczba reguł						
	4	5	6	7	8	9	10
Optimized Subtractive Clustering							0.0482
Optimized Subtractive Clustering + ANFIS							0.0458
ANBLIR	0.0087	0.0043	0.0038	0.0029	0.0031	0.0023	0.0022
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami</b>	<b>0.0060</b>	0.0067	0.0092	0.0091	0.0117	0.0110	0.0106

Poniżej zestawiono najlepsze uzyskane wyniki, zaprezentowane w pracach [29], [48], [96], [98], [111], [113] oraz wyniki uzyskane metodą prezentowaną w niniejszej rozprawie.

Tabela 6.16: Porównanie uzyskanych wyników  $J_{RMSE}$  zdolności uogólniania dla zagadnienia predykcji szeregów czasowych (Mackey’a-Glassa) z wynikami uzyskanymi innymi metodami.

Metoda	Liczba reguł	$J_{RMSE}$ testowy
Chung & Duan	20	0.0139
Cho & Wang	23	0.0114
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma iloczyn, implikacja Kleene-Dienes)</b>	4	<b>0.0060</b>
Jang & Sun	16	0.0015
ANNBFIS	15	0.0014

Z powyższego zestawienia wynika, że w zagadnieniu predykcji chaotycznych szeregów czasowych (problem Mackey’a-Glassa), prezentowana w niniejszej rozprawie metoda jest konkurencyjna dla baz wiedzy składających się z małej liczby reguł, natomiast zdolność uogólniania uzyskanych tą drogą systemów spada wraz ze wzrostem liczby reguł, w przeciwieństwie do systemu zaprezentowanego w publikacji [111], natomiast uzyskuje znacznie lepsze wyniki od systemu neuronowo-rozmytego bazującego na systemie opisanym w publikacji [98] oraz od systemów opisanych w [29], [96]. W powyższym zestawieniu na uwagę zasługuje również fakt, że najlepszy system ewolucyjno-rozmyty z parametryzowanymi konkluzjami, uzyskany prezentowaną w niniejszej rozprawie trójetapową metodą posiada bazę wiedzy składającą się jedynie z 4 rozmytych reguł.

## Podsumowanie

Uzyskane wyniki potwierdzają wysoką zdolność opisywanej trójetapowej metody uczenia do tworzenia systemów rozmytych, które cechują się dobrą zdolnością uogólniania posiadanej wiedzy. Metoda uzyskuje zadowalające wyniki dla dowolnej kombinacji implikacji i operatora



$t$ -normy. Rezultaty potwierdzają uniwersalność systemu w przedstawionych zastosowaniach. W rozważanym zagadnieniu system wykazuje porównywalną zdolność uogólniania w stosunku do innych metod, natomiast dla określonych przypadków uzyskuje wyniki znacznie lepsze.

#### 6.2.4 Wycena wartości nieruchomości (Boston Housing)

Przedstawione w niniejszym podrozdziale wyniki dotyczą zagadnienia wyceny nieruchomości na przedmieściach miasta Boston (USA). Baza danych wykorzystana w niniejszym eksperymencie pochodzi z [15] i zawiera 506 rekordów danych, z których każda pozycja opisuje za pomocą 13 parametrów cechy nieruchomości, natomiast ostatnia pozycja odpowiada jej wartości.

Przyjęto równoliczny, losowy podział bazy danych pomiędzy zbiór uczący i zbiór testowy. Zbiór uczący powstał poprzez wylosowanie 253 rekordów z bazy danych Boston Housing, natomiast pozostałe 253 rekordy utworzyły zbiór testowy.

Jako wskaźnik jakości przyjęto błąd średni AVE (ang. Average Error), celem umożliwienia porównania wyników z rezultatami uzyskanymi w pracy [146], oraz wyliczony równolegle z powyższym wskaźnikiem błąd średniokwadratowy (jak w pozostałych eksperymentach).

W celu wygenerowania systemu rozmytego zastosowano trójetapową metodę uczenia, w której ostatni, trzeci etap uczenia był wykonywany trzykrotnie, przy czym w drugim uruchomieniu procesu dostrajania parametrów (trzeciego etapu metody trójetapowej) jako bazę obliczeń przyjęto najlepszy system, uzyskany w uruchomieniu poprzednim. Najlepszy system rozmyty poddawany ocenie stanowił osobnik o najwyższej wartości przystosowania, uzyskany w drugiej iteracji trzeciego etapu trójetapowej metody uczenia (w procesie dostrajania parametrów).

Poniżej zestawiono zbiór parametrów sterujących wykonaniem poszczególnych etapów.

Tabela 6.17: Parametry sterujące trójetapowej metody uczenia dla zagadnienia wyceny nieruchomości (Boston Housing). Uczenie dla zbioru 253 losowo wybranych rekordów.

Etap	Parametr	Symbol	Wartość
1.	Liczebność pokolenia bazowego.	$\mu$	60
	Liczebność pokolenia potomnego.	$\lambda$	20
	Liczba pokoleń.	$\mathcal{L}$	50
	Prawdopodobieństwo krzyżowania.	$P_c$	1
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.8
	Współczynnik rekordów pokrywanych przez regułę do rekordów niepokrywanych (zastosowany w funkcji oceny osobnika).	$\kappa$	0.05
	Minimalny stopień kompatybilności danych z regułą (zastosowany w funkcji oceny osobnika)..	$\omega$	0.1
	Minimalny stopień pokrycia rekordu przez reguły. Różne wartości parametru pozwoliły na uzyskanie różnej liczebności reguł (zastosowany na poziomie algorytmu pokrywającego).	$\varepsilon$	0.4
			0.6
			0.8
			1.5
	Parametr krzyżowania uśredniającego.	$\alpha$	0.35

	Parametr periodycznej mutacji nierównomiernej.	$b$	10
	Poziom funkcji gaussowskich dla wyliczenia szerokości nośnika w funkcjach przynależności przesłanek reguł.	$G_L$	0.25
2.	Liczebność pokolenia bazowego.	$M$	50
	Liczba pokoleń.	$K$	150
	Prawdopodobieństwo mutacji (dla genu)	$P_m$	0.005
	Prawdopodobieństwo krzyżowania.	$P_c$	0.75
	Minimalny wskaźnik pokrycia danych przez bazę reguł (zastosowany w funkcji oceny osobnika).	$\tau$	0.25
3.	Liczebność pokolenia bazowego.	$\mu$	100
	Liczebność pokolenia potomnego .	$\lambda$	20
	Liczba pokoleń dla pojedynczego uruchomienia strategii ewolucyjnej.	$\mathcal{L}$	150
	Prawdopodobieństwo krzyżowania (dla pary osobników)	$P_c$	0.6
	Prawdopodobieństwo mutacji (dla osobnika).	$P_m$	0.6
	Poziom funkcji gaussowskich dla wyliczenia nośnika.	$G_L$	0.9
	Parametr krzyżowania uśredniającego.	$a$	0.25
	Parametr periodycznej mutacji nierównomiernej.	$b$	5

W procesie ustalania parametrów konkluzji metodą uczenia globalnego dla źle uwarunkowanych macierzy ( $D^T D$ ) (równanie 2.115) zastosowano na ich przekątnej głównej korektę równą  $1.15 \cdot 10^{-2}$ .

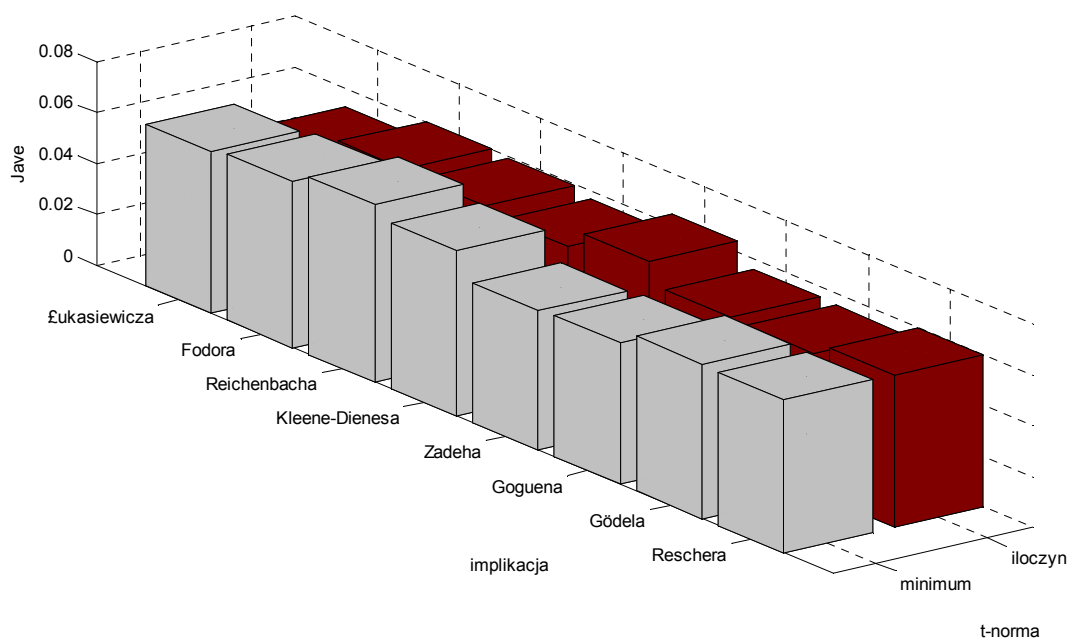
W procesie ewolucyjnym uzyskano systemy rozmyte z parametryzowanymi konkluzjami o bazie wiedzy składającej się odpowiednio z 4, 5, 6 i 14 rozmytych reguł warunkowych **if-then** (w zależności od przyjętego parametru  $\varepsilon$ ). Najlepszą zdolność uogólniania (w sensie błędu AVE) uzyskano dla systemu wykorzystującego logiczną interpretację reguł z implikacją Łukasiewicza,  $t$ -normą iloczyn oraz bazą wiedzy składającą się z 5 reguł. W poniższej tabeli zestawiono najlepsze uzyskane wyniki zdolności uogólniania, w zależności od stosowanej implikacji i operatora  $t$ -normy.

Tabela 6.18: Porównanie uzyskanych wyników uogólniania dla zagadnienia wyceny nieruchomości (Boston Housing). Wyniki dla zbioru uczącego i testującego.

Implikacja	liczba reguł / $t$ -norma	$J_{RMSE}$		$J_{AVE}$
		uczący	Testowy	testowy
Łukasiewicza	4 / minimum	2.9402	4.1311	0.0634
	5 / iloczyn	2.1919	3.4153	<b>0.0526</b>
Fodora	5 / minimum	2.2417	4.6109	0.0653
	5 / iloczyn	2.5580	3.8201	0.0602
Reichenbacha	6 / minimum	2.9645	4.6052	0.0694
	5 / iloczyn	2.2815	3.9176	0.0589
Kleene-Dienes	5 / minimum	2.0819	4.3103	0.0647
	6 / iloczyn	2.3678	4.0580	0.0564
Zadeha	6 / minimum	2.3777	3.4151	0.0546
	6 / iloczyn	2.2667	4.2864	0.0641
Goguena	14 / minimum	2.2492	3.3706	0.0554
	6 / iloczyn	2.4990	3.4119	0.0556
Gödela	5 / minimum	2.8527	4.2082	0.0606

	5 / iloczyn	2.6801	3.5338	0.0550
Reschera	5 / minimum	2.2015	4.2986	0.0600
	4 / iloczyn	2.4761	3.7587	0.0596

Analizując wyniki zawarte w powyższej tabeli można zauważyć, że zarówno aproksymacja danych pomiarowych (dla zbioru uczącego) jak i zdolność uogólniania systemu (dla zbioru testującego) w nieznacznym stopniu zależą od zastosowanej implikacji i operatora  $t$ -normy. Dwa najlepsze wyniki należą do systemów wykorzystujących implikację Łukasiewicza i Zadeha. Na uwagę zasługuje fakt, że poza implikacją Goguena i Zadeha, lepsze wyniki są osiągane dla  $t$ -normy iloczyn niż dla  $t$ -normy minimum. Najlepszy wynik w sensie błędu AVE zaznaczono czcionką pogrubioną.



Rysunek 6.19: Graficzna prezentacja zdolności uogólniania dla zagadnienia wyceny nieruchomości, w podziale na implikacje i  $t$ -normy.

Poniżej przedstawiono szczegółowe wyniki, w podziale na liczbę reguł.

Tabela 6.19: Porównanie uzyskanych wyników  $J_{\text{RMSE}}$  zdolności uogólniania dla zagadnienia wyceny nieruchomości (Boston Housing). Wyniki dla zbioru testującego.

Implikacja	$t$ -norma	Liczba reguł			
		4	5	6	14
Łukasiewicza	minimum	4.1311	5.4152	12.8041	8.0297
	iloczyn	3.8878	<b>3.4153</b>	5.2819	6.6180
Fodora	minimum	5.7180	4.6109	9.9330	5.3231
	iloczyn	3.9832	3.8201	8.5706	5.8167
Reichenbacha	minimum	18.7353	4.7393	4.6052	8.4701
	iloczyn	4.9539	3.9176	6.4183	14.3730
Kleene-Dienes	minimum	4.8215	4.3103	11.2231	11.6845
	iloczyn	4.3210	4.1728	4.5080	13.4609
Zadeha	minimum	4.5285	5.0527	<i>3.4151</i>	9.7834
	iloczyn	4.5618	4.5297	4.2864	6.1501
Goguena	minimum	4.0682	4.1894	4.0727	<b>3.3706</b>
	iloczyn	3.8178	3.8299	<b>3.4119</b>	4.7632
Gödel	minimum	4.5323	4.2082	7.8310	6.7055
	iloczyn	<b>3.7087</b>	3.5338	8.6566	4.0149
Reschera	minimum	4.6131	4.2986	4.4401	36.5539
	iloczyn	<i>3.7587</i>	3.7735	3.7758	3.7800

Z powyższego zestawienia nie wynikają wyraźne zależności jakości systemu od liczby reguł tworzących bazę danych. Na uwagę zasługuje fakt, iż system rozmyty dla implikacji Reschera oraz  $t$ -normy iloczyn uzyskuje podobne wyniki dla wszystkich przebadanych liczebności bazy wiedzy.

Poniżej zestawiono najlepsze uzyskane wyniki z wynikami zaprezentowanymi w pracy [146] i [147].

Tabela 6.20: Porównanie uzyskanych wyników  $J_{\text{AVE}}$  zdolności uogólniania dla zagadnienia wyceny nieruchomości (Boston Housing), z wynikami uzyskanymi innymi metodami.

Metoda	Liczba reguł	$J_{\text{AVE}}$ testowy
FNB (Fuzzy Naïve Bayes)	-	0.0824
LID3 (Linguistic ID3)	$5^3 / 3^4$	0.0820
FSNB (Fuzzy Semi-Naive Bayes)	-	0.0770
$\varepsilon$ -SVR (Support Vector Regression)	10	0.0545
<b>System ewolucyjno-rozmyty z parametryzowanymi konkluzjami (<math>t</math>-norma iloczyn, implikacja Kleene-Dienes)</b>	5	<b>0.0526</b>

<sup>3</sup> Liczba etykiet lingwistycznych.

<sup>4</sup> Liczba poziomów w drzewie decyzyjnym (Linguistic Decision Tree).

Jak widać z powyższego zestawienia, prezentowana w niniejszej rozprawie metoda uzyskuje konkurencyjne wyniki, w stosunku do rezultatów osiąganych innymi metodami.

### Podsumowanie

Przedstawione w powyższych zestawieniach rezultaty praktycznie dowodzą przydatności reprezentowanej metody w przedstawianym zagadnieniu. Uzyskiwanie dobrych wyników niezależnie od zastosowanej implikacji i operatora  $t$ -normy powoduje, że trójetapowa metoda uczenia wymaga jedynie minimalnej wiedzy początkowej na temat rozważanego zagadnienia, tworząc systemy rozmyte z parametryzowanymi konkluzjami o wysokiej zdolności uogólniania oraz bardzo dobrej aproksymacji danych pomiarowych.

## 6.3 Czasy obliczeń i platforma sprzętowa

Obliczenia prowadzono w środowisku MATLAB w wersji 6.5 (jednoprocessorowym), przeznaczonym dla platformy PC. Ponieważ system nie użytkuje intensywnie dużych obszarów pamięci, rozważania ograniczono do zagadnienia czasu obliczeń.

Wykorzystywana wersja środowiska MATLAB nie posiada wsparcia dla systemów wieloprocessorowych oraz technologii HT, dlatego też najistotniejszym czynnikiem mającym wpływ na czas utworzenia systemu z zastosowaniem opisywanej w rozprawie metody ma prędkość pojedynczego potoku procesora. Poniżej zestawiono czasy obliczeń dla wersji opracowanego środowiska zoptymalizowanej pod kątem wydajności obliczeniowej (dla dużych ilości iteracji algorytmów ewolucyjnych oraz liczebności pokoleń przedstawianej w niniejszym rozdziale).

Podane wartości są orientacyjne, ponieważ wszystkie udostępnione komputery pełniły jednocześnie rolę serwerów sieciowych, jak również sporadycznie były wykorzystywane do innych zastosowań.

Do przeprowadzenia obliczeń wykorzystywano 2 komputery:

- Jednostka dwuprocessorowa z dwoma procesorami 2 x Intel XEON MP @ 2.4GHz .
- Jednostka jednoprocessorowa 1x Intel P4 @ 1.6GHz .

Zastosowana w procesorach XEON technologia HT pozwoliła na zrównoleglenie obliczeń na poziomie zestawów parametrów sterujących wykonaniem poszczególnych etapów trójetapowej metody uczenia. Podane czasy dotyczą obliczeń na jednostce o wyższej częstotliwości taktowania procesorów, przy pełnym obciążeniu wszystkich potoków obliczeniowych. Podane czasy dotyczą obliczeń dla pojedynczego eksperymentu w pełnym zakresie badanych implikacji i operatorów  $t$ -normy.

Tabela 6.21: Czasy obliczeń.

Etap	Zagadnienie	Przybliżony czas obliczeń
1	Identyfikacja pieca gazowego, pełny zbiór uczący. 7 reguł.	10.5h
	Identyfikacja pieca gazowego, pełny zbiór uczący. 9 reguł.	13h
	Identyfikacja pieca gazowego, zbiór uczący 100 rekordów. 7 reguł.	5h
	Identyfikacja pieca gazowego, zbiór uczący 100 rekordów. 9 reguł.	5.5h
	Przewidywanie ilości płam na słońcu, zbiór uczący 100 rekordów. 16 reguł.	12h
	Przewidywanie ilości płam na słońcu, zbiór uczący 100 rekordów. 6 reguł.	1.5h
	Predykcja szeregów czasowych, zbiór uczący 500 rekordów, 9 reguł.	7h
	Predykcja szeregów czasowych, zbiór uczący 500 rekordów, 7 reguł.	5.5h
2	Identyfikacja pieca gazowego, pełny zbiór uczący. Redukcja z wykorzystaniem algorytmu genetycznego, początkowa baza reguł o 9 regułach rozmytych.	14h
	Identyfikacja pieca gazowego, zbiór uczący 100 rekordów. Redukcja z wykorzystaniem algorytmu genetycznego, początkowa baza reguł o 9 regułach rozmytych.	7h
	Identyfikacja pieca gazowego, pełny zbiór uczący. Redukcja bazy reguł z wykorzystaniem wyszukiwania wyczerpującego, liczba kombinacji $2^9-1$ .	45min.
3	Identyfikacja pieca gazowego, 3 iteracje trzeciego etapu, pełny zbiór uczący. 7 reguł.	1d21h
	Identyfikacja pieca gazowego, 3 iteracje trzeciego etapu, pełny zbiór uczący. 9 reguł.	2d23h
	Identyfikacja pieca gazowego, 3 iteracje trzeciego etapu, zbiór uczący 100 rekordów. 7 reguł.	10h

	Identyfikacja pieca gazowego, 3 iteracje trzeciego etapu, zbiór uczący 100 rekordów. 9 reguł.	12h
	Przewidywanie ilości plam na słońcu, 3 iteracje trzeciego etapu, zbiór uczący 100 rekordów. 3 reguły.	7h
	Predykcja szeregów czasowych Mackey'a-Glassa, 3 iteracje trzeciego etapu, zbiór uczący 500 rekordów, 9 reguł.	2d7h
	Predykcja szeregów czasowych Mackey'a-Glassa, 3 iteracje trzeciego etapu, zbiór uczący 500 rekordów, 7 reguł.	2d1h

### Podsumowanie

Przedstawiony powyżej czas obliczeń jest znaczny, w szczególności po zsumowaniu łącznego wykonania kroków 1, 2 i 3, jednakże pozwala na uzyskanie wysoce precyzyjnych systemów rozmytych z parametryzowanymi konkluzjami.

Należy zwrócić uwagę na fakt, iż wykorzystywana platforma sprzętowa nie jest dedykowana do skomplikowanych obliczeń, w szczególności wykorzystujących wektory i macierze danych, a parametry szybszej jednostki obliczeniowej wykorzystywanej do pomiaru czasu obliczeń dorównują domowym komputerom klasy PC. Późniejsze wykorzystywanie utworzonego prezentowaną metodą systemu rozmytego z parametryzowanymi konkluzjami, rozumiane jako uzyskanie odpowiedzi systemu dla podanych argumentów wejściowych, nie stawia tak wygórowanych wymagań przed jednostką obliczeniową i odbywa się w czasie poniżej 1s przy założeniu bazy wiedzy składającej się z ok. 10 reguł i 10 wejść systemu klasy MISO, dla jednego rekordu danych wejściowych.





## Podsumowanie i wnioski końcowe

W rozprawie przedstawiono rozwiązanie dla problemu automatycznego wyodrębniania rozmytych reguł warunkowych **if-then** z parametryzowanymi konkluzjami, na podstawie danych numerycznych, z wykorzystaniem metod ewolucyjnych. Rozwiązanie to bazuje na trójetapowej metodzie uczenia, wykorzystującej iteracyjne pokrywanie danych numerycznych, strategię ewolucyjną i algorytm genetyczny. Uzyskane tą drogą systemy rozmyte z parametryzowanymi konkluzjami, o logicznej interpretacji reguł (z wykorzystaniem implikacji rozmytych) cechują się bardzo dobrą jakością (mierzoną błędem średniokwadratowym pomiędzy odpowiedzią systemu i danymi wzorcowymi), potwierdzoną przeprowadzonymi eksperymentami. Zastosowanie rozmytych reguł warunkowych pozwala na łatwą interpretację zawartej w nich wiedzy, a logiczna interpretacja reguł powoduje, że wnioskowanie rozmyte z ich wykorzystaniem pozostaje w zgodzie z logiką klasyczną. Trójetapowa metoda uczenia obok mechanizmu automatycznego pozyskiwania wiedzy na podstawie danych numerycznych pozwala na wprowadzenie wiedzy od eksperta – człowieka. Dodatkowym atutem jest weryfikacja i ewentualna korekcja tej wiedzy na etapie dostrajania parametrów systemu.

W niniejszej rozprawie praktycznie udowodniono przydatność technik ewolucyjnych do uczenia systemu rozmytego z parametryzowanymi konkluzjami poprzez zastosowanie trójetapowej metody uczenia. Zastosowanie obliczeń ewolucyjnych pozwoliło również na zrezygnowanie z precyzyjnego dobierania warunków początkowych dla poszczególnych etapów rozważanego zagadnienia, w przeciwieństwie np. do systemów neuronowo-rozmytych, gdzie parametry początkowe mają istotny wpływ na końcową jakość systemu. Dodatkowym atutem zastosowanego połączenia obliczeń ewolucyjnych z metodami wnioskowania rozmytego jest możliwość skutecznego dostarczania rozwiązań w oparciu o dane niekompletne, nieściśle i nieprecyzyjne.

Do oryginalnych osiągnięć w niniejszej rozprawie można zaliczyć:

- opracowanie metody połączenia technik ewolucyjnych oraz systemów rozmytych z parametryzowanymi konkluzjami w regułach warunkowych **if-then** celem uzyskania uniwersalnej metody, wykazującej dużą skuteczność działania dla szerokiego zakresu zagadnień,
- opracowanie strategii ewolucyjnych i algorytmów genetycznych wraz z dedykowanymi operatorami genetycznymi, pozwalających na uzyskanie systemów rozmytych o wysokiej jakości,
- integrację algorytmu grupowania rozmytego ze strategią ewolucyjną,

- wykazanie w drodze przeprowadzonych badań użyteczności prezentowanej metody,
- wykazanie konkurencyjności prezentowanej metody dla podanych przykładów w zestawieniu z innymi metodami opisywanymi w literaturze.

Zaprezentowana w niniejszej rozprawie trójetapowa metoda automatycznego wyodrębniania rozmytych reguł warunkowych z parametryzowaną konkluzją składa się z trzech autonomicznych procesów. Pierwszy etap pokrywa zbiór numerycznych danych uczących za pomocą iteracyjnego algorytmu, w którym w każdej iteracji jest odnajdywana pojedyncza reguła rozmyta. Proces ten wykorzystuje strategię ewolucyjną. W tym kroku przyjęto model ze zmienną bazą danych systemu rozmytego. Uzyskane reguły mogą być uzupełnione o dane pochodzące od eksperta – człowieka. Możliwe jest również dołączenie takich reguł na etapie generowania pokoleń startowych dla strategii ewolucyjnej, co pozwala na ich automatyczną weryfikację w procesie ewolucji. Drugi etap redukuje uzyskaną bazę wiedzy na poziomie bazy reguł, usuwając reguły redundantne i sprzeczne, przy jednoczesnym zapewnieniu odpowiedniego pokrycia danych uczących. W tym celu wykorzystano algorytm ewolucyjny. Po zakończeniu drugiego etapu również istnieje możliwość uzupełnienia bazy wiedzy o reguły pochodzące od eksperta – człowieka. Etap trzeci tworzy system rozmyty z parametryzowanymi konkluzjami na podstawie wyników uzyskanych w pierwszych dwóch etapach, a następnie dostraja bazę danych tego systemu, wykorzystując w tym celu strategię ewolucyjną.

Zastosowana modularność metody pozwala na niezależne traktowanie poszczególnych etapów, a w szczególności na wielokrotne, łańcuchowe uruchomienie operacji dostrajania, gdzie najlepszy system uzyskany w poprzedzającym uruchomieniu stanowi bazę do dalszych obliczeń w uruchomieniu kolejnym, celem uzyskania systemu o jeszcze lepszych parametrach.

Uzyskanie wysokiej jakości systemów za pomocą prezentowanej metody jest uzależnione od poprawnego doboru parametrów poszczególnych procesów wchodzących w skład trójetapowej metody uczenia, w szczególności związanych z przebiegiem procesu ewolucyjnego. Wartości te nie są jednak krytyczne. Podane w rozprawie wartości parametrów dobrano za pomocą serii eksperymentów, bez wykorzystania metod analitycznych, mając jednak na uwadze osiągnięte rezultaty w zastosowaniach praktycznych, można przypuszczać, iż są one bliskie optimum. Szczególnie istotnym elementem jest zapewnienie zbieżności algorytmu genetycznego i strategii ewolucyjnych, jak również osiągnięcie zadowalających rezultatów w skończonym i jak najkrótszym czasie, mierzonym liczbą iteracji. Z tego powodu do prezentowanego systemu ewolucyjno-rozmytego została dołączona na etapie generowania pokoleń startowych metoda grupowania rozmytego FCM.

Eksperymenty praktyczne zostały przeprowadzone przy minimalnej zmienności parametrów sterujących metodą, głównie związanych z ilością rozmytych reguł warunkowych **if-then**, celem umożliwienia wiarygodnego porównania uzyskanych rezultatów z rezultatami prezentowanymi w literaturze. Oznacza to, że prezentowana metoda jest uniwersalna i nadaje się do zastosowania w szerokim wachlarzu zagadnień oraz nie wymaga prawie żadnej wiedzy o modelowanym zjawisku.

Osiągnięte wyniki w wielu zagadnieniach (w szczególności w zagadnieniu identyfikacji systemów) klasyfikują prezentowaną metodę wśród najlepszych dotychczas opracowanych.

Mając na uwadze powyższe informacje, wydaje się, że postawione w rozprawie cele częściowe zostały osiągnięte, tj.:

- Opracowano strategię ewolucyjną wraz z dedykowanymi operatorami genetycznymi i wykorzystano ją do iteracyjnego pozyskiwania reguł warunkowych z parametryzowanymi konkluzjami metodą pokrywania zbioru uczącego, z zastosowaniem kodowania Michigan.
- Opracowano algorytm genetyczny wraz z operatorami genetycznymi do uproszczenia zbioru reguł uzyskanych w wyniku działania algorytmu pokrywania.
- Opracowano strategię ewolucyjną wraz z dedykowanymi operatorami genetycznymi, o kodowaniu Pittsburgh, a następnie zastosowano ją w procesie dostrajania parametrów reguł uzyskanych w poprzednich krokach metody.
- Przeprowadzono praktyczne eksperymenty w zaimplementowanym środowisku obliczeniowym uzyskując bardzo dobre wyniki, konkurencyjne względem wyników uzyskiwanych innymi metodami.

Mając na uwadze spełnienie powyższych celów częściowych, nie stwierdzono podstaw do odrzucenia tezy postawionej we wstępie niniejszej rozprawy doktorskiej:

**Możliwe jest poprawienie jakości systemu rozmytego z regułami if-then o parametryzowanych konkluzjach, poprzez wyodrębnianie ich na podstawie danych numerycznych, za pomocą wieloetapowej metody uczenia, z wykorzystaniem technik ewolucyjnych.**

Prezentowana praca nie wyczerpuje z pewnością wszystkich możliwości dalszej poprawy jakości systemów rozmytych. Dlatego też rozważana tematyka pozostawia miejsce do dalszych badań. Wydaje się, że najbardziej obiecującymi kierunkami badań nad jakością systemów rozmytych i ich automatycznym pozyskiwaniem na podstawie danych numerycznych z wykorzystaniem technik ewolucyjnych są:

- Integracja wszystkich technik obliczeń miękkich, w szczególności połączenie w jednym systemie sztucznych sieci neuronowych i systemów ewolucyjno-rozmytych.
- Wprowadzenie ewolucyjnego,  $\varepsilon$ -nieczułego modelowania rozmytego.
- Przeprowadzenie obliczeń ewolucyjnych na większej liczbie parametrów systemu rozmytego, a w szczególności wykorzystanie systemów o zmiennych funkcjach przynależności w przesłankach i konkluzjach oraz systemów, w których parametry konkluzji będą podlegać ewolucyjnej optymalizacji.
- Opracowanie precyzyjniejszej metody dobierania parametrów konkluzji dla systemów rozmytych z parametryzowaną konkluzją.

- Wykorzystanie obliczeń ewolucyjnych do automatycznego wyboru modelu systemu rozmytego.
- Poprawa zbieżności algorytmów genetycznych i strategii ewolucyjnych poprzez deterministyczne lub niedeterministyczne dobieranie parametrów sterujących przebiegiem procesów ewolucyjnych.

Zaprezentowane w niniejszej rozprawie rozwiązania wykazują duże możliwości aplikacyjne. W szczególności mogą one objąć rzeczywiste zagadnienia m.in. z dziedziny medycyny, a wraz ze wzrostem możliwości obliczeniowych współczesnych komputerów mogą znaleźć zastosowanie do tworzenia inteligentnych aplikacji, wspomagających systemy decyzyjne czy nawet podejmujących autonomiczne decyzje, np. w zagadnieniu prognozowania zapotrzebowania na usługi medyczne [45]. Możliwości te znacznie rozszerzają tematykę dalszych badań.

## Literatura

- [1] Alba, E., Cotta, C., Troya, J.M., *Evolutionary Design of Fuzzy Logic Controllers Using Strongly-typed GP*. Mathware Soft Comput., 6 (1), 1999, 109–124.
- [2] Alcalá, R., Benitez, J. M., Casillas, J., Cordon, O., Pérez, R., *Fuzzy Control of HVAC Systems Optimized by Genetic Algorithms*. Applied Intelligence, 18, 2003, 155-177.
- [3] Alcalá, R., Casillas, J., Cordon, O., Herrera, F., *Linguistic Modeling with Weighted Double-Consequent Fuzzy Rules Based on Cooperative Coevolutionary Learning*. Raport nr DECSAI-010114, Wydział Informatyki i Sztucznej Inteligencji, University of Granada, 2001.
- [4] Angelov, P. P., *Evolving Rule-Based Models. A Tool for Design of Flexible Adaptive Systems*. Physica-Verlag, Wurzburg, 2002.
- [5] Arabas, J., *Wykłady z algorytmów ewolucyjnych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2001.
- [6] Babuska, R., *Fuzzy Modeling for Control*. Kluwer Academic Press, Dordrecht, 1998.
- [7] Baldwin, J. F., *A New Approach to Approximate Reasoning Using a Fuzzy Logic*. Fuzzy Sets and Systems, 2, 1979, 309-325.
- [8] Baldwin, J. F., Guild, N. C. F., *Feasible Algorithms For Approximate Reasoning Using Fuzzy Logic*. Fuzzy Sets and Systems, 3, 1980, 225-251.
- [9] Baldwin, J. F., Guild, N. C. F., *Comparison of Fuzzy Sets on the Same Decision Space*. Fuzzy Sets and Systems, 2, 1979, 213-231.
- [10] Baldwin, J. F., Pilsworth, B., W., *Axiomatic Approach to Implication for Approximate Reasoning with Fuzzy Logic*. Fuzzy Sets and Systems, 3, 1980, 193-219.
- [11] Baron, L., Achiche, S., Balazinski, M., *Fuzzy Decision Support System Knowledge Base Generation Using a GeneticAlgorithm*. International Journal of Approximate Reasoning, 28 (1), 2001, 125–148.
- [12] Bertoni, A., Dorigo, M., *Implicit Parallelism in Genetic Algorithms*. Artificial Intelligence, 61, 1993, 308-314.
- [13] Bethke, A. D., *Genetic Algorithms as Function Optimizers*. Rozprawa doktorska, Uniwersytet Michigan, 1981.
- [14] Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [15] Blake, C., Merz, C. J., *UCI machine learning repository*. Materiały on-line w sieci Internet: <http://www.ics.uci.edu/~mllearn/MLRepository.html> .

- [16] Bonari, A., *Evolutionary Learning of Fuzzy Rules: Competition and Cooperation*. W: W. Pedrycz Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Press, Norwell, 1996, 265–284.
- [17] Bonissone, P. P., Khedkar P. S., Chen, Y., *Genetic Algorithms for Automated Tuning of Fuzzy Constollers: A Transportation Application*. W: Proc. of the 5<sup>th</sup> IEEE International Conference on Fuzzy Systems, New Orleans, USA, 1996, 674-680.
- [18] Booch, G., Jacobson, I., Rumbaugh, J., *OMG Unified Modeling Language Specification, Version 1.4*. Materiały on-line w sieci Internet: <http://www.omg.org>, 2001.
- [19] Borwein, J. M., Borwein, P. B., *Pi & the AGM: A Study in Analytic Number Theory and Computational Complexity*. Wiley, New York, 1987.
- [20] Box, G. E. P., Jenkins, G. M., *Time Series Analysis. Forecasting and Control*. Holden-Day, San Francisco, 1976.
- [21] Brindle, A., *Genetic Algorithms for Function Optimization*. Rozprawa doktorska, University of Alberta, 1981.
- [22] Buckles, B.P., Petry, F.E., Prabhu, D., George, R., Srikanth, R., *Fuzzy Clustering with Genetic Search*. W: Proc. of the 1<sup>st</sup> IEEE Conference on Evolutionary Computation, Orlando, USA, 1994, 46–50.
- [23] Bullen, P. S., *Handbook of Means and Their Inequalities*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.
- [24] Carse, B., Fogarty, T.C., Munro, A. *Evolving Fuzzy Rule Based Controllers Using Genetic Algorithms*. Fuzzy Sets and Systems, 80, 1996, 273–294.
- [25] Casillas, J., Cordon, O., Del Jesus, M. J., Herrera, F., *Genetic Feature Selection in a Fuzzy Rule-Based Classification System Learning Process for High Dimensional Problems*. Inform. Sci., 136, 2001, 169–191.
- [26] Chen, J. Q., Xi, Y., G., Zhang, Z., J., *A Clustering Algorithm for Fuzzy Model Identification*. Fuzzy Sets and Systems, 98, 1998, 319-329.
- [27] Cheong, F., Lai, R., *Constraining the Optimization of a Fuzzy Logic Controller Using an Enhanced Genetic Algorithm*. IEEE Trans. Systems Man Cybernet., B 30, 2000, 31–46.
- [28] Chi, Z., Yan, H., Pham T., *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. World Scientific, Singapore, 1996.
- [29] Cho, K. B., Wang., B. H., *Radial Basis Function Based Adaptive Fuzzy Systems and Their Applications to Systems Identification and Prediction*. Fuzzy Sets and Systems, 83, 1996, 325-339.
- [30] Chien, B.-C., Lin, J. Y., Hong, T.-P., *Learning Discriminant Functions with Fuzzy Attributes for Classification Using Genetic Programming*. Expert Systems Appl., 23 (1), 2002, 31–37.
- [31] Chiu, S. L., *Selecting Input Variables for Fuzzy Models*. Journal of Intelligent & Fuzzy Systems, 4 (4), 1996, 243-256.
- [32] Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L., *Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends*. Fuzzy Sets and Systems, 141, 2004, 5-31.

- [33] Cordon, O., Herrera, F., *A Proposal for Improving the Accuracy of Linguistic Modeling*. IEEE Trans. Fuzzy Systems, 8 (3), 2000, 335–344.
- [34] Cordon, O., Herrera, F., *Identification of Linguistic Fuzzy Models by Means of Genetic Algorithms*. Springer-Verlag, 1997, 215-250.
- [35] Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L., *Genetic Fuzzy Systems-Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. World Scientific, Singapore, 2001.
- [36] Cordon, O., Herrera, F., *A Three-stage Evolutionary Process for Learning Descriptive and Approximative Fuzzy Logic Controller Knowledge Bases from Examples*. International Journal of Approximate Reasoning, 17 (4), 1997, 369-407.
- [37] Cordon, O., Herrera, F., *A Two-stage Evolutionary Process for Designing TSK Fuzzy Rule-based Systems*. IEEE Trans. Systems Man Cybernet., 29 (6), 1999, 703–715.
- [38] Cordon, O., Herrera, F., *Hybridizing Genetic Algorithms with Sharing Scheme and Evolution Strategies for Designing Approximate Fuzzy Rule-based Systems*. Fuzzy Sets and Systems, 118 (2), 2001, 235–255.
- [39] Cordon, O., Herrera, F., Magdalena, L., Villar, P., *A Genetic Learning Process for the Scaling Factors, Granularity and Contexts of the Fuzzy Rule-Based System Data Base*. Inform. Sci., 136 (1–4), 2001, 85–107.
- [40] Cordon, O., Herrera, F., Villar, P., *Generating the Knowledge Base of a Fuzzy Rule-based System by the Genetic Learning of the Data Base*. IEEE Trans. Fuzzy Systems, 9 (4), 2001, 667–674.
- [41] Cordon, O., Herrera, F., Zwir, I., *Linguistic Modeling by Hierarchical Systems of Linguistic Rules*. IEEE Trans. Fuzzy Systems, 10 (1), 2002, 2–20.
- [42] Cordon, O., Del Jesús, M. J., Herrera, F., Lozano, M., *MOGUL: a Methodology to Obtain Genetic Fuzzy Rule-based Systems Under the Iterative Rule Learning Approach*. International Journal of Intelligent Systems, 14 (11), 1999, 1123–1153.
- [43] Czabański, R., *Automatyczne wyznaczanie reguł rozmytych jeżeli-to na podstawie danych numerycznych*. Rozprawa doktorska, Politechnika Śląska w Gliwicach, 2002.
- [44] Czekalski, P., *Automated User's Role Generation from UML Use Case Diagrams*, Studia Informatica Vol. 24., 3 (55), 2003, 55-63.
- [45] Czekalski, P., *Requirement Prognosis for Medical Services*, Studia Informatica Vol. 24., 3 (55), 2003, 77-91.
- [46] Czogała, E., Łęski, J., *A New Fuzzy Inference System with Moving Consequents in If-then Rules. Application to Pattern Recognition*. Bulletin of the Polish Academy of Science, 45 (4), 1996, 643-655.
- [47] Czogała, E., Fodor, J., Łęski, J., *On Modeling Destructive Logical Type Systems Using Fodor Fuzzy Implication*. Proc. 5<sup>th</sup> European Congress Intelligent Techniques and Soft Computing EUFIT, Aachen, 1997, 340-343.
- [48] Czogała, E., Łęski, J., *Fuzzy and Neuro-fuzzy Intelligent Systems*. Physica-Verlag, A Springer-Verlag Company, Heidelberg, New York, 1999.

- [49] Czogała, E., Łęski, J., *On Equivalence of Approximate Reasoning Results Using Different Interpretations of Fuzzy If-then Rules*. Fuzzy Sets and Systems, 117, 2001, 279-296.
- [50] Czogała, E., Pedrycz W., *Elementy i metody teorii zbiorów rozmytych*. Wydawnictwo Politechniki Śląskiej, Gliwice, 1983.
- [51] Damousis, L. G., Dokopoulos, P., *A Fuzzy Expert System for the Forecasting of Wind Speed and Power Generation in Wind Farms*. W: Proc. of the 22<sup>nd</sup> International Conference on Power Industry Computer Applications, 2001, Sydney, 63-69.
- [52] Davis, L., *Adapting Operator Probabilities in Genetic Algorithms*. W: Proc. of the 3<sup>rd</sup> International Conference on Genetic Algorithms, 1989, 61-69.
- [53] De Jong, K. A., *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Rozprawa doktorska, University of Michigan, 1975.
- [54] De Jong, K. A., *Learning with Genetic Algorithms: an Overview*. Mach. Learning, 3 (3), 1988, 121-138.
- [55] Dianati, M., Song, I., Treiber, M., *An Introduction to Genetic Algorithms and Evolution Strategies*. Materiały własne University of Waterloo, Ontario, Canada, 2002.
- [56] Driankov, D., Hellendoorn, H., Reinfrank, M., *Wprowadzenie do sterowania rozmytego*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1996.
- [57] Eiben, A. E., Aarts, E. H. L., Van Hee, K. M., *Global Convergence of Genetic Algorithms: An Infinite Markov Chain Analysis*. W: Proc. of the 1<sup>st</sup> International Conference on Parallel Problem Solving from Nature, Springer, Berlin, 1991, 4-12.
- [58] Fodor, J. C., *Contrapositive Symmetry of Fuzzy Implications*. Fuzzy Sets and Systems, 69, 1995, 141-156.
- [59] Fodor, J. C., *On Fuzzy Implication Operators*. Fuzzy Sets and Systems, 42, 1991, 293-300.
- [60] Fodor, J. C., *Fuzzy Implications*. Proc. of the International Conference on Soft and Intelligent Computing, Budapest, 1996, 91-98.
- [61] Fodor, J. C., *Fuzzy Preference Modeling and Multicriteria Decision Support*. Kluwer Academic, Dordrecht, 1994.
- [62] Fogel, D. B., *Evolving Artificial Intelligence*. Rozprawa doktorska, University of California, San Diego, 1992.
- [63] Fogel, D. B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, New Jersey, 1995.
- [64] Fogel, D. B., Atmar, W., *Hierarchic Methods of Evolutionary Programming*. W: Proc. of the 1<sup>st</sup> International Conference on Evolutionary Programming, Evolution Programming Society, Los Angeles, 1992, 175-182.
- [65] Fogel, D. B., Sebald, A. V., *Steps Toward Controlling Blood Pressure During Surgery using Evolutionary Programming*. W: Proc. of 4<sup>th</sup> Annual Conference on Evolutionary Programming, MIT Press, 1995, 69-82.



- [66] Fogel, L. J., Owens, A. J., Walsh, M. J.,: *Artificial Intelligence Through Simulated Evolution*. John Wiley & Sons, 1966.
- [67] Fowler, M., Scott, K., *UML w kropelce*. LTP, Warszawa, 2002.
- [68] Fuller, R., *Introduction to Neuro-Fuzzy Systems*. Physica-Verlag, Wurzburg, 1999.
- [69] Geyer-Schulz, A. *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning*. Physica-Verlag, Heidelberg, 1995.
- [70] Goldberg, D. E., *Algorytmy genetyczne i ich zastosowania*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2003.
- [71] Goldberg, D. E., Korb, B., Deb, K., *Messy Genetic Algorithms: Motivation, Analysis, and First Results*. Complex Systems, 3, 1989, 493-530.
- [72] Goldberg, D. E., Segrest, P., *Finite Markov Chain Analysis of Genetic Algorithms*. W: Proc. of the 2<sup>nd</sup> International Conference on Genetic Algorithms, 1993, 1-8.
- [73] González, A., Pérez, R., *SLAVE: a Genetic Learning System Based on an Iterative Approach*. IEEE Trans. Fuzzy Systems, 7 (2), 1999, 176–191.
- [74] González, A., Pérez, R., *Selection of Relevant Features in a Fuzzy Genetic Learning Algorithm*. IEEE Trans. System Man Cybernet., B 31 (3), 2001, 417–425.
- [75] Gómez-Skarmeta, A.F., Jiménez, F., *Fuzzy Modeling with Hybrid Systems*. Fuzzy Sets and Systems, 104, 1999, 199–208.
- [76] Hall, L.O., Bezdek, J.C., Boggavarapu, S., Bensaid, A., *Genetic Fuzzy Clustering*. W: Proc. NAFIPS'94, San Antonio, 1994, 411–415.
- [77] Harel, D., *Rzecz o istocie informatyki. Algorytmika*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2000.
- [78] Herrera, F., Lozano, M., Verdegay, J. L., *A Learning Process for Fuzzy Control Rules Using Genetic Algorithms*. Fuzzy Sets and Systems, 100, 1998, 143-158.
- [79] Herrera, F., Lozano, M., Verdegay, J. L., *Generating Fuzzy Rules from Examples using Genetic Algorithms*. Fuzzy Logic and Soft Computing, World Scientific, 1995.
- [80] Herrera, F., Lozano, M., Verdegay, J. L., *Tuning Fuzzy Logic Controllers by Genetic Algorithms*. International Journal of Approximate Reasoning, 12, 1995, 299-315.
- [81] Herrera, F., Verdegay, J. L., *Genetic Algorithms and Soft Computing*. Physica-Verlag, Wurzburg, 1996.
- [82] Hoffmann, F., Nelles, O., *Genetic Programming for Model Selection of TSK-Fuzzy Systems*. Inform. Sci., 136 (1–4), 2001, 7–28.
- [83] Hoffmann, F., Pfister, G., *Evolutionary Design of a Fuzzy Knowledge Base for a Mobile Robot*. International Journal of Approximate Reasoning 17 (4), 1997, 447–469.
- [84] Hoffmeister, F., Bäck, T., *Genetic Algorithms and Evolution Strategies: Similarities and Differences*. W: Proc. of the 1<sup>th</sup> International Conference on Parallel Problem Solving from Nature, Springer, Berlin, 1991, 455-469.

- [85] Holland, J. H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [86] Holland, J. H., Reitman J. S., *Cognitive Systems Based on Adaptive Algorithms*. W: Waterman D. A., Hayes-Roth, F., *Pattern-Directed Inference Systems*. Academic Press, New York, 1978, 313-329.
- [87] Hong, T.-P., Chen, J.-B., *Finding Relevant Attributes and Membership Functions*, *Fuzzy Sets and Systems*, 103 (3), 1999, 389–404.
- [88] Hwang, H. S., *Control Strategy for Optimal Compromise Between Trip Time and Energy Consumption in a High-speed Railway*. *IEEE Trans. Systems, Man and Cybernetics*, 28 (6), 1998, 791-802.
- [89] Ishibuchi, H., Fujioka, R., Tanaka, H., *Neural Network That Learn From Fuzzy If-then Rules*. *IEEE Trans. Fuzzy Systems*, 1(2), 1993, 85-97.
- [90] Ishibuchi, H., Murata, T., Türksen, I. B., *Single-objective and Two-objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems*. *Fuzzy Sets and Systems*, 89, 1997, 135–150.
- [91] Ishibuchi, H., Nakashima, T., *Effect of Rule Weights in Fuzzy Rule-based Classification Systems*. *IEEE Trans. Fuzzy Systems*, 9 (4), 2001, 506–515.
- [92] Ishibuchi, H., Nakashima, T., Murata, T., *Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems*. *IEEE Trans. Systems, Man and Cybernetics*, 29, 1999, 601–618.
- [93] Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H., *Selecting Fuzzy If-then Rules for Classification Problems Using Genetic Algorithms*. *IEEE Trans. Fuzzy Systems*, 3 (3), 1995, 260–270.
- [94] Jang, J.-S. R., *Self-learning Fuzzy Controllers Based on Temporal Back Propagation*. *IEEE Trans. Neural Networks*, 3 (5), 1992, 714-723.
- [95] Jang, J.-S. R., Sun, C.-T., *Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems*. *IEEE Trans. Neural Networks*, 4 (1), 1993, 156-159.
- [96] Jang, J.-S. R., Sun, C.-T., *Neuro-fuzzy modeling and control*. *Proc of IEEE*, 83 (3), 1995, 378-406.
- [97] Jang, J.-S. R., Sun, C.-T., Mizutani, E., *Neuro-Fuzzy and Soft Computing*. Prentice-Hall, Englewood Cliffs, 1997.
- [98] Juuso, R., Koivisto, H., *Optimised Subtractive Clustering for Neuro-Fuzzy Models*. 3<sup>rd</sup> WSES Int. Conference on Fuzzy Systems (FSFS'02), Interlaken, 2002.
- [99] Kim, E., Park, M., Ji, S., *A new approach to fuzzy modeling*. *IEEE Trans. Fuzzy Systems*, 5 (3), 1997, 328-337.
- [100] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., *Optimization by Simulated Annealing*. *Science*, 220 (4598), 1983, 671-680.

- [101] Kosko, B., *Fuzzy associative memories*. W: Kandel, A., *Fuzzy expert systems*, Addison-Wesley, 1987.
- [102] Koza, J. R., *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*. Raport nr STAN-CS-90-134, Stanford University, 1990.
- [103] Koza, J. R., *Evolving a Computer Program to Generate Random Numbers Using Genetic Programming Paradigm*. W: Proc. of the 4<sup>th</sup> International Conference of Genetic Algorithms, Morgan Kaufmann Publishers, 1991, 37-44.
- [104] Koza, J. R., *Genetic Programming*. MIT Press, Cambridge, 1992.
- [105] Koza, J. R., *Genetic Programming – 2*. MIT Press, Cambridge, 1994.
- [106] Lee, H.-M., Chen, C.-M., Chen, J.-M., Jou, Y.-L., *An Efficient Fuzzy Classifier with Feature Selection Based on Fuzzy Entropy*. IEEE Trans. Systems Man Cybernet., B 31 (3), 2001, 426–432.
- [107] Lee, M. A., Takagi, H., *Integrating Design Stages of Fuzzy Systems Using Genetic Algorithms*. W: Proc. of the 2<sup>nd</sup> IEEE International Conference on Fuzzy Systems, San Francisco, 1993, 613-617.
- [108] Lin, Y. Cunningham, H., A., *A new approach to fuzzy-neural modeling*. IEEE Trans. Fuzzy Systems, 3, 1995, 190-197.
- [109] Liska, J., Melsheimer, S. S., *Complete Design of Fuzzy Logic System Using Genetic Algorithms*. W: Proc. of the 3<sup>rd</sup> IEEE International Conference on Fuzzy Systems, 2, Orlando, 1994, 1377-1382.
- [110] Łęski, J., Materiały on-line w sieci Internet: <http://zeb.iele.polsl.gliwice.pl>.
- [111] Łęski, J., *Systemy neuronowo-rozmyte*. Wydawnictwa Naukowo-Techniczne, Warszawa, złożone w redakcji.
- [112] Łęski, J., *TSK-Fuzzy Modeling Based on  $\epsilon$ -Insensitive Learning*. IEEE Trans. Fuzzy Systems, w druku.
- [113] Łęski, J., Czogała, E., *A New Artificial Neural Network Based Fuzzy Inference System with Moving Consequents in If-then Rules and Selected Applications*. Fuzzy Sets and Systems, 108, 1999, 289-297.
- [114] Łukasiewicz, J., *Elementy logiki matematycznej*. Państwowe Wydawnictwo Naukowe, Warszawa, 1958.
- [115] Magdalena, L., Monasterio, F., *A Fuzzy Logic Controller with Learning Through the Evolution of its Knowledge Base*. International Journal of Approximate Reasoning, 16 (3-4), 1997, 335-358.
- [116] Mamdani, E. H., Assilian, S., *An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller*. International Journal of Man-Machine Studies, 7 (1), 1975, 1-13.
- [117] MathWorks, *MATLAB Classes and Objects*. Materiały on-line w sieci Internet: [http://www.mathworks.com/access/helpdesk/help/techdoc/matlab\\_prog/matlab\\_prog.html](http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_prog/matlab_prog.html)
- [118] McCulloch, W. S., Pitts, W., *A Logical Calculus of the Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics, 5, 1943, 115-133.

- [119] Michalewicz, Z., *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2003.
- [120] Michalewicz, Z., Janikow, C., *Handling Constraints in Genetic Algorithms*. W: Proc. of the 4<sup>th</sup> International Conference on Genetic Algorithms, 1991, 151-157.
- [121] Michalewicz, Z., Janikow, C., *GENOCOP: A Genetic Algorithm for Numerical Optimization Problems with Linear Constraints*. Communications of the ACM, 39 (12es), 1996, 175-201.
- [122] Michalewicz, Z., Fogel, D. B., *How to Solve It: Modern Heuristic*. Springer-Verlag, 1999.
- [123] Mitra, S., Pal, S. K., *Fuzzy Multi-layer Perceptron, Inferencing and Rule Generation*. IEEE Trans. Neural Networks, 6, 1995, 51-63.
- [124] Muller, R. J., *Bazy danych język UML w modelowaniu danych*. MIKOM, Warszawa, 2000.
- [125] Nascimientto, S., Moura-Pires, F., *A Genetic Approach for Fuzzy Clustering with a Validity Measure Function*. W: Proc. of the 2<sup>nd</sup> International Conference on Intelligent Data Analysis, Springer-Verlag, Berlin, 1997, 325-335.
- [126] Nauck, D., Klawoon, F., Kruse, R., *Foundations of Neuro-Fuzzy Systems*. John Wiley & Sons, New York, 1997.
- [127] Papadakis, S.E., Theocharis, J.B., *A GA-based Fuzzy Modeling Approach for Generating TSK Models*. Fuzzy Sets and Systems, 131 (1), 2002, 121-152.
- [128] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1984.
- [129] Park, D., Kandel, A., Langholz, G., *Genetic-based New Fuzzy Reasoning Models with Application to Fuzzy Control*. IEEE Trans. Systems Man Cybernet., 24 (1), 1994, 39-47.
- [130] Parodi, A., Bonelli, P., *A New Approach to Fuzzy Classifier Systems*, W: Proceedings of 5<sup>th</sup> International Conference on Genetic Algorithms, Morgan Kaufmann, Los Altos, 1993, 223-230.
- [131] Pedersen, J. T., Moulton, J., *Genetic Algorithms for Protein Structure Prediction*. Current Opinion in Structural Biology, 6, 1996, 227-231.
- [132] Pedrycz, W., *An Identification Algorithm in Fuzzy Relational Systems*. Fuzzy Sets and Systems, 13, 1984, 153-167.
- [133] Pedrycz, W., *Fuzzy Evolutionary Computation*. Kluwer Academic Publishers, Dordrecht, 1997.
- [134] Pedrycz, W., *Fuzzy Modelling: Paradigms and Practices*. Kluwer Academic Publishers, Boston, 1996.
- [135] Pedrycz, W., *Identification in fuzzy systems*. IEEE Trans. Systems, Man and Cybernetics, 14, 1984, 361-366.
- [136] Peña-Reyes, C.A., Sipper, M., *Fuzzy CoCo: A Cooperative Coevolutionary Approach to Fuzzy Modeling*. IEEE Trans. Fuzzy Systems, 9 (5), 2001, 727-737.
- [137] Pham, D. T., Karaboga, D., *Optimum Design of Fuzzy Logic Controllers Using Genetic Algorithms*. Journal of Systems Engineering, 1 (2), 1991, 114-118.

- [138] Piegat, A., *Modelowanie i sterowanie rozmyte*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 1999.
- [139] Pochopień, B., *Arytmetyka systemów cyfrowych*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2002.
- [140] Praca zbiorowa, *Nowa encyklopedia powszechna PWN, tom 1*. WNT, Warszawa, 1995.
- [141] Praca zbiorowa, *Nowa encyklopedia powszechna PWN, tom 2*. WNT, Warszawa, 1995.
- [142] Praca zbiorowa, *Nowa encyklopedia powszechna PWN, tom 3*. WNT, Warszawa, 1995.
- [143] Praca zbiorowa, *Nowa encyklopedia powszechna PWN, tom 4*. WNT, Warszawa, 1995.
- [144] Praca zbiorowa, *Nowa encyklopedia powszechna PWN, tom 5*. WNT, Warszawa, 1995.
- [145] Praca zbiorowa, *Poradnik inżyniera-matematyka, tom 1*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1986.
- [146] Qin, Z., Lawry, J., Randon, N., *Prediction and Query Evaluation Using Linguistic Decision Trees*. International Journal of Approximate Reasoning, złożone w redakcji, 2004.
- [147] Randon, N. J., *Fuzzy and Random Set Based Induction Algorithms*. Rozprawa doktorska, Dept. Of Engineering Mathematics, University of Bristol, 2004.
- [148] Rechenberg I.: *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [149] Rojas, I., Gonzalez, J., Pomares, H., Rojas, F.J., Fernández, F.J., Prieto, A., *Multidimensional and Multideme Genetic Algorithms for the Construction of Fuzzy Systems*. International Journal of Approximate Reasoning, 26 (3), 2001, 179–210.
- [150] Rose, K., *Deterministic Annealing, Clustering and Optimization*. Praca doktorska, California Institute of Technology, Pasadena, 1991.
- [151] Rose, K., *Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems*. Proc. IEEE, 86 (11), 1998, 2210–2239.
- [152] Roubos, H., Setnes, M., *Compact and Transparent Fuzzy Models Through Iterative Complexity Reduction*. IEEE Trans. Fuzzy Systems, 9 (4), 2001, 515–524.
- [153] Rudolph, G., *Convergence Analysis of Canonical Genetic Algorithms*. IEEE Trans. Neural Networks, wydanie specjalne poświęcone obliczeniom ewolucyjnym, 5 (1), 1994, 96–101.
- [154] Rutkowska, D., Piliński, M., Rutkowski L., *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Wydawnictwo Naukowe PWN, Warszawa, 1997.
- [155] Sánchez, L., Corrales, J.A., *Niching Scheme for Steady State GA-P and its Application to Fuzzy Rule Base Classifiers Induction*. Mathware Soft Comput., 7 (2–3), 2000, 337–350.
- [156] Sánchez, L., Couso, I., Corrales, J.A., *Combining GP Operators with SA Search to Evolve Fuzzy Rule-Based Classifiers*. Inform. Sci., 136 (1–4), 2001, 175–191.
- [157] Sanchez, E., Shibata, T., Zadeh, L., *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*. World Scientific, Singapore, 1997.

- [158] Satyadas, A., KrishnaKumar, K., *EMF-based Controllers for Space Attitude Control: Applications and Analysis*. W: Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996, 152-171.
- [159] Schuster, H. G., *Deterministic chaos*. Wydanie drugie, VCH Verlagsgesellschaft, New York, 1988.
- [160] Schwefel, H. P., *Numerical Optimization for Computer Models*. John Wiley, Chichester, 1981.
- [161] Silipo, R., Berthold, M. R., *Input Features' Impact on Fuzzy Decision Processes*. IEEE Trans. Systems, Man Cybernetics, B 30 (6), 2000, 821-834.
- [162] Smith, S. F., *A Learning System Based on Genetic Adaptive Algorithms*. Rozprawa doktorska, Wydział Informatyki uniwersytetu w Pittsburgh, 1980.
- [163] Sugeno, M., Kang, G. T., *Structure Identification of Fuzzy Model*. Fuzzy Sets and Systems, 28, 1988, 15-33.
- [164] Sugeno, M., Yasukawa, T., *A Fuzzy-Logic Based Approach to Qualitative Modeling*. IEEE Trans. Fuzzy Systems, 1 (1), 1993, 7-31.
- [165] Szałas, A., Michalewicz, Z., *Contractive Mapping Genetic Algorithms and Their Convergence*. Raport nr 006-1993, The University of North Carolina, Charlotte, 1993.
- [166] Tadeusiewicz, R., *Elementarne wprowadzenie do sieci neuronowych z przykładowymi programami*. Akademicka Oficyna Wydawnicza, 1998.
- [167] Tadeusiewicz, R., Flasiński, M., *Rozpoznawanie obrazów*. PWN, Warszawa, 1991.
- [168] Tadeusiewicz, R., *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, Warszawa, 1993.
- [169] Takagi, T., Sugeno, M., *Fuzzy Identification of Systems and its Application to Modeling and Control*. IEEE Trans. Systems, Man and Cybernetics, 15 (1), 1985, 116-132.
- [170] Thrift, P., *Fuzzy Logic Synthesis with Genetic Algorithms*. W: Proc. of the 4<sup>th</sup> International Conference on Genetic Algorithms, Morgan Kaufmann, Los Altos, 1991, 509-513.
- [171] Tong, R. M., *The Evaluation of Fuzzy Models Derived from Experimental Data*. Fuzzy Sets and Systems, 4, 1980, 1-12.
- [172] Valenzuela-Rendón, M., *The Fuzzy Classifier System: Motivations and First Results*. W: Proc. of the 1<sup>st</sup> International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Berlin, 1991, 330-334.
- [173] Van Le, T., *Evolutionary Fuzzy Clustering*. W: Proc. of the 2<sup>nd</sup> IEEE Conference on Evolutionary Computation, 2, Perth, Australia, 1995, 753-758.
- [174] Velasco, J.R., *Genetic-Based On-Line Learning for Fuzzy Process Control*. International Journal of Intelligent System, 13 (10-11), 1998, 891-903.
- [175] Venturini, G., *SLA: A Supervised Inductive Algorithm with Genetic Search for Learning Attribute Based Concepts*. W: Proc. European Conference on Machine Learning, Vienna, 1993, 280-296.
- [176] Wang, C. H., Hong, T. P., Tseng, S. S., *Integrating Fuzzy Knowledge by Genetic Algorithms*. IEEE Trans. Evolutionary Computation, 2 (4), 1998, 138-149.

- [177] Wang, L., Langari, R., *Building Sugent-type Models Using Fuzzy Discretization and Orthogonal Parameter Estimation Techniques*. IEEE Trans. Fuzzy Systems, 3, 1995, 454-458.
- [178] Weigend, A. S., Huberman, B. A., Rumelhart, D. E., *Predicting the Future: A Connectionist Approach*. International Journal of Neural Systems, 1, 1990, 193-209.
- [179] Weisstein, E. W., *Banach Fixed Point Theorem*. W: MathWorld - A Wolfram Web Resource, materiały on-line w sieci Internet: <http://mathworld.wolfram.com/BanachFixedPointTheorem.html>
- [180] Weisstein, E. W., *Markov Chain*. W: MathWorld - A Wolfram Web Resource, materiały on-line w sieci Internet: <http://mathworld.wolfram.com/MarkovChain.html>.
- [181] Weisstein, E. W., *Normal Distribution*. W: MathWorld - A Wolfram Web Resource, materiały on-line w sieci Internet: <http://mathworld.wolfram.com/NormalDistribution.html>
- [182] Wirth, N., *Algorytmy + struktury danych = programy*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2001.
- [183] Xu, C., W., Lu, Y., Z., *Fuzzy Modeling Identification and Self-Learning for Dynamic Systems*. IEEE Trans. Systems, Man and Cybernetics, 17, 1987, 683-689.
- [184] Yager, R. R., Filev, D. P., *Podstawy modelowania i sterowania rozmytego*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1995.
- [185] Zadeh, L. A., *Fuzzy Sets*. Information and Control, 8, 1965, 338-353.
- [186] Zadeh, L. A., *Fuzzy Algorithms*. Information and Control, 12, 1968, 94-102.
- [187] Zadeh, L. A., *Towards a theory of fuzzy systems*. W: Kalman, R.E., DeClaris, R.N., Aspects of Network and System Theory. Holt, Rinehart and Winston, New York, 1971, 469-490.
- [188] Zadeh, L. A., *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*. IEEE Trans. Systems, Man and Cybernetics, 3 (1), 1973, 28-44.
- [189] Zadeh, L. A., *The Concept of a Linguistic Variable and Its Application to Approximate Reasoning*. cz. 1, Inf. Science, 8, 199-249, 1975.
- [190] Zadeh, L. A., *The Concept of a Linguistic Variable and Its Application to Approximate Reasoning*. cz. 2, Inf. Science, 8, 301-357, 1975.
- [191] Zadeh, L. A., *The Concept of a Linguistic Variable and Its Application to Approximate Reasoning*. cz. 3, Inf. Science, 9, 43-80, 1975.
- [192] Zieliński, R., *Generatory liczb losowych. Programowanie i testowanie na maszynach cyfrowych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1979.
- [193] Zikidis, K., C., Vasilakos, A., V., *ASAFES2: A Novel, Neuro-Fuzzy Architecture for Fuzzy Computing, Based on Functional Reasoning*. Fuzzy Sets and Systems, 83, 1996, 63-68.





## Dodatek A – parametry funkcji przynależności oraz parametry konkluzji

Poniżej zestawiono szczegółowe parametry opisujące gaussowskie funkcje przynależności przesłanek oraz szerokość nośnika konkluzji systemów rozmytych z parametryzowanymi konkluzjami w regułach **if-then** dla przeprowadzonych badań.

### Zagadnienie aproksymacji danych pomiarowych dla problemu identyfikacji pieca gazowego – Box Jenkins

Najlepsze rezultaty uzyskano dla logicznej interpretacji reguł, z wykorzystaniem implikacji Łukasiewicza oraz operatora minimum jako  $t$ -normy.

Każda kolumna prezentuje pojedynczą regułę rozmytą (łącznie 9 reguł), gdzie pierwsze 20 pozycji opisuje parametry funkcji przynależności w przesłankach (odpowiednio najpierw środek gaussowskiej funkcji przynależności, a następnie jej dyspersja), natomiast 21 pozycja zawiera szerokość nośnika konkluzji:

55.834	57.939	49.716	59.498	48.336	59.453	47.052	51.359	54.077
18.118	6.3424	6.38	8.4401	3.3333	8.9965	4.4506	4.1355	4.4321
56.161	56.613	49.443	56.791	50.238	60.83	48.886	51.552	53.473
11.448	4.5165	4.6505	6.5559	6.0964	9.9508	6.8319	1.2415	3.2624
53.598	56.586	49.191	59.429	49.422	58.658	49.316	51.231	53.108
17.055	6.2999	8.8886	7.793	11.019	12.244	8.8037	2.655	3.5951
52.237	57.733	49.052	57.857	47.115	60.103	48.689	51.761	51.326
13.869	9.7118	10.166	8.5396	13.498	9.726	9.9558	3.165	0.68925
2.5206	-0.31024	0.93501	-0.93083	1.563	-0.98329	2.3312	-1.0371	-2.1372
5.8272	5.3154	4.7812	4.0638	3.8612	3.3432	2.2184	2.0626	0.48058
1.4415	-0.65681	0.69529	-0.8669	1.34	-1.41	2.5173	-0.54027	-2.1626
5.9724	3.7226	6.447	3.6942	3.3458	3.096	3.3437	3.8954	0.83184
-1.2045	-1.3581	1.9989	-0.50276	1.8058	-2.039	1.9889	-0.099329	-1.5011
4.7235	4.0712	4.1462	5.3475	2.5074	2.4146	3.5578	2.5426	2.1335
0.3882	-1.0921	1.8328	-0.1226	2.6867	-1.3312	1.8142	-0.29183	-1.1083
4.066	3.3362	4.874	3.559	2.7271	3.6927	3.4295	3.4895	2.6357
0.68656	-1.1502	1.6859	-0.95021	2.0998	-2.0264	2.695	0.96115	0.19577
2.705	3.1495	3.0196	3.0068	1.9782	2.2145	2.7214	1.9879	2.0728

1.1261	-1.8822	1.933	-1.0653	1.4969	-1.3309	3.0083	1.1135	-0.22866
2.6764	3.8625	2.5958	4.0169	2.3418	4.0934	3.3713	1.8457	2.5603
3.8139	3.9137	3.7117	3.6839	3.1974	2.4261	1.6416	1.4207	1.6839

Parametry liniowej kombinacji położenia konkluzji. Każda kolumna zawiera parametry dla pojedynczej reguły:

147.01	1259	-204.49	-1402.6	405.3	87.685	470.18	-423.55	349.12
-8.7999	-44.982	17.748	54.72	-1.9137	0.59073	-52.812	20.943	-20.086
14.682	25.354	-12.372	-32.704	-15.488	-7.6053	58.645	-14.702	64.01
-0.59056	-6.1917	-6.3032	-2.7636	14.567	17.515	-12.835	1.0054	-40.548
-7.341	3.0257	5.7747	8.4676	-3.914	-10.762	-1.2	1.8455	-8.7897
21.631	11.242	1.302	-16.802	0.055227	-21.314	-23.015	-0.037742	-48.783
-29.345	-7.0404	1.4863	7.3971	1.5659	37.524	31.505	-4.9648	159.53
20.579	-56.346	8.0765	74.454	-56.509	-55.867	9.3858	56.213	-117.37
-10.407	67.04	-9.1282	-86.351	86.785	45.342	-86.315	-66.953	-36.166
16.068	-35.122	14.78	37.005	-34.193	-29.842	-4.5987	28.474	82.172
-13.982	-51.468	-5.5178	65.205	-25.688	12.292	48.75	14.141	1.4078

### **Zagadnienie uogólniania systemu dla problemu identyfikacji pieca gazowego – Box-Jenkins**

Najlepsze rezultaty uzyskano dla logicznej interpretacji reguł, z wykorzystaniem implikacji Goguena oraz operatora iloczynu arytmetycznego jako  $t$ -normy.

Każda kolumna prezentuje pojedynczą regułę rozmytą (łącznie 7 reguł), gdzie pierwsze 20 pozycji opisuje parametry funkcji przynależności w przesłankach (odpowiednio najpierw środek gaussowskiej funkcji przynależności, a następnie jej dyspersja), natomiast 21 pozycja zawiera szerokość nośnika konkluzji:

52.192	48.137	55.019	47.095	56.922	51.376	45.917
5.4604	6.0465	5.3032	6.9516	5.0303	10.588	3.8336
52.9	47.724	56.162	46.234	56.818	50.797	46.884
11.986	5.2089	7.3263	2.1295	5.4517	7.6186	5.4256
55.085	46.702	56.431	46.752	56.694	49.626	48.549
12.518	4.7285	9.1007	8.3454	5.9899	4.7951	8.2786
50.515	48.705	55.5	48.123	54.934	50.316	50.014
13.767	5.9671	8.7345	8.5707	6.0167	4.4163	8.657
0.57753	-0.1145	-1.6818	0.83216	-1.5785	-1.9495	0.82668
5.3911	2.7532	4.9412	2.597	2.8315	0.71077	1.0614
-0.3132	0.42762	-0.37766	0.60598	-1.7806	-1.5321	1.5599
4.2145	2.4828	3.012	2.2164	2.4067	0.85104	1.7538
-0.14787	0.083529	-0.98608	0.81501	-1.1774	-1.7359	1.2108
3.9024	2.2463	3.5934	2.1722	2.2748	2.4086	0.878

0.60231	0.69503	-1.7173	1.1875	-1.9235	-1.3885	1.8502
3.2818	2.7812	2.8501	2.7311	2.8236	3.0698	0.17503
0.017651	0.40732	-1.4674	1.2858	-1.5262	-1.621	1.8724
2.2965	2.2998	2.7483	2.2362	2.4049	2.7879	-0.25592
-0.5084	1.399	-1.1987	1.3344	-1.9916	-0.14726	1.6741
1.6535	2.1742	2.201	0.94372	1.8192	2.53	2.5317
2.5801	3.2318	3.0377	1.4275	2.2807	2.8564	-0.1504

Parametry liniowej kombinacji położenia konkluzji. Każda kolumna zawiera parametry dla pojedynczej reguły:

0.3423	0.52073	0.45022	0.25492	0.35797	0.50832	-0.024523
0.68714	0.91045	1.4626	1.0436	0.96256	1.8325	0.034651
-0.058391	-0.18015	-0.39172	0.27472	-0.31415	-0.10908	0.13438
0.011106	0.16825	-0.14385	0.08657	0.061019	-0.1005	0.16003
0.30323	0.10713	0.19119	-0.43663	0.21436	-0.68715	0.21152
1.2423	-0.14469	-0.25881	0.41696	0.051881	-0.01989	-0.020423
0.17852	-0.7883	-0.7073	0.09696	-0.31159	-0.52606	0.017183
-0.045167	-0.97941	-0.30615	-0.14706	-0.2334	-0.53682	0.03514
0.51037	-0.13281	0.39725	0.036392	0.08742	0.18342	0.010562
0.37096	-0.54063	0.15094	-0.41228	-0.18175	0.010251	0.031028
0.66288	0.59407	0.76967	-0.11418	0.19088	0.2084	0.0067777

### **Zagadnienie uogólniania systemu dla problemu przewidywania ilości plam na słońcu – Sunspots**

Najlepsze rezultaty uzyskano dla logicznej interpretacji reguł, z wykorzystaniem implikacji Zadeha oraz operatora minimum jako  $t$ -normy.

Każda kolumna prezentuje pojedynczą regułę rozmytą (łącznie 4 reguły), gdzie pierwsze 24 pozycji opisuje parametry funkcji przynależności w przesłankach (odpowiednio najpierw środek gaussowskiej funkcji przynależności, a następnie jej dyspersja), natomiast 25 pozycja zawiera szerokość nośnika konkluzji:

0.70649	0.49554	0.44557	0.75387
1.3437	0.97833	1.2167	0.99903
0.37352	0.498	0.84777	0.89127
1.1889	0.94027	1.1492	0.83473
0.58729	0.36995	0.71281	0.52559
0.9706	0.93113	1.3518	0.53681
0.51942	0.69762	0.65512	0.67956
0.92102	0.90318	1.299	0.86796
0.53905	0.43761	0.59875	0.34181
0.99311	1.082	0.89838	0.74622

0.63553	0.3916	0.62271	0.3106
1.1214	1.0198	1.0542	0.56893
0.34473	0.48288	0.35495	0.40408
0.87772	0.91508	1.0832	0.51725
0.39212	0.34377	0.49307	0.58026
1.0652	0.91891	1.5319	1.2725
0.67509	0.6176	0.66359	0.85888
1.1089	0.85715	1.2413	0.96647
0.61974	0.31233	0.54443	0.94474
0.96796	0.80725	1.3048	0.78944
0.58982	0.51205	0.81249	0.96569
0.95777	1.3172	0.75712	0.85893
0.37528	0.42743	0.67388	0.69435
1.1545	0.99015	1.1445	0.82223
0.69531	0.69206	0.74842	0.72273

Parametry liniowej kombinacji położenia konkluzji. Każda kolumna zawiera parametry dla pojedynczej reguły:

-0.3975	0.4856	0.12948	-0.072021
0.041491	0.095536	-0.15056	-0.029991
-0.02073	-0.27188	0.84007	0.20847
-0.18626	-0.33447	0.3101	-0.19842
0.097905	0.18166	-0.0065001	-0.20847
-0.048412	0.37075	-0.46929	-0.24185
0.033388	0.33651	-0.075966	0.26206
-0.26142	-0.026689	-0.055396	0.24661
-0.05639	0.075685	-0.043495	0.17361
-0.24711	-0.40586	0.12284	0.087075
0.12379	-0.027463	0.061286	0.11851
-0.11432	-0.18538	-0.44552	-0.029348
1.6968	1.968	-0.26267	-0.19326

### **Zagadnienie uogólniania systemu dla problemu predykcji chaotycznych szeregów czasowych – Mackey-Glass**

Najlepsze rezultaty uzyskano dla logicznej interpretacji reguł, z wykorzystaniem implikacji Kleene-Dienes oraz operatora iloczynu algebraicznego jako  $t$ -normy.

Każda kolumna prezentuje pojedynczą regułę rozmytą (łącznie 4 reguły), gdzie pierwsze 8 pozycji opisuje parametry funkcji przynależności w przesłankach (odpowiednio najpierw środek gaussowskiej funkcji przynależności, a następnie jej dyspersja), natomiast 9 pozycja zawiera szerokość nośnika konkluzji:

1.2177	0.62563	0.97776	1.0511
0.29493	0.60466	0.49173	0.56703
1.2284	0.72773	1.1602	0.72275
0.38495	0.95288	0.33128	0.50455
1.0529	0.9266	1.0256	0.75024
0.82014	0.66525	0.54025	0.52168
0.67018	1.0454	1.1032	0.68649
0.66138	0.37826	0.52867	0.70564
0.35749	0.32934	0.54896	0.41853

Parametry liniowej kombinacji położenia konkluzji. Każda kolumna zawiera parametry dla pojedynczej reguły:

0.44873	0.15768	1.627	-0.057674
-1.9153	-0.27993	0.46744	0.2196
1.4985	0.60918	-2.072	0.55985
1.467	-0.3878	1.0889	0.29955
-2.072	0.88584	-0.016959	0.65941

### **Zagadnienie uogólniania dla problemu wyceny nieruchomości – Boston Housing**

Najlepsze rezultaty uzyskano dla logicznej interpretacji reguł, z wykorzystaniem implikacji Łukasiewicza oraz operatora iloczynu algebraicznego jako  $t$ -normy.

Każda kolumna prezentuje pojedynczą regułę rozmytą (łącznie 5 reguł), gdzie pierwsze 26 pozycji opisuje parametry funkcji przynależności w przesłankach (odpowiednio najpierw środek gaussowskiej funkcji przynależności, a następnie jej dyspersja), natomiast 27 pozycja zawiera szerokość nośnika konkluzji:

0.60965	12.405	0.046986	47.056	0.44107
19.402	56.055	19.887	73.343	10.472
42.797	30.41	30.597	110.33	55.875
137.14	172.42	126.34	163.1	157.41
4.2257	20.511	9.9556	19.754	5.0294
36.17	16.192	23.243	33.238	21.906
0.22727	0.16168	0.18362	0.33458	0.055223
1.5083	0.87844	1.4611	1.2635	1.0568
0.48625	0.58335	0.57331	0.80411	0.52024
0.40265	0.1909	0.61708	0.62058	0.57959
6.0473	6.16	7.1273	7.6754	7.9835
4.6203	2.5532	4.4097	5.3485	2.6497
43.536	100.74	67.257	61.46	45.369
168.2	136.64	199.46	212.55	120.9
4.7027	5.5549	7.3309	4.723	4.8485

11.49	9.1649	10.766	12.895	14.93
4.9816	23.288	7.4354	25.95	2.9006
17.826	21.468	13.777	18.929	11.949
392.26	735.18	291.99	696.82	195.3
558.96	256.8	347.32	571.62	665.19
17.206	21.428	17.349	20.674	13.727
9.0532	4.865	14.323	12.82	7.8616
390.29	318.82	450.91	341.65	433.45
217.67	398.22	269.47	308.74	235.15
11.128	25.523	13.442	21.876	3.9667
33.59	27.886	29.922	31.911	22.284
42.601	33.261	42.601	34.248	34.253

Parametry liniowej kombinacji położenia konkluzji. Każda kolumna zawiera parametry dla pojedynczej reguły:

-9.9137	0.50085	-15.716	0.96668	-19.483
8.7163	-2.2958	-15.693	0.9748	9.0276
-1.5428	0.39434	1.2004	-2.1546	0.84209
-2.9345	5.5464	-4.0294	-2.7876	10.12
14.406	53.817	-9.134	-14.516	-17.054
8.344	13.08	-2.4513	4.5044	-9.6003
8.4957	23.719	0.87677	-26.861	41.175
-0.87911	0.26528	0.81481	0.020333	-0.063047
-27.432	2.2344	33.286	-3.8767	-4.4104
-5.8072	29.441	4.6015	-47.852	14.043
-0.35135	-1.5032	0.36751	1.3322	-0.59778
35.22	3.416	-45.613	26.428	-2.3908
0.23587	-0.056682	0.18704	-0.028819	-0.41384
-2.2322	-0.10839	1.5374	-0.94224	-0.67098