

Raport Badawczy

RB/51/2016

Research Report

**Sieci komparatorów obiektów
złożonych – uczenie struktury**

Ł. Sosnowski

**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) 3810100

fax: (+48) (22) 3810105

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

Warszawa 2016

SIECI KOMPparatorÓW ObiektÓw ZŁożonych - UCZENIE STRUKTURy

Łukasz Sosnowski

Instytut Badań Systemowych Polskiej Akademii Nauk
e-mail: Lukasz.Sosnowski@ibspan.waw.pl

Streszczenie. Artykuł przedstawia opis algorytmu uczenia struktury sieci komparatorów obiektów złożonych w celu optymalizacji efektywości i wydajności badanej sieci. Tematyka dotyczy podejścia wnioskowania bazującego na podobieństwie przy użyciu którego rozwiązywane są problemy rozpoznawania i identyfikacji obiektów złożonych.

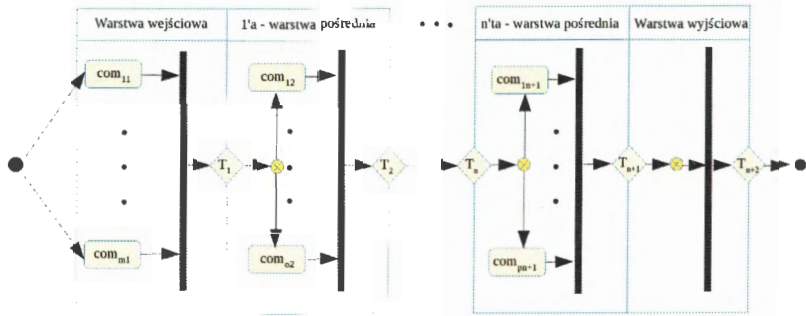
Słowa kluczowe: sieci komparatorów, podobieństwo obiektów złożonych, uczenie sieci komparatorów, rozpoznawanie i identyfikacja.

1 WPROWADZENIE

Sieci komparatorów to jedno z narzędzi umożliwiających realizację rozpoznawania i identyfikacji obiektów złożonych bazując na ich wzajemnych podobieństwie. Podobieństwo jest pojęciem dość często spotykanym w pracach dotyczących rozpoznawania wzorców, ale poszczególne realizacje różnią się dość znacznie od siebie. Do tej pory nie występowała zuniifikowana i spójna metodyka postępowania budowania rozwiązań bazujących na podobieństwie, lecz w każdym przypadku rozwiązanie to tworzone było w nieco inny sposób [2, 3].

Komparatory obiektów złożonych jak również budowane z nich sieci wykorzystują regułę NPO [4] (najbardziej podobnego obiektu) do wyznaczenia obiektów, które są wystarczająco podobne do obiektu wejściowego. Obiekty podobne przynależą do wspólnych klas, charakteryzują się podobnym zestawem atrybutów (interpretowanych jako cechy), jak również ich wartości. Pozwala to na ekstrakcję z nich wiedzy o ich relacjach i na tej podstawie wnioskowanie o przynależności do relacji badanego obiektu.

Sieć komparatorów zbudowana jest z warstw komunikujących się między sobą poprzez translator [5]. Centralnym elementem warstwy



Rys. 1. Ogólny schemat sieci komparatorów. Schemat nie jest wykonany przy użyciu UML, a jedynie używa dostępnej tam analogicznej notacji. Użyte skróty: $com_{j,i}$ - komparatory, T_j - translatory. Symbole: owale - komparatory, pionowe pogrubione czarne linie - agregatory, romby - translatory, koła z krzyżykiem w środku - moduły rzutowania.

jest agregator [14], z którym powiązane są pozostałe elementy. Jego zadaniem w poszczególnych warstwach jest wytypowanie najlepszego rozwiązania na potrzeby dalszego przetwarzania. Stanowi on element spajający, gdyż tylko dzięki niemu następuje jednokierunkowa komunikacja pomiędzy poszczególnymi elementami. Do elementów warstwy zaliczają się także komparatory (różne rodzaje), moduły rzutowania oraz translator (zgodnie z rys. 1). Każdy komparator przekazuje wyniki do agregatora. Komparatory nie są ze sobą bezpośrednio połączone. Poszczególne komparatory badają zdefiniowane cechy niezależnie. Przeprowadzają swoje obliczenia współbieżnie oraz zwracają niezależne wyniki. Wynikiem działania pojedynczego komparatora jak również całej sieci jest wektor bliskości zdefiniowany przez elementy zbioru referencyjnego [11] oraz wartości podobieństw pomiędzy poszczególnymi parami obiektów. W przypadku pojedynczego komparatora, wektor wyraża lokalne podobieństwo względem pojedynczej cechy. Wektor bliskości dla sieci komparatorów przedstawia złożenie lokalnych podobieństw z uwzględnieniem reguł ich selekcji oraz doboru siły oddziaływania na podobieństwo globalne.

Metody inteligentne charakteryzują się możliwością adaptacji rozwiązania do postawionego problemu, przy jednoczesnym zachowaniu odpowiedniego poziomu ogólności dającego możliwość generalizacji podczas wnioskowania. Adaptacja tego rodzaju nosi nazwę uczenia algorytmów sztucznej inteligencji. W literaturze wyróżnionych jest kilka rodzajów uczenia. Podział ten stworzony został ze względu na metody

pozyskiwania informacji o obiektach, przypadkach pozytywnych i negatywnych rozpoznawania jak również innych kryteriów (np. pomiaru skuteczności uczenia). W niniejszej publikacji opisana została metoda uczenia struktury sieci komparatorów jako jednego z najważniejszych problemów dotyczących tworzenia sieci komparatorów do rozwiązywania zadań rozpoznawania wzorców dla obiektów złożonych.

Artykuł składa się z 5 sekcji. Pierwsza zawiera informacje wprowadzającą dającą możliwość czytelnikowi zapoznania się z ogólną tematyką związaną z rozpoznawaniem obiektów złożonych w dziedzinie sztucznej inteligencji przy użyciu opisanych we wcześniejszych publikacjach sieciach komparatorów [6, 9]. Sekcja druga zawiera opis metod użytych do realizacji algorytmu uczenia sieci komparatorów, w postaci opisu algorytmów ewolucyjnych stanowiących trzon tej metody. Sekcja trzecia zawiera opis właściwego algorytmu uczenia, zawiera również szczegóły implementacyjne istotne z punktu widzenia wdrożenia tego rozwiązania. Rozdział czwarty zawiera przedstawienie wyników optymalizacji struktury sieci komparatorów skonstruowanej do rozpoznawania znaków drukowanych ograniczonych do cyfr. Sieć taka została opisana w publikacji [16], a we wspomnianej sekcji dokonano porównania skuteczności rozpoznawania bez uczenia oraz po zastosowaniu opisanego algorytmu. Ostatnia sekcja stanowi podsumowanie oraz opis przyszłych prac związanych z opisanym algorytmem.

2 METODY EWOLUCYJNE

Popularną metodą uczenia stosowaną w systemach decyzyjnych jest wykorzystanie metod ewolucyjnych. Ogólnie rzecz ujmując, są to algorytmy wzorowane na otaczającej przyrodzie i naturalnej ewolucji. Teoria ewolucji jest jedną z podstawowych teorii w dziedzinie biologii. Pozwala na opisanie i zrozumienie procesów zachodzących w świecie organizmów żywych i roślin. W ramach tej teorii rozpatrywane jest pojęcie selekcji naturalnej, zaprezentowane przez Darwina w XIX wieku.

Na bazie tych doświadczeń skonstruowano szereg algorytmów naśladujących mechanizmy zachodzące w przyrodzie. Początkowo metody te nosiły różne nazwy, ponieważ rozwijane były niezależnie przez różne ośrodki naukowe. Po pewnym czasie zauważono jednak, iż metody te mają wspólne podłoże i można je uogólnić i ująć w jednym standardowym schemacie postępowania z licznymi wariantami [10].

Przedmiotem działania algorytmu jest populacja rozwiązań zwanych osobnikami. Oznacza się je jako P^t . W szczególnym przypadku P^0 nazywana jest populacją początkową. Listing 1 przedstawia ogólny schemat algorytmu ewolucyjnego z wyróżnieniem typowych składowych [8], które charakteryzują się zadanymi funkcjonalnościami, choć mogą być implementowane na różne sposoby. W dalszej części tej sekcji opisane zostaną poszczególne elementy algorytmu oraz przykładowe metody ich implementacji.

```
Algorytm genetyczny
t=0
Inicjalizacja  $P^0$ 
Ocena  $P^0$ 
while warunek stopu do
     $T^t =$  Reprodukacja( $P^t$ )
     $O^t =$  Operacje Genetyczne( $T^t$ )
    Ocena( $O^t$ )
     $P^{t+1} =$  Sukcesja( $P^t, O^t$ )
    t=t+1
end
```

Algorytm 1: Pseudokod algorytmu ewolucyjnego

Populacja to zbiór osobników reprezentujących przykładowe rozwiązanie zadanego problemu, np. wartości szukanych parametrów zadanej metody. Liczność populacji jest parametrem zadanym. Poszczególne osobniki opisane są poprzez genotyp, który składa się z jednego lub większej liczby chromosomów. Chromosom rozumiany jest jako łańcuch uporządkowanych genów, za pomocą których osobnik jest kodowany. Gen natomiast utożsamiany jest z cechą zwaną atrybutem chromosomu. Poszczególne geny posiadają wartości reprezentujące warianty danej cechy. Wartości te nazywane są allelami. Poszczególne allele tworzą łącznie strukturę zwaną fenotypem, czyli zdekodowaną strukturę stanowiącą konkretne rozwiązanie.

Istnieje kilka metod reprezentacji osobników w postaci chromosomów. Jedną z nich jest reprezentacja binarna. W takim przypadku poszczególne osobniki opisywane są jako chromosomy zakodowane w systemie dwójkowym. Stosuje się różne konstrukcje chromosomów, zarówno pojedyncze, jak i złożone. Zależy to od rozwiązywanego problemu. Poszczególne geny przyjmują wartości 0 lub 1, co jest utożsamiane z występowaniem lub niewystępowaniem danej cechy. Od wyboru metody reprezentacji zależą kolejne etapy przetwarzania i doboru metod

przetwarzania ewolucyjnego. Pewną wadą jest konieczność kodowania i dekodowania rozwiązania, natomiast zaletą jest mocno rozwinięty aparat metod stosowanych na kolejnych etapach przetwarzania.

Populacja początkowa może być tworzona losowo lub z uwzględnieniem wiedzy dziedzinowej (o ile taka jest dana). Przeważnie jednak stosuje się metody randomizowane, które uwzględniają różne rozkłady i parametry. W niektórych przypadkach nie wszystkie kombinacje wartości genotypu mają sens. Może to prowadzić do utworzenia osobników niepoprawnych. Należy zatem uwzględnić walidację poprawności osobnika oraz stosowną procedurę obsługi tego zdarzenia.

Stworzona populacja początkowa poddawana jest ocenie dopasowania. Polega ona na wyliczeniu wartości osiąganych przez optymalizowaną metodę, przy założeniu użycia parametrów reprezentowanych przez danego osobnika. Wartości te zapamiętywane są przez cały cykl życia populacji. W tym miejscu algorytmu zaczyna się właściwy cykl ewolucyjny. W pierwszej kolejności rozpatrywany jest warunek stopu. Jeśli nie jest on spełniony, dokonywana jest reprodukcja. Proces ten bazuje na selekcji osobników, które będą uczestniczyły w operacjach genetycznych generujących nowe rozwiązania. Bazując na terminologii ewolucyjnej, proces ten jest wskazaniem rodziców, którzy mają największe prawo do potomstwa. Ogólna zasada zbieżna z selekcją naturalną określa, iż są to rodzice o najlepszym współczynniku dopasowania. Można rozpatrywać różne warianty reprodukcji, które zależą od liczby osobników w populacji przejściowej, czyli tej, która stanowi wynik reprodukcji. W przypadku liczby osobników zbliżonej do liczby osobników populacji wyjściowej rozpatrywać można metodę wyboru z powtórzeniami. Umożliwia to pojawienie się wielu kopii tego samego osobnika. Wyróżnia się kilka metod selekcji osobników. Najbardziej popularną jest metoda ruletki, której nazwa pochodzi od analogii do gry. Polega ona na symbolicznym wyznaczeniu pól na tarczy ruletki proporcjonalnej wielkości do osiągniętej wartości funkcji dopasowania danego osobnika. Biorąc pod uwagę wartości dopasowania wszystkich osobników populacji, dla i -tego osobnika prawdopodobieństwo selekcji można zapisać w postaci następującego wzoru:

$$P_s(ch_i) = \frac{f_{fit}(ch_i)}{\sum_{j=1}^K f_{fit}(ch_j)} \quad (1)$$

gdzie f_{fit} jest funkcją dopasowania, a K - wielkością populacji. Zgodnie z zasadą ruletki następuje losowanie, a podział tarczy ruletki (wy-

ciniek koła $v(ch_i)$ realizowany jest przy użyciu prawdopodobieństw podanych we wzorze (1) w następujący sposób:

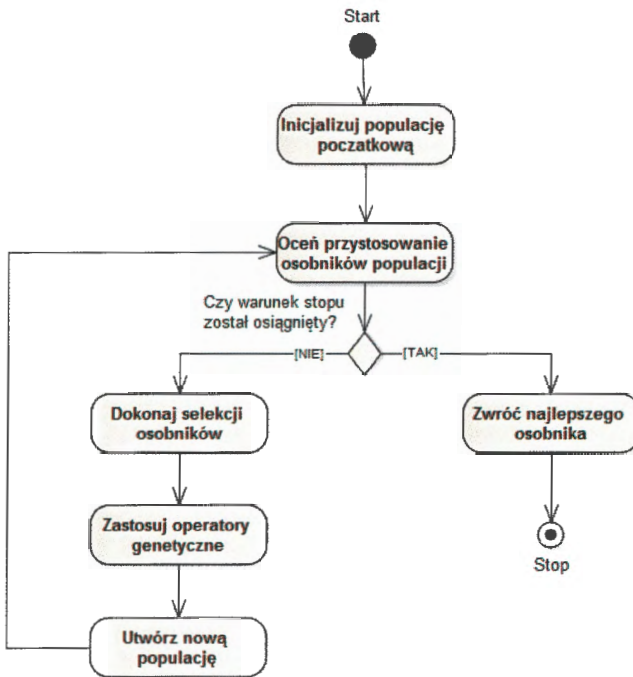
$$v(ch_i) = P_s(ch_i) * 100\% \quad (2)$$

Populacja T^t uzyskana w procesie reprodukcji jest następnie poddawana operacjom genetycznym, na które standardowo składają się krzyżowanie i mutacja [13]. Pierwsza operacja jest zdecydowanie istotniejsza ze względu na prawdopodobieństwo jej wykonania, które dla niej przyjmuje się jako wartość z zakresu $[0.5, 1]$. Krzyżowanie jest wymianą genów pomiędzy osobnikami, w wyniku czego powstaje nowy osobnik (potomek). Mutacja natomiast stanowi operację na pojedynczym genie, która zachodzi z dużo mniejszym prawdopodobieństwem. Zazwyczaj określa się ją jako wartość z przedziału $[0, 0.1]$, co jest zgodne z intuicją zaczerpniętą z biologii.

Krzyżowanie rozpoczyna się wyborem par osobników, które podlegają wymianie genów. Istnieje wiele rodzajów tej operacji. Jedną z łatwiejszych, a zarazem mocno rozpowszechnionych metod jest krzyżowanie jednopunktowe. Polega ono na losowym wyborze punktu dla poszczególnych chromosomów, w którym następuje zamiana alleli. Jako wynik operacji powstaje jeden nowy osobnik o wartościach genów do punktu podziału identycznych z pierwszym rodzicem, a od punktu podziału identycznych z drugim rodzicem. Należy zwrócić uwagę na fakt, iż operator krzyżowania jest ściśle związany z wybraną reprezentacją. Opisany przykład łatwo stosuje się do reprezentacji binarnej, lecz nie zawsze do reprezentacji rzeczywistoliczbowej. Jednakże w literaturze opisanych zostało bardzo wiele przykładów operatorów, odpowiednich dla różnych reprezentacji [13, 8].

Mutacja wywołuje zmianę wartości genu chromosomu na inną. Operacja ta, w zależności od reprezentacji, może przybierać różne formy. Jak wcześniej wspomniano, dzieje się to przy uwzględnieniu prawdopodobieństwa, które jest stosunkowo niskie. Istnieje kilka wariantów mutacji, które sprowadzają się do wyboru genów do mutacji. Jednym z rozwiązań jest losowy wybór dla każdego genu, pozwalający stwierdzić, czy dany gen ma być poddany mutacji.

Po wykonaniu operacji genetycznych następuje ewaluacja nowych osobników oraz sukcesja. Polega ona na stworzeniu nowej populacji P^{t+1} z P^t i O^t , czyli z poprzedniej populacji wyjściowej oraz populacji powstałej po operacjach genetycznych. Analogicznie jak w przypadku reprodukcji, dokonywana jest selekcja. Należy pamiętać, iż nie musi to być taka sama metoda selekcji, jak w innych miejscach algorytmu.



Rys. 2. Diagram UML ogólnego algorytmu ewolucyjnego w postaci diagramu aktywności.

Istnieje również kilka podejść do konstrukcji zbioru, z którego dokonywana jest selekcja. Można połączyć oba zbiory i wybierać osobniki z całego zbioru lub utrzymać podział w założonych proporcjach, np. 30% z pierwszego zbioru oraz 70% z drugiego.

W tym miejscu algorytmu kończy się pojedynczy cykl ewolucji. Wykonywany jest on wielokrotnie, aż do osiągnięcia warunku stopu. Warunek ten może być zdefiniowany bezwzględnie, w postaci zadanej liczby iteracji (epok) lub względnie, w postaci liczby ostatnich iteracji, w których nie nastąpiła poprawa rozwiązania. Możliwe są też inne opisane w literaturze metody zakończenia przetwarzania. Po zakończeniu przetwarzania iteracyjnego zwracany jest wynik końcowy w postaci najlepszego osobnika, tzn. takiego, który uzyskał najwyższą wartość funkcji dopasowania. Szczegółowy schemat algorytmu ewolucyjnego został przedstawiony na rys. 2.

3 ZAGADNIENIE UCZENIA STRUKTURY SIECI KOMPARATORÓW

Niniejsza sekcja zawiera opis jednego z dwóch kluczowych zagadnień dotyczących efektywności sieci komparatorów, którym jest dobór cech przetwarzanych przez sieć oraz struktura sieci. Zagadnienie to dotyczy bezpośrednio obiektu, opisu jego cech, relacji i zależności. Obiekty usystematyzowane są poprzez opis w ontologii [17], który operuje na pojęciach i relacjach między nimi. Definiuje również powiązania pomiędzy instancjami a konkretnymi pojęciami. W tym przypadku ontologia określa zbiór cech możliwych do wykorzystania przy konstrukcji sieci. W wielu sytuacjach są one redundantne lub obejmują cechy mniej znaczące z punktu widzenia problemu, jaki się ma rozwiązywać. W niniejszej publikacji sieć komparatorów rozpatrywana jest na potrzeby rozpoznawania oraz identyfikacji obiektów przy użyciu podobieństwa. Optymalny wybór cech dla danego problemu jest zatem problemem minimalnego podzbioru atrybutów, jednoznacznie identyfikującego obiekt lub jego klasę. W przypadku problemu rozpoznawania będzie to podzbiór identyfikujący typ obiektów, natomiast w przypadku identyfikacji instancji będą to indywidualne cechy opisujące konkretne obiekty. Minimalnych podzbiorów atrybutów może być wiele i mogą one być równoważne z punktu widzenia jakości. W literaturze problem taki znany jest pod nazwą reduktu, zwłaszcza w teorii zbiorów przybliżonych i eksploracji danych [7]. W opisywanym przypadku cechy mają bezpośredni wpływ na schemat sieci, a zatem również na jej efektywność. Problem projektowania i konstrukcji sieci został opisany w artykule [22]. Proces doboru struktury sieci daje się przekształcić w problem wyznaczenia istotności cech, a w szczególności możliwości ich wyrugowania. Dlatego też jest to problem, do którego zastosowano uczenie maszynowe zrealizowane poprzez użycie algorytmów ewolucyjnych.

Ten rodzaj zagadnienia stanowi uczenie z nauczycielem, tzn. musi istnieć zbiór uczący oraz zbiór testowy, na których można przeprowadzić procedurę uczenia i walidację wyuczonych parametrów. Zbiór ten będzie dotyczył konkretnego problemu, dla jakiego sieć ma zostać nauczona struktury. Jeśli będzie to problem klasyfikacji lub identyfikacji, to będzie zawierał przykłady obiektów, które stanowią obiekty wejściowe sieci oraz przypisane im poprawne etykiety.

Procedura uczenia zakłada istnienie zbioru cech dotyczących przetwarzanych obiektów. Zadaniem procedury jest wybór minimalnej licz-

by cech oraz wskazanie rozlokowania komparatorów w warstwach, zapewniając prawidłowe działanie sieci przy ustalonych rodzajach agregatorów lokalnych, translatorach oraz liczbie warstw.

Do przeprowadzenia procedury uczącej zastosowany został algorytm genetyczny. Osobniki przedstawiają poszczególne konfiguracje struktury sieci, tzn. cechy przypisanej wraz z komparatorem do konkretnej warstwy. W algorytmie została zastosowana reprezentacja całkowitoliczbowa zakodowana w postaci chromosomu składającego się z tylu części, ile w przestrzeni rozważań istnieje komparatorów. Pojedynczy chromosom ma zatem postać: $\mathbf{x} = \langle (i, k) \rangle$, gdzie i jest identyfikatorem komparatora oraz $k \in \{0, 1, \dots, n - 1\}$, gdzie n jest liczbą warstw sieci. Ze względu na to, że w warstwie wyjściowej nie mogą znajdować się komparatory, to maksymalna wartość jest o jeden mniejsza niż liczba warstw. Wartość k reprezentuje numer warstwy, przy czym 0 oznacza brak wyboru komparatora do którejkolwiek warstwy. Kolejność w warstwie nie ma znaczenia ze względu na współbieżne przetwarzanie. Komparator może być przypisany co najwyżej jednej warstwy. Jeśli miałby być użyty ten sam komparator (pod względem funkcjonalnym), powinien on być reprezentowany przez unikalny identyfikator.

Na początku przetwarzania losowana jest populacja początkowa. Należy zwrócić uwagę na fakt, iż przyjęta postać chromosomu umożliwia wygenerowanie nieprawidłowej postaci osobnika. Dlatego też walidacja poprawności następuje przy tworzeniu wartości chromosomu. W przypadku losowania populacji początkowej, osobniki losowane są aż do momentu powstania pełnej populacji poprawnych osobników. Rozmiar populacji jest parametrem algorytmu. Niepoprawny osobnik to taki, z którego nie da się skonstruować poprawnej sieci, np. $\langle (1, 0), \dots, (n - 1, 0) \rangle$, lub zawierający puste warstwy.

W każdej iteracji (zwanej epoką) następują operacje reprodukcji, zastosowania metod genetycznych, ewaluacji oraz sukcesji. Schemat zastosowanego algorytmu genetycznego zgodny jest z opisaną wcześniej teorią metod ewolucyjnych.

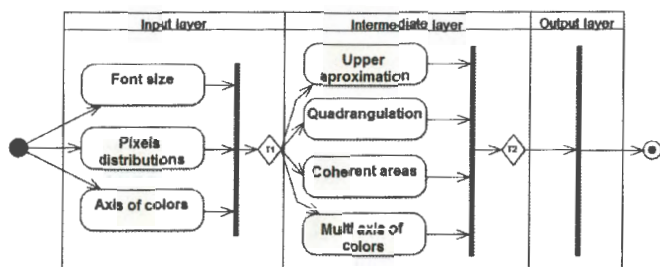
Reprodukcja została zrealizowana z użyciem selekcji turniejowej, która przyjmuje parametr l oznaczający liczbę osobników losowanych do turnieju. Liczba osobników powstałych w procesie reprodukcji jest również parametrem algorytmu. W wyniku reprodukcji powstaje lokalna populacja, która poddawana jest operacjom krzyżowania i mutacji. Do krzyżowania losowane są dwa osobniki. Procedura polega na

wylosowaniu punktu odcięcia, w którym zamienione zostają elementy dwóch chromosomów. Od pozycji zerowej do wylosowanej pozycji pozostają geny pierwszego chromosomu, a od wylosowanej pozycji do końca geny drugiego chromosomu. Operacja ta może jednak prowadzić do powstania niepoprawnego osobnika. W takim przypadku na wyjściu zwrócony zostaje pierwszy z wejściowych chromosomów. Operacja mutacji polega na dodaniu do bieżącej wartości genu losowej liczby naturalnej oraz wykonaniu operacji $\text{mod } n$, gdzie n jest liczbą warstw. Mutacja jest wykonywana z pewnym przyjętym prawdopodobieństwem dla każdego genu chromosomu. Jedną z istotniejszych części algorytmu, indywidualnie dostosowaną do specyfiki sieci komparatorów, jest funkcja ewaluacji (oceny dopasowania populacji). Ogólna idea polega na dynamicznym zbudowaniu sieci komparatorów, a następnie uruchomieniu przetwarzania. Do wyliczenia wartości dopasowania brane są pod uwagę zarówno wyniki końcowe (np. rozpoznawania), jak i koszt wykonania obliczenia przez komparator. Ta ostatnia wartość wyliczana jest na podstawie kosztu jednostkowego uruchomienia komparatora oraz liczności zbioru referencyjnego w danej warstwie. Ogólna postać funkcji dopasowania wygląda następująco:

$$f_X(ch) = \left(\sum_U f1score(x, ch) \right) - \left(\frac{\sum_{i=1}^{n-1} \sum_{j=1}^m f_{ij}^{cost} * |ref_i|}{|C| * |ref|} - \alpha \right), \quad (3)$$

gdzie ch jest chromosomem reprezentującym daną strukturę sieci, X jest zbiorem obiektów wejściowych sieci, stanowiących zbiór uczący, C jest zbiorem wszystkich komparatorów rozważanych w danym problemie, f_{ij}^{cost} jest funkcją kosztu jednostkowego wykonania obliczeń przez komparator j -ty w warstwie i -tej. Wartości tej funkcji są z przedziału $(0, 1)$, ref_i jest podzbiorem zbioru ref występującego jako zbiór referencyjny w i -tej warstwie, a α jest wartością dodatnią bliską zero. W ten sposób wartość funkcji kary (człon w drugim nawiasie funkcji (3)) jest zawsze mniejsza od 1, dzięki czemu wyżej promowane są poprawnie zidentyfikowane elementy, niż prosta struktura sieci, choć ten drugi element też jest istotny. W podanym wzorze $f1score$ wyraża skuteczność wykonania rozpoznawania (lub identyfikacji) dla danego osobnika na zbiorze uczącym.

Sukcesja wykonywana jest w analogiczny sposób, co reprodukcja, tj. przy użyciu selekcji turniejowej. W tym przypadku obie populacje są łączone i z nich wybierane są osobniki do turniejów. Turniej rozgrywany jest tyle razy, ile wynosi liczebność populacji docelowej. Liczebność



Rys. 3. Schemat sieci komparatorów zaprojektowanej do rozpoznawania znaków drukowanych wykorzystanych w eksperymencie opisanym w publikacji [16].

populacji jest również parametrem metody. Warunkiem zakończenia algorytmu jest zadana liczba epok, w których nie następuje poprawa rozwiązania.

Do wdrożenia opisanej procedury może zostać użyta metoda wielokrotnego próbkowania (ang. re-sampling). Polega ona na losowym podziale dostępnych danych na zbiór uczący i testowy przeważnie w proporcjach odpowiednio $\frac{1}{3}$ i $\frac{2}{3}$. Na zbiorze uczącym przeprowadzane jest uczenie, tzn. poszukiwanie suboptymalnej struktury sieci, natomiast na drugim zbiorze rozwiązanie jest testowane. Procedura ta powtarzana jest k krotnie, gdzie $k \geq 10$.

4 OSIĄGNIĘTE WYNIKI

Do zobrazowania wyników wykorzystany został eksperyment wykonany w ramach publikacji [16]. Polegał on na rozpoznawaniu znaków drukowanych w dwóch wariantach: samych cyfr oraz znaków występujących w tablicach rejestracyjnych pojazdów zarejestrowanych w Polsce. Eksperyment źródłowy wykonany został przy użyciu sieci komparatorów pokazanej na Rys. 3. Do budowy sieci wykorzystano komparatory zdefiniowane w Tabeli 1.

Celem eksperymentu z niniejszej publikacji było zbadanie w jaki sposób i w jakim stopniu algorytm uczenia struktury wpływa na uzyskiwane wyniki oraz pośrednio złożoność obliczeniową i czasową.

Na potrzeby niniejszego artykułu wykorzystany został pierwszy wariant opisanego wcześniej eksperymentu, tzn. rozpoznawanie cyfr drukowanych. Tabela 2 zawiera wyniki uzyskane dla architektury sieci użytej w poprzednich badaniach, jak również dla wybranych dwóch

Tabela 1. Zbiór komparatorów wykorzystany do budowy sieci rozpoznającej znaki drukowane

Id	Komparator	Koszt	Opis działania
1	Axis of colors	0.10	Analiza współrzędnej pionowej i poziomej
2	Coherent areas	0.60	Analiza spójności obszarów
3	Quadrangulation	1.00	Podział znaku na n części i niezależna analiza
4	Font size	0.10	Analiza wysokości czcionki, liczby pikseli
5	Upper aproximation	0.30	Analiza aktywności obszarów obrazu
6	Multi axis of colors	0.20	Analiza wielu współrzędnych pionowych i poziomych
7	Pixel distributions	0.60	Analiza rozkładu pikseli w obrazie

Tabela 2. Wyniki uzyskane we wcześniejszym eksperymencie oraz w obecnym. Oznaczenia: PR - precyzja, RE - czułość, F1 - F1score. Sufiksy: 1 - poprzedni eksperyment dla architektury sieci: 1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 2, 4 \rightarrow 1, 5 \rightarrow 2, 6 \rightarrow 2, 7 \rightarrow 1, 2 - nowy eksperyment dla architektury sieci: 1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 2, 4 \rightarrow 2, 5 \rightarrow 2, 6 \rightarrow 1, 7 \rightarrow 2, 3 - nowy eksperyment dla architektury sieci: 1 \rightarrow 1, 2 \rightarrow 0, 3 \rightarrow 0, 4 \rightarrow 2, 5 \rightarrow 1, 6 \rightarrow 0, 7 \rightarrow 2 gdzie $x \rightarrow y$ oznacza: x - identyfikator kompartora. y - numer warstwy sieci

Znak	PR1	RE1	F1-1	PR2	RE2	F1-2	PR3	RE3	F1-3
0	0.96	1.00	0.98	1.00	0.96	0.98	0.84	0.77	0.80
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.89	1.00	0.94	0.96	0.96	0.96	1.00	1.00	1.00
3	0.95	0.83	0.89	0.96	0.96	0.96	1.00	1.00	1.00
4	0.96	0.96	0.96	1.00	1.00	1.00	0.96	0.86	0.90
5	0.88	0.96	0.92	1.00	1.00	1.00	0.96	0.96	0.96
6	0.76	0.92	0.83	0.96	0.96	0.96	0.92	0.92	0.92
7	1.00	1.00	1.00	1.00	0.89	0.94	1.00	1.00	1.00
8	1.00	0.96	0.98	1.00	1.00	1.00	0.75	0.72	0.73
9	0.94	0.67	0.78	0.96	0.96	0.96	0.92	0.92	0.92
Ogólnie	0.93	0.82	0.87	0.98	0.97	0.98	0.93	0.91	0.92

wyuczonych struktur poprzez zaprojektowany algorytm. Jak widać algorytm zaproponował rozwiązanie o lepszej efektywności od pierwotnego, dla którego odnotowano wzrost precyzji o ponad 5%, czułości o ponad 15% a F1score o 5%, jednakże czas wykonywania rozpoznawania wzrósł o ponad 33%. Inne zaproponowane rozwiązanie utrzymało precyzję na tym samym poziomie, lecz zwiększyło czułość o ok. 9%. Jednakże czas procesu rozpoznawania został skrócony o $\frac{1}{3}$ długości procesu pierwotnego. Czas przetwarzania jest ściśle zależny od złożoności obliczeniowej i czasowej, a one od liczby użytych komparatorów. Rozwiązanie oznaczone w Tabeli 2 sufiksem 2 zwiększyło czas przetwarzania, gdyż użyło wszystkich dostępnych komparatorów z rozpatrywanego zbioru oraz w pierwszej warstwie ulokowała ich mniejszą liczbę. W takim przypadku prawdopodobnie większa liczba

elementów zbioru referencyjnego została wyznaczona do przetwarzania w drugiej warstwie. W tej sytuacji komparatory o większym koszcie musiały wykonać więcej porównań. Analogicznie w przypadku oznaczonym sufiksem 3. W tym przypadku kilka komparatorów zostało wyrugowanych i nie wystąpiło w żadnej z warstw. Natomiast rozkład pozostałych komparatorów pomiędzy warstwami w większym stopniu preferował warstwę pierwszą (w sensie procentowego udziału spośród wszystkich użytych komparatorów), przy czym przydzielone tam komparatory miały stosunkowo niskie współczynniki kosztu.

5 PODSUMOWANIE

Przedstawione rozwiązanie daje satysfakcjonujące wyniki mogące w znaczącym stopniu podnieść efektywność budowanego rozwiązania problemu rozpoznawania obiektów. Jednakże należy pamiętać również o pewnych założeniach, których spełnienie jest wymagane. Przede wszystkim algorytm uczenia w obecnej postaci przyjmuje z góry podaną maksymalną liczbę warstw sieci. Powoduje to pewne ograniczenia i zawężenie poszukiwanych rozwiązań do pewnego podzbioru wszystkich możliwych, przy czym ten wybór jest stricte arbitralny i nie podlega modyfikacjom dokonywanym przez algorytm. To niewątpliwie jeden z elementów, który wymaga dalszych badań i modyfikacji.

Inny element wymagający omówienia to założenie o istnieniu dobrze zdefiniowanego zbioru referencyjnego oraz zbioru testowego. Wyniki działania algorytmu uczenia struktury zależą bardzo ściśle od poprawności i jakości zbioru referencyjnego oraz losowości zbioru treningowego i testującego.

Podsumowując otrzymane wyniki są bardzo obiecujące i wykazują bardzo duży potencjał algorytmu. W ramach przyszłych prac należy przebadać inne postacie funkcji dopasowania, kładące w różnym stopniu nacisk na wybrane aspekty optymalizacji, np. minimalizacja użytych komparatorów, ograniczenie procentowe zbioru referencyjnego w pierwszej warstwie, itp. Dodatkowo należy opracować możliwość iteracyjnego uczenia, w którym optymalizowana byłaby zarówno struktura jak i zbiór referencyjny.

Literatura

1. J. Kacprzyk. *Multistage Fuzzy Control: A Model-based Approach to Fuzzy Control and Decision Making*. John Wiley Sons. Limited, 2012.

2. J.M. Medina et al., Generalized fuzzy comparators for complex data in a fuzzy object-relational database management system. *Information Processing and Management of Uncertainty in Knowledge-Based Systems Applications Vol. 81*, Springer-Verlag, Berlin, 126-136
3. A. Janusz, D. Slezak, H.S. Nguyen, Unsupervised Similarity Learning from Textual Data, *Fundamenta Informaticae* vol. 119, 2012
4. L. Chmielewski, J. L. Kulikowski, A. Nowakowski, *Biocybernetyka i inżynieria biomedyczna*. Tom 8, Obrazowanie biomedyczne
5. L. Sosnowski and D. Slezak, "Networks of compound object comparators," In *Proceedings of the FUZZ-IEEE, India, Hyderabad 2013*. pp. 1-8.
6. Ł. Sosnowski, "Framework of Compound Object Comparators," *Intelligent Decision Technologies*, 2015.
7. Z. Pawlak, A. Skowron, *Rough Sets: Some Extensions*, *Information Sciences*. vol. 177, no. 1, pp. 28-40, 2007
8. J. Arabas, *Wykłady z algorytmów ewolucyjnych*, Wydawnictwo WNT, 2004, Warszawa
9. Ł. Sosnowski "Applications of comparators in data processing systems," *Technical Transactions, Automatic Control*, pp. 81-98, 2013.
10. J. Łęski, *Systemy neuronowo-rozmyte*, Wydawnictwa Naukowo-Techniczne 2008
11. Ł. Sosnowski and D. Ślęzak, "Comparators for Compound Object Identification," in *Proc. of RSFDGrC 2011*, ser. *Lecture Notes in Computer Science*, vol. 6743, 2011, pp. 342-349.
12. L. Sosnowski and D. Slezak. "Fuzzy set interpretation of comparator networks." in *Pattern Recognition and Machine Intelligence - 6th International Conference, PRMI 2015*. Warsaw, Poland, June 30 - July 3, 2015, *Proceedings*, 2015, pp. 345-353. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19941-2_33
13. T.D. Gwiazda, *Algorytmy genetyczne: kompendium. Operator krzyżowania dla problemów numerycznych*, t. 1, Wydawnictwo Naukowe PWN, 2007
14. Ł. Sosnowski, A. Pietruszka, and S. Łazowy, "Election algorithms applied to the global aggregation in networks of comparators," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. *Annals of Computer Science and Information Systems*, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 2. IEEE, 2014, pp. pages 135-144. [Online]. Available: <http://dx.doi.org/10.15439/2014F494>
15. D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, ser. *Artificial Intelligence*. Addison-Wesley, 1989. [Online]. Available: http://books.google.pl/books?id=3_RQAAAAAAAJ
16. Ł. Sosnowski, A. Pietruszka, Characters recognition based on network of comparators. In A. Myliski, editor, *Techniki informacyjne teoria i zastosowania*, volume 4, pages 123134. IBS PAN, 2014.
17. S.Staab, A. Maedche, *Knowledge Portals: Ontologies at Work*, *AI Magazine*. vol. 22, no. 2, pp. 63-75
18. E. Elkind, J. Lang, and A. Saffidine, "Choosing collectively optimal sets of alternatives based on the condorcet criterion." in *IJCAI*, T. Walsh, Ed. *IJCAI/AAAI*, 2011, pp. 186 191. [Online]. Available: <http://dblp.uni-trier.de/db/conf/ijcai/ijcai2011.htmlElkindLS11>
19. D. Nohlen, *Prawo wyborcze i system partyjny: o teorii systemów wyborczych*. Scholar, 2004. [Online]. Available: <http://books.google.pl/books?id=ckODAAAACAAJ>
20. L. Polkowski, *Approximate Reasoning by Parts: An Introduction to Rough Mereology*, ser. *Intelligent Systems Reference Library*. Springer, 2011.
21. A. Gomolinska and M. Wolski, "Rough inclusion functions and similarity indices." in *CSP*, ser. *CEUR Workshop Proceedings*, M. S. Szczuka, L. Czaja, and M. Kacprzak. Eds., vol. 1032. CEUR-WS.org, 2013, pp. 145-156. [Online]. Available: <http://dblp.uni-trier.de/db/conf/csp/csp2013.html#GomolinskaW13>

22. Ł. Sosnowski, D. Ślęzak. How to Design a Network of Comparators. Brain and Health Informatics, 2013, pp. 389-398

NETWORKS OF COMPOUND OBJECTS COMPARATORS - STRUCTURE
LEARNING

Abstract. The paper presents the algorithm of learning the networks of compound objects comparators in case of optimizing the network structure for increasing efficiency of recognition. The subject is about similarity based reasoning used for recognition and identification of compound objects.

Keywords: networks of comparators, similarity of compound objects, learning of networks of comparators, recognition and identification.

the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million (15.5% of the population).

There is a growing awareness of the need to address the needs of older people, and the Government has set out a strategy for doing so in the White Paper on *Ageing Better: A New Strategy for Older People* (Department of Health 2000). This paper sets out the following objectives:

- (i) to improve the health and well-being of older people;
- (ii) to improve the opportunities for older people to participate in society;
- (iii) to improve the opportunities for older people to live independently;
- (iv) to improve the opportunities for older people to live in their own homes.

The White Paper also sets out a number of key principles that should guide the development of policies and services for older people:

- (i) older people should be seen as individuals, not just as members of a group;
- (ii) older people should be seen as active citizens, not just as passive recipients of services;
- (iii) older people should be seen as a resource, not just as a burden;
- (iv) older people should be seen as a diverse group, not just as a homogeneous group.

The White Paper also sets out a number of key areas for action, including:

- (i) health and well-being;
- (ii) participation in society;
- (iii) living independently;
- (iv) living in their own homes.

The White Paper also sets out a number of key actions that should be taken to achieve these objectives:

- (i) to improve the health and well-being of older people, the Government will: (a) invest in research and development to improve the health and well-being of older people; (b) invest in services that help older people to live healthily and independently; (c) invest in services that help older people to participate in society.

(ii) to improve the opportunities for older people to participate in society, the Government will:

- (a) invest in services that help older people to participate in society;
- (b) invest in services that help older people to live independently;
- (c) invest in services that help older people to live in their own homes.

(iii) to improve the opportunities for older people to live independently, the Government will:

- (a) invest in services that help older people to live independently;
- (b) invest in services that help older people to live in their own homes.

(iv) to improve the opportunities for older people to live in their own homes, the Government will:

