

181/2003

Raport Badawczy
Research Report

RB/58/2003

**Genetyczne projektowanie
topologii rozmytych
sieci logicznych**

P. Nycz

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2003

Genetyczne projektowanie topologii rozmytych sieci logicznych

18 grudnia 2003

1 Wprowadzenie

Dotychczas opracowano kilka rodzajów sieci neuronowych, które znalazły swoje zastosowania w praktyce. Podczas stosowania sieci do rozwiązania konkretnego zagadnienia pojawia się problem doboru właściwej topologii sieci oraz sposobu jej nauki. Na topologię składa się liczba neuronów, ich układ w warstwach oraz połączenia pomiędzy nimi z użyciem wag o odpowiednich wartościach. Zadaniem algorytmu uczącego jest dobór wartości wag, który pozwoli na otrzymywanie wartości z sieci zgodnie z oczekiwaniami. O ile problem uczenia sieci jest dobrze opracowany to dobór architektury jest problemem otwartym, który podlega ciągłym badaniom.

Za cel projektu postawiono próbę zautomatyzowania procesu doboru topologii oraz wag rozmytej sieci logicznej, które zaliczane są do systemów „neuronowo-rozmytych” (ang. fuzzy neural networks). Łączą cechy sztucznych sieci neuronowych z możliwością pracy z danymi obciążonymi brakiem precyzji. Problem ten postanowiono rozwiązać z użyciem algorytmu genetycznego.

2 Podstawy działania rozmytych sieci logicznych

Tak jak w przypadku sieci neuronowych rozmyte sieci logiczne składają się z pewnej ilości neuronów, które pogrupowane są w warstwy oraz połączone są wagami o pewnych wartościach. W kolejnych podpunktach zostaną przedstawione podstawowe wiadomości na temat działania rozmytych sieci logicznych.

2.1 Neurony logiczne

Niech $x = [x_1, x_2, \dots, x_n]$ będzie wektorem sygnałów wejściowych, a $w = [w_1, w_2, \dots, w_n]$ odpowiednio wektorem wag. Dodatkowo przyjmuje się, że obydwie te wektory spełniają ograniczenie: $x, w \in [0, 1]^n$, co oznacza, że mogą być one traktowane jako elementy n -wymiarowej hiperkostki.

- Neuron logiczny typu OR realizuje odwzorowanie $[0, 1]^n \rightarrow [0, 1]$ opisaną formułą:

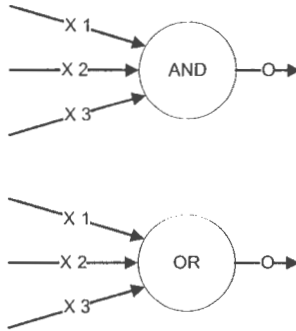
$$o = \text{OR}(x; w) = \text{OR}[x_1 \text{ AND } w_1, x_2 \text{ AND } w_2, \dots, x_n \text{ AND } w_n]. \quad (1)$$

Przyjmuje się, że operatory AND oraz OR są szczególnym przypadkiem kolejno T -normy oraz S -normy. Zgodnie z tym założeniem wyrażenie to można zapisać następująco:

$$o = S_{i=1}^N [x_i \text{ t } w_i]. \quad (2)$$

- Neuron typu AND realizuje odwzorowanie $[0, 1]^n \rightarrow [0, 1]$ opisaną poniższą formułą:

$$o = \text{AND}(x; w) = \text{AND}[x_1 \text{ OR } w_1, x_2 \text{ OR } w_2, \dots, x_n \text{ OR } w_n] \quad (3)$$



Rysunek 1: Neurony typu AND oraz OR.

Zgodnie z tym co napisano wcześniej powyższe wyrażenie można zapisać następująco:

$$o = T_{i=1}^N [x_i \text{ s } w_i]. \quad (4)$$

Wyżej zdefiniowane odwzorowania należy traktować jako uogólnione operacje logiczne na danych wejściowych. Staje się to widoczne przy założeniu, że wartości wszystkich wejść neuronów są równe 0 lub 1. Dla neuronu typu OR, przy ustalonym wektorze wag $w = 1$ odpowiednia formuła redukuje się do postaci:

$$o = x_1 \text{ OR } x_2 \text{ OR } \dots \text{ OR } x_n. \quad (5)$$

W przypadku neuronu typu AND oraz wektora $w = 0$ otrzymamy odpowiednio:

$$o = x_1 \text{ AND } x_2 \text{ AND } \dots \text{ AND } x_n. \quad (6)$$

2.2 Negacja wejść neuronów logicznych

Powyżej zaprezentowano dwa modele obliczeniowe rozmytych neuronów logicznych, które pozwalają na realizację operacji rozmytej dyzjunkcji oraz

koniunkcji danych wejściowych wprowadzanych w postaci prostej. Poniżej przedstawiono operator rozmytej negacji logicznej, który pozwala na doprowadzenie sygnałów poszczególnych zmiennych do neuronów w postaci zanegowanej.

Operacje rozmytej negacji przeprowadza się zgodnie z definicją dopełnienia zbioru rozmytego. W kontekście rozmytych neuronów logicznych zanegowana wartość zmiennej doprowadzonej do i -tego wejścia neuronu wyznaczana jest z formuły:

$$NOT(x_i) = \bar{x}_i = 1 - x_i, \quad (7)$$

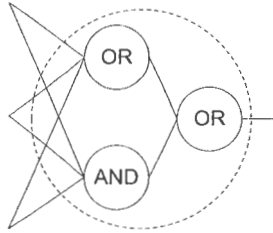
co odpowiada operatorowi dopełnienia zbioru rozmytego.

3 Sposoby łączenia neuronów logicznych

Neurony typu AND, OR oraz NOT są podstawowymi elementami wystarczającymi do konstruowania sieci o dowolnym stopniu złożoności. Jak napisano w [?], naturalnym wydaje się przeprowadzenie uogólnienia, które polega na wprowadzeniu elementów o cechach „pośrednich” pomiędzy neuronami AND oraz OR. Taki efekt polega na połączeniu w jednorodną dwuwarstwową strukturę neuronów typu AND oraz OR w sposób jak pokazano na rysunku 2.

Główną motywacją do konstruowania tego typu uogólnień jest możliwość uzyskiwania odwzorowań pośrednich pomiędzy wcześniej omawianymi typami neuronów. Dzięki takiemu połączeniu oraz odpowiednio ustalonym wartościom wag v_1 oraz v_2 istnieje możliwość „płynnego” przełączania się pomiędzy sygnałami pochodzącymi z neuronów z warstwy wejściowej. W szczególności gdy $v_1 = 1$ oraz $v_2 = 0$ neuron działa jako neuron typu AND, natomiast gdy $v_1 = 0$ oraz $v_2 = 1$ realizowania jest funkcja neuronu typu OR. Przyjęta notacja:

$$o = OR/AND(x; w; v) \quad (8)$$



Rysunek 2: Architektura neuronu typu AND / OR.

podkreśla sposób pośredniego działania neuronu AND / OR. Omawianą strukturę zapisujemy jako:

$$\begin{aligned}
 o &= \text{OR}(\{z_1 \ z_2\}; v), \\
 z_1 &= \text{AND}(x; w_1), \\
 z_2 &= \text{OR}(x, w_2),
 \end{aligned} \tag{9}$$

gdzie $v = [v_1 \ v_2]$, $w = [w_{i_1}, \ w_{i_2}, \ \dots, \ w_{i_n}]$, $i = 1, 2$, są wagami przy odpowiednich wejściach neuronów.

4 Uczenie rozmytych sieci logicznych

Do uczenia rozmytych sieci logicznych możliwe jest zastosowanie np. metod gradientowych, jednak proces ten dla złożonych sieci jest bardzo czasochłonny. Dotychczas nie opracowano metody nauki na wzór algorytmu wstecznej propagacji stosowanego dla klasycznych sieci neuronowych, pozwalającej na wyznaczenie wartości błędów w warstwach ukrytych. Dlatego też w chwili obecnej uczenie rozmytych sieci logicznych przeprowadza się z użyciem algorytmów ewolucyjnych, które opisano poniżej lub zaprezentowanych w [?] metod hybrydowych.

4.1 Struktura chromosomu

W algorytmie ewolucyjnym zaprojektowanym do uczenia sieci neuronowej osobnik w populacji reprezentujący potencjalne rozwiązanie zawiera elementy wektora wag W całej sieci. Elementy wektora W w rozmytej sieci logicznej przyjmują wartości z zakresu: $w \in [0, 1]$. Każde z wejść może przysyłać informacje w postaci prostej - x lub zanegowanej - $\text{NOT}(x)$. Ilość wag w sieci zależy od jej gęstości (ilości neuronów oraz połączeń pomiędzy nimi). Jak dotychczas nie udało się stwierdzić jaka metoda kodowania w algorytmie genetycznym jest najlepiej dopasowana do rozwiązania tego problemu.

Założenie że, zakodowane wartości wag będą liczbami $w' \in [-1, 1]$ umożliwi reprezentację wejść prostych oraz zanegowanych. Podczas procesu dekodowania przyjęto następującą zasadę: jeżeli $w'_i < 0$ to wejście jest w postaci zanegowanej ($1 - x$) a wartość wagi $w = |w'|$. Jeżeli $w' > 0$ wówczas wejście jest w postaci prostej, natomiast wartość wagi $w = w'$.

Kodowanie binarne. W celu osiągnięcia odpowiednio wysokiej dokładności (przyjęto 4 cyfry po przecinku), do reprezentacji użyto 14 bitów dla każdej wagi. Długość chromosomu (wektora) w tym wypadku wynosi liczba wag*14.

Kodowanie zmiennopozycyjne. W tym przypadku dokładność została narzucona przez reprezentację liczby typu double w kompilatorze C++. Długość chromosomu odpowiada ilości wag.

5 Ewolucyjna metoda doboru topologii rozmytej sieci logicznej.

W niniejszym podrozdziale opisano podstawowy aparat pojęć niezbędnych przy genetycznym projektowaniu topologii sieci neuronowych. Podczas doboru metody kodowania osobnika można wyróżnić pewne własności, które

uważane są za kluczowe dla wydajności oraz skuteczności procesu poszukiwania szczegóły zawarto w [?] oraz [?]. Odnoszą się one do genotypu (osobnik będący elementem populacji, która jest zbiorem architektur sieci) dekodowanego do konkretnej architektury sieci. Poniższe definicje potraktowano jako pewne wytyczne do projektowania struktury chromosomu osobnika, które mają zapewnić wysoką skuteczność pracy algorytmu.

Definicja 5.1. *Zupełność (ang. completeness). System kodowania jest zupełny, jeżeli każdy element ze zbioru architektur może być przedstawiony za pomocą przyjętej metody kodowania.*

Definicja 5.2. *Domkniętość (ang. closure). System kodowania jest domknięty, jeżeli każdy osobnik utworzony przy pomocy operatorów genetycznych jest rozwiązaniem dopuszczalnym.*

Definicja 5.3. *Bliskość (ang. proximity). Warunek sąsiedniości jest spełniony, gdy przy dokonywaniu małych zmian w strukturze osobnika wprowadza niewielkie zmiany w architekturze sieci.*

Definicja 5.4. *Izomorfizm (ang. isomorphism). Warunek izomorfizmu jest spełniony, jeżeli każdy genotyp reprezentuje dokładnie jedno rozwiązanie (dokładnie jedną architekturę sieci).*

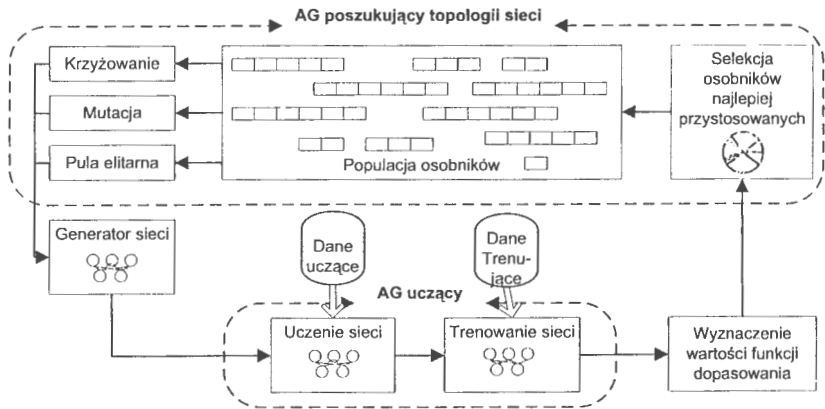
Definicja 5.5. *Modularność (ang. modularity). Niech sieć N_1 zawiera w kilku miejscach kopie podsieci N_2 . Genotyp sieci N_1 jest modularny, jeżeli zawiera genotyp sieci N_2 dokładnie jeden raz.*

Proces poszukiwania topologii może zostać przeprowadzony na dwa sposoby.

- **Jednoetapowy**, który polega na zakodowaniu w chromosomie osobnika architektury sieci wraz z jej odpowiednimi wagami. Zakłada się że genotyp osobnika przedstawia sieć już nauczoną, która może zostać poddana procesowi oceny.

- **Dwuetapowy.** Zakłada się że, osobnik w głównej pętli algorytmu ewolucyjnego zawiera wyłącznie „opis” architektury sieci, natomiast optymalizacja zestawu wag jest wykonywana w formie podprocedury w momencie wyznaczania wartości funkcji przystosowania osobnika.

Pierwszy ze sposobów jest niewątpliwie szybszy, natomiast trudniejszy do implementacji ze względu na skomplikowaną strukturę chromosomu. Badania przeprowadzane w tym kierunku pokazują jednak, że taki sposób postępowania jest możliwy do realizacji co przedstawiono w [?]. Drugi ze sposobów pozwala na większą swobodę w kwestii doboru metody nauki sieci oraz zapewnia większą kontrolę tego procesu. Na rysunku 3 przedstawiono diagram



Rysunek 3: Diagram genetycznego poszukiwania topologii rozmytej sieci logicznej

algorytmu ewolucyjnego, przeznaczonego do poszukiwania topologii rozmytej sieci logicznej w sposób dwuetapowy.

5.1 Proponowany - „klastrowy model kodowania topologii”

Do wyjaśnienia klastrowej metody kodowania niezbędne jest wprowadzenie dwóch nowych pojęć.

- **Klaster** - zbiór neuronów wchodzący w skład warstwy. Warstwa może zawierać od jednego do kilku klastrów. Każdy klaster składa się z jednego lub większej ilości neuronów dowolnego typu (w przypadku rozmytej sieci logicznej AND lub OR).
- **Rejestr warstwy** - zmienna całkowita o wartościach dodatnich. Jest wykorzystywana do przechowywania bieżącej wartości indeksu warstwy. W skrócie oznaczany rw .
- **Rejestr klastra** - zmienna całkowita o wartościach dodatnich. Jest wykorzystywana do przechowywania bieżącej wartości indeksu tworzonego klastra. W skrócie oznaczany rk .

Jak napisano wyżej metody kodowania pośredniego zawierają ciąg instrukcji, którym przypisuje się odpowiednią interpretację. W omawianym modelu przyjęto, że do kodowania będą używane znaki z alfabetu:

$$A = \{A, O, >, <\}.$$

Interpretacja poszczególnych znaków alfabetu:

- A – utwórz neuron typu AND,
- O – utwórz neuron typu OR,
- $>$ – zwiększenie o 1 rejestru warstwy. Przed inkrementacją rejestru warstwy wykonywane są następujące czynności:

1. Wszystkie neurony utworzone od momentu poprzedniego wystąpienia znaku innego niż symbole neuronów zalicz do jednego klastra.
 2. Dodaj klaster do warstwy o indeksie zawartym w rejestrze warstwy. Jeżeli jest to możliwe, utwórz połączenie metodą każdy z każdym pomiędzy neuronami z klastra o największym indeksie z warstwy poniżej. W przeciwnym wypadku nie wykonuj żadnych czynności.
- < – Zmniejszenie o 1 rejestru warstwy. Przeprowadzenie tej operacji jest możliwe tylko w przypadku, gdy wartość rejestru jest większa od 0 (wartość 0 oznacza warstwę wejściową). Przed dekrementacją wykonywane są następujące czynności:

1. wszystkie neurony utworzone od momentu wystąpienia znaku innego niż symbole neuronów zalicz do jednego klastra.
2. dodaj klaster do warstwy o indeksie zawartym w rejestrze warstwy. Jeżeli jest to możliwe utwórz połączenie metodą każdy z każdym z klastra o największym indeksie z warstwy poniżej. W przeciwnym przypadku nie podejmuj żadnych działań.

Przykład 5.1. W celu ilustracji działania algorytmu podano przykład dekodowania osobnika postaci $OA > AOOA > OO$.

1. *Inicjalizacja rejestru indeksem 0.*
2. *Odczytane symbole: „OA”. Utworzono klaster zawierający dwa neurony - AND oraz OR; $rk = 1, rw = 0$.*
3. *Odczytany symbol „>” - Wykonane działania:*
 - (a) *Utworzony klaster umieszczono w warstwie 0.*

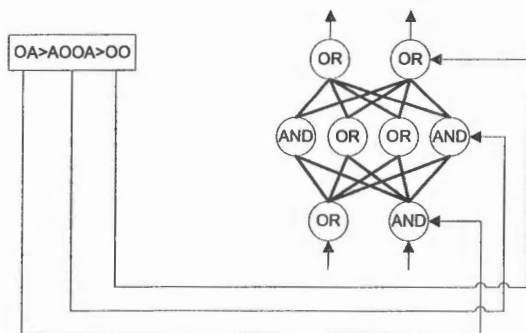
- (b) Inkrementacja rejestru warstwy ($rw = 1$).
 - (c) Nie zdefiniowano połączeń (klastr został utworzony jako pierwszy).
4. Odczytane symbole: „AOOA”. Utworzono klastr z neuronów - AND, OR, OR, AND; $rw = 1$, $rk = 2$.
5. Odczytany symbol „>”; $rw = 1$. Wykonane działania:
- (a) Dodano klastr 2 do warstwy 1.
 - (b) Utworzono pełne połączenie klastra 2 z klastrem 1.
 - (c) Inkrementacja rejestru warstwy ($rw = 2$).
6. Odczytane symbole: „OO”. Utworzono klastr nr. 3.
7. Koniec łańcucha znaków. Dodano klastr 3 do warstwy 2, utworzono pełne połączenie klastra 3 z klastrem 2.

Po wykonaniu powyższego algorytmu otrzymano sieć przedstawioną na rysunku 4. Opisany algorytm ma jednak ograniczone zastosowanie, gdyż nie umożliwia on tworzenia tzw. sieci modułarnych. Przykład architektury takiej sieci przedstawiono na rys. 5.

5.2 Rozszerzony model klastrowy

Jak napisano na zakończenie poprzedniego podpunktu model klastrowy cechuje się pewnymi ograniczeniami uniemożliwiającymi osiągnięcie pewnych klas topologii sieci. W celu wyeliminowania tych wad model rozszerzono o kilka własności.

1. Wprowadzono dodatkowy rejestr rp umożliwiający tworzenie połączeń pomiędzy klastrami utworzonymi w różnej kolejności, dowolnie rozmieszczonymi w sieci. Zawiera on liczbę całkowitą dodatnią będącą



Rysunek 4: Sieć otrzymana podczas procesu translacji z przykładu 5.1.

indeksem aktualizowanym po każdej operacji tworzenia nowego klastra zgodnie z formułą:

$$rp(rk_n) = \max_{j=1, \dots, n-1} \{rk_j : rk_j < rk_n - 1 \wedge nw(rk_j) < nw(rk_n)\},$$

gdzie:

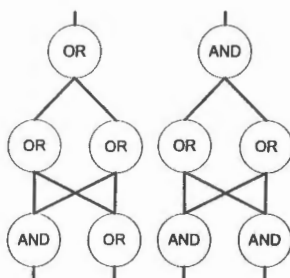
$\{rk_1, \dots, rk_{n-1}\}$ – jest zbiorem wszystkich poprzednio utworzonych klastrów,

$rp(rk_j)$ – nowa wartość rejestru połączeń dla klastra rk_j ,

$nw(rk_j)$ – indeks warstwy, w której znajduje się klaster rk_j .

2. Do alfabetu dodano dwa symbole: „c” oraz „-”, którym nadano znaczenie:

- Symbol „c” wymusza utworzenie połączenia pomiędzy klastrem ostatnio utworzonym, a klastrem o indeksie wskazanym w rejestrze połączeń.



Rysunek 5: Sieć otrzymana podczas procesu translacji z przykładu 5.2.

- Symbol "–" zmniejsza wartość rejestru połączeń zgodnie ze wzorem:

$$rp(rk_n) = \max_{j=1, \dots, n-2} \{ rk_j < rp(rk_{n-1}) \wedge nw(rk_j) < nw(rk_n) \},$$

gdzie indeks $n - 1$ wskazuje na klaster utworzony bezpośrednio przed klastrem rk_n .

Jak wynika z formuły podczas pomniejszania rejestru, zostaje wyznaczony klaster utworzony jako przedostatni, ale leżący w warstwie poniżej. W przypadku, gdy klaster wyznaczony wg. formuły nie istnieje, rejestr przyjmuje wartość połączenie niemożliwe.

Przykład 5.2. W celu ilustracji rozszerzonego modelu klastrowego przedstawiono przykład tworzenia sieci modularnej, która jest reprezentowana przez osobnik: „AO > OO > Oc << AA > OO > Ac – c – c”

1. Inicjalizacja:

- $rk = 0$,
- $rw = 0$,

- τp = połączenie niemożliwe.

2. Odczytane symbole "AO" Utworzono klaster „AO”, dodano go do warstwy 0; $\tau k = 1$

3. Odczytane symbole ">"; $\tau w = 1$.

4. Odczytane symbole „OO”:

(a) Utworzono klaster "OO" nadano mu indeks 2, dodano go do warstwy 1,

(b) Utworzono pełne połączenie klastra 1 z klastrem 2,

(c) τp = połączenie niemożliwe.

5. Odczytany symbol „>”:

(a) τp = połączenie niemożliwe.

(b) $\tau w = 2$.

6. Odczytany symbol "O”:

(a) Utworzono klaster "O" nadano mu indeks 3, dodano go do warstwy 2,

(b) Utworzono pełne połączenie klastra 3 z klastrem 2,

(c) $\tau p = 1$.

7. Odczytany symbol „c”. Wymuszenie pełnego połączenia pomiędzy klastrem 3 a klastrem 1.

8. Odczytane symbole „<<”:

(a) $\tau w = 0$,

(b) $\tau p = 1$.

9. Odczytane symbole „AA”:
- (a) Utworzono klaster „AA”, nadano mu indeks 4. Dodano go do warstwy 0.
 - (b) $rw = 1$,
 - (c) $rp =$ połączenie niemożliwe.
10. Odczytany symbol „>”:
- (a) $rw = 1$.
 - (b) $rp =$ połączenie niemożliwe.
11. Odczytane symbole „OO”:
- (a) Utworzono klaster OO, nadano mu indeks 5. Dodano go do warstwy 1.
 - (b) $rw = 1$,
 - (c) $rp = 1$,
 - (d) Utworzono pełne połączenie pomiędzy klastrem 5 a 4.
12. Odczytane symbole „>”:
- (a) $rw = 2$,
 - (b) $rp = 1$.
13. Odczytane symbole „A”:
- (a) Utworzono klaster A, nadano mu indeks 6. Dodano go do warstwy 2.
 - (b) Utworzono pełne połączenie pomiędzy klastrem 6 a 5.
 - (c) $rw = 1$,

(d) $rp = 1$.

14. Odczytany symbol „c”: Wymuszenie połączenia pomiędzy klastrem 6 a 4.

15. Odczytany symbol „-”:

(a) $rp = 2$,

(b) $rw = 2$.

16. Odczytany symbol „c”: Wymuszenie połączenia pomiędzy klastrem 6 a 2.

17. Odczytany symbol „-”:

(a) $rp = 1$,

(b) $rw = 2$.

18. Odczytany symbol „-”. Wymuszenie połączenia pomiędzy klastrem 6 a 1.

Efektem wykonania powyższego algorytmu jest sieć przedstawiona na rysunku 5.

Zalety i wady przyjętego sposobu kodowania. Odnosząc się do cech wymienionych na początku podrozdziału 5 wyznaczających jakość sytemu kodowania, przedstawiono zalety oraz wady rozszerzonego modelu klastrowego, w skrócie oznaczonego RMK.

Zupełność.(*ang. completeness*) RMK jest zupełny. Dzięki wprowadzeniu rejestru połączeń możliwa jest budowa sieci neuronowej o dowolnej strukturze.

Domkniętość.(*ang. closure*) RMK spełnia warunek domkniętości, gdyż każdy genotyp o dowolnej „zawartości” jest dekodowany do poprawnej sieci.

Wrażliwość. (*ang. proximity*) RMK jest systemem mało wrażliwym, gdyż zmiana jednego symbolu nie powoduje dużej zmiany w strukturze całej sieci.

Podsumowując można stwierdzić, że proponowana metoda kodowania posiada większość cech wyznaczających wysoką jakość systemu kodowania. Nie wszystkie założenia zostały spełnione, gdyż otrzymany efekt jest swojego rodzaju kompromisem. Należy przypuszczać, że spełnienie wszystkich wymiennych wymagań jest bardzo trudne o ile jest możliwe.

Analizując dogłębnie zaprezentowany model można zauważyć dodatkowo cechy, które czynią wybraną metodę atrakcyjną.

- Liczność zbioru znaków w alfabecie jest niewielka i jednocześnie bezpośrednio odzwierciedla model topologii.
- Długość genotypu wzrasta liniowo w stosunku do wielkości sieci. Jest to spowodowane faktem braku konieczności bezpośredniego kodowania połączeń pomiędzy neuronami. Kodowaniu podlegają tylko neurony.
- Operowanie klastrem jako podstawową jednostką składową bardzo ułatwia tworzenie sieci modularnych.
- RMK cechuje się tzw. krótkimi schematami, które umożliwiają przeniesienie poszczególnych elementów sieci (np. podsieci) poprzez wymianę zawartości pomiędzy genotypami. Zostało to podkreślone na rysunku 4, gdzie zaznaczono fragmenty (podsieci) odpowiadające odpowiednim fragmentom osobnika.
- RMK jest systemem elastycznym, umożliwiającym wzbogacanie go o nowe symbole np. nowe typy neuronów lub specjalne typy połączeń. Tej tematyki nie poruszano w niniejszej pracy, niemniej jednak proponowany system jest przygotowany na takie rozszerzenia.

Przyjęta struktura chromosomu dla optymalizacji rozmytej sieci logicznej.

Rozszerzony model klastrowy pozwala na tworzenie sieci o dowolnej architekturze. W praktyce poszukiwania topologii ilość neuronów w warstwie wejściowej oraz wyjściowej sieci jest narzucona przez strukturę danych trenujących. Problem ten można rozwiązać przez poszukiwanie topologii tylko dla warstw ukrytych, co zostało uczynione.

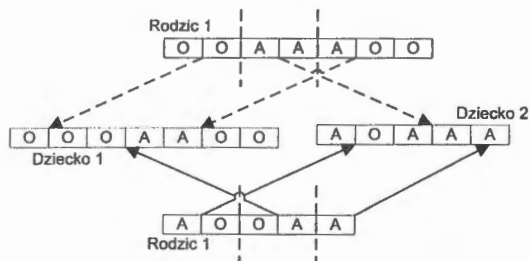
Funkcja oceny. Jak napisano wcześniej wybór odgrywa istotną rolę w mechanizmie genetycznego poszukiwania rozwiązań. Dobrą metodą określania jakości odwzorowania realizowanego przez sieć jest średnia kwadratów błędów popełnianych dla poszczególnych wzorców. Proces uczenia sieci jest przeprowadzany podczas oceny osobnika. Przedstawiono go na schemacie przyjętego algorytmu poszukiwania optymalnej topologii. Założono, że wyznaczanie optymalnego zestawu wag odbywa się w sposób ewolucyjny. Jak napisano w [?], formułę wyznaczającą błąd warto znormalizować w sposób uwzględniający liczbę neuronów w warstwie wejściowej oraz wyjściowej, jak przedstawiono poniżej:

$$Q = 100 \frac{o_{max} - o_{min}}{N P} \sum_{p=1}^P \sum_{i=1}^N (o_{pi} - l_{pi})^2, \quad (10)$$

gdzie: o_{max} oraz o_{min} są maksymalnymi i minimalnymi wartościami współczynników wyjściowych reprezentowanego problemu (zakłada się, że są one jednakowe dla wszystkich wyjść), N jest liczbą wyjść sieci, natomiast P jest liczbą wzorców zbioru uczącego.

Operatory genetyczne.

- Jako operatora reprodukcji, który wyznacza nowe pokolenie chromosomów opisujących strukturę sieci użyto operacji krzyżowania dwupunktowego pozwalającego na uzyskanie zmiennej długości chromosomu. Schemat działania operatora zaprezentowano na rysunku 6. Postępowanie jest wzorowane na standardowej metodzie krzyżowania dwupunktowego. W pierwszym etapie dokonuje się losowania dwóch punktów krzy-



Rysunek 6: Zasada działania operatora krzyżowania genotypów różnej długości.

zowania z zakresu krótszego chromosomu i wymienia zawartość chromosomów pomiędzy wylosowanymi punktami.

- Operator mutacji zdefiniowano jako zmianę symbolu na inny (z dozwolonego zestawu), którego wartość i pozycja wyznaczana jest losowo.

6 Podsumowanie

Rozpoczynając pracę nad projektem rozpatrywano wiele propozycji rozwiązania problemu optymalnej topologii. W miarę postępu prac okazało się, że genetyczna metoda przynosi zadawalające efekty w krótkim czasie poszukiwania rozwiązania. Nie przeprowadzono jeszcze gruntownych testów proponowanego rozwiązania. Analizując wstępne obserwacje, obserwacje można przypuszczać, że zaproponowane podejście jest poprawne. Testy będą obejmowały zbiory danych, które zawierają silne związki pomiędzy poszczególnymi zmiennymi. Takie sprawdzanie poprawności stosuje się dla algorytmów

uczących klasyczne sieci neuronowe. Dopiero na podstawie wyników pracy algorytmu z takimi danymi można będzie ocenić słuszność proponowanego podejścia.

