

Archipelag sztucznej inteligencji. Część III

Ryszard Tadeusiewicz

Wprowadzenie

W dwóch poprzednich numerach miesięcznika „Napędy i Sterowanie” opisywałem, czym jest sztuczna inteligencja (AI). Dla ożywienia narracji porównałem sztuczną inteligencję do archipelagu wysp, a poszczególne metody AI opisałem jako wyspy (rozumiane oczywiście metaforycznie, ale na zasadzie umowy pisane bez cudzysłowu). W grudniowym numerze NiS (z ubiegłego roku) opisałem w ten sposób metody symboliczne, sieci neuronowe i systemy ekspertowe. W numerze styczniowym prezentowałem metody zbiorów rozmytych i logiki rozmytej, zbiory przybliżone i rozpoznawanie obrazów (*pattern recognition*). Dzisiaj kilka kolejnych metod – opisywanych jako wyspy, ale zaprezentowanych solidnie poprzez podanie najważniejszych cech rozważanych metod. Jako pierwsze omówimy metody analizy skupień.

Analiza skupień (cluster analysis)

Metaforyczny obraz wyspy, symbolizującej metody analizy skupień, przedstawia rysunek 1.

Metody analizy skupień są w pewnym sensie odwrotne w stosunku do omawianych w poprzednim artykule (ze stycznia br.) metod rozpoznawania obrazów. Przy rozpoznawaniu obrazów mamy z góry ustalone pewne kategorie, a algorytm *pattern recognition* ma dla nowego obiektu ustalić, do jakiej kategorii można go zaliczyć. Natomiast przy analizie skupień mamy do dyspozycji dane wielu obiektów, a algorytm *cluster analysis* ma ustalić, do ilu skupień (odpowiedników kategorii) można te obiekty przyporządkować i jak podzielić rozważane obiekty pomiędzy wykryte skupienia (rys. 2).

Żartobliwa ilustracja tego, czym jest skupienie pokazana jest na rysunku 3.

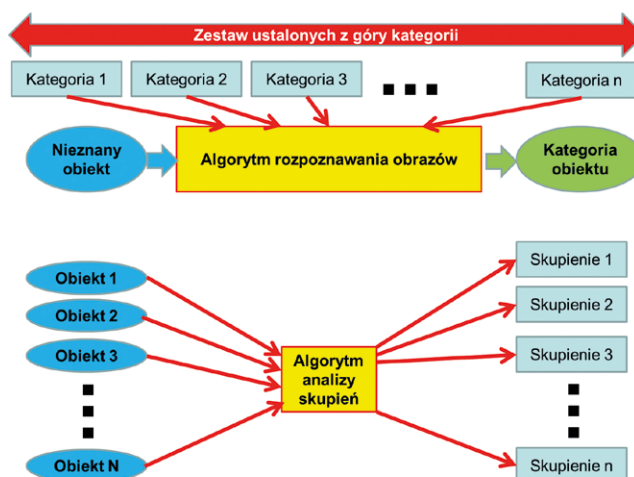
Dość często metody analizy skupień są stosowane do tego, by prawidłowo zdefiniować zadanie rozpoznawania [1].

W metodach analizy skupień chodzi zwykle o wykrycie (i wykorzystanie!) faktu, że pewne obiekty są do siebie **podobne**. Zwykle poznaje się to po tym, że w tak zwanej przestrzeni cech (omawianej w poprzednim artykule) ich reprezentacje są blisko siebie – tak jak te pingwiny na rysunku 3 albo palmy występujące w dwóch skupieniach na rysunku 1. Takie blisko siebie położone obiekty można zgrupować, a potem traktować jednakowo we wszystkich procesach wykonywanych na tych danych – na przykład przy podejmowaniu decyzji. Na kilku kolejnych rysunkach pokazano etapy takiego grupowania. Na rysunku 4 pokazano stan początkowy, kiedy każdy obiekt jest traktowany indywidualnie i żadnych skupień się nie wykrywa.

Zastosowanie metody analizy skupień polega na tym, żeby umieścić obiekty położone blisko siebie w jednym skupieniu (klastrze). Owo skupienie w dalszej analizie będzie zastępowało



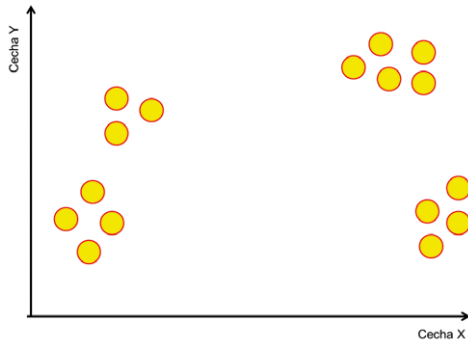
Rys. 1. Wyspa symbolizująca metody analizy skupień



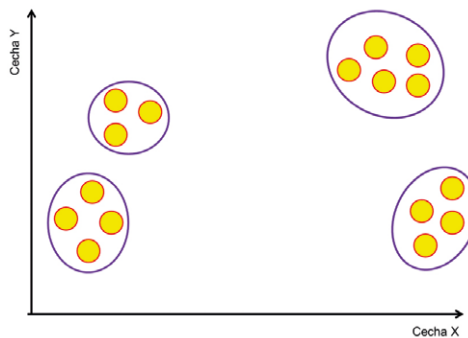
Rys. 2. Porównanie działania rozpoznawania obrazów (u góry) i analizy skupień (u dołu). Ważna uwaga przy analizie tego rysunku: liczba N (obiektów wyznaczających skupienia) jest znacznie większa od liczby n wyznaczonych skupień



Rys. 3. Przykładowe skupienie



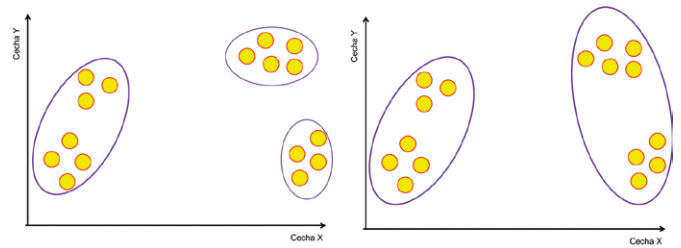
Rys. 4. Punkt wyjścia do analizy skupień. Obiekty nie są pogrupowane



Rys. 5. Pierwszy etap grupowania - wykryto 4 skupienia

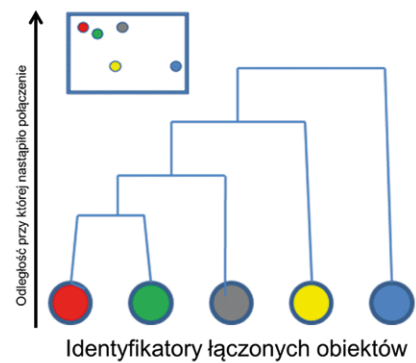
zawarte w nim pojedyncze obiekty (rys. 5). Korzyść jest taka, że skupień jest znacznie mniej niż obiektów.

Kluczowym zagadnieniem jest określenie odległości, która pozwala zaliczyć obiekty do jednego skupienia. Im mniejsza jest ta odległość, tym lepiej skupienie może zastępować w dalszych analizach zawarte w nim obiekty. Z drugiej jednak strony liczba skupień jest tym mniejsza, im ta odległość będzie większa (rys. 6).



Rys. 6. Zmiana liczby skupień na skutek zmiany odległości, przy której następuje łączenie obiektów

Żeby zaznaczyć, przy jakiej odległości nastąpiło połączenie określonych obiektów, efekty pracy analizy skupień często przedstawia się w postaci tak zwanego dendrytu (rys. 7).

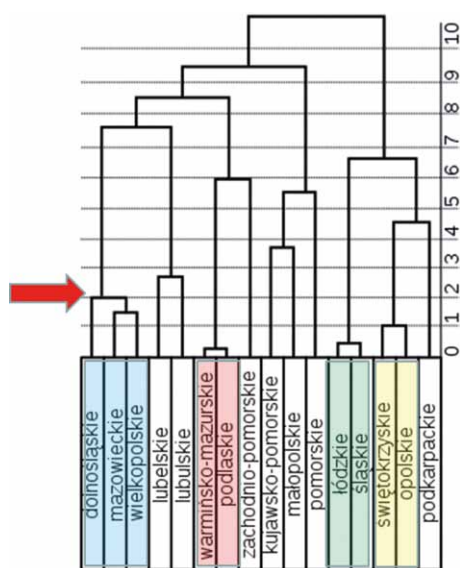


Rys. 7. Prezentacja wyniku działania analizy skupień w postaci dendrytu. W lewym górnym rogu pokazano rozmieszczenie obiektów w przestrzeni cech

Jak widać na rysunku – im większą odległość dopuścimy jako umożliwiającą łączenie obiektów w jedno skupienie, tym bardziej liczna jest zawartość tego skupienia. Na końcu, przy przyjęciu bardzo dalekiego zasięgu procesu grupowania, wszystkie obiekty zlewają się w jedno skupienie, co oczywiście

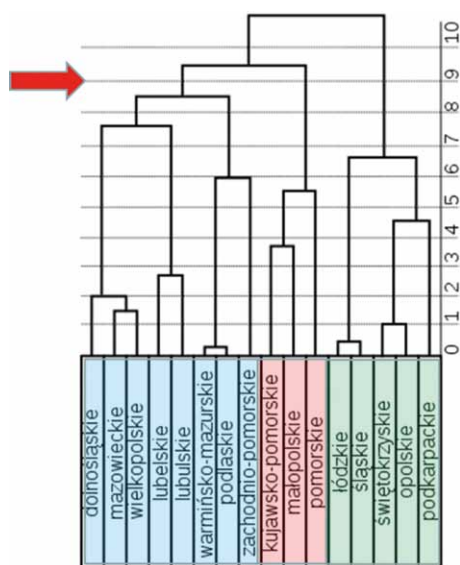
jest całkiem nieprzydatne. Żeby uzyskać poprawne grupowanie, trzeba przyjąć (zwykle arbitralnie, bo tu nie ma dobrych procedur automatycznych) jakiś poziom „cięcia” dendrytu. Na rysunkach 8 i 9 pokazane są skutki wyboru niskiej i wysokiej wartości cięcia. W badaniach, których te rysunki dotyczą, przestrzenią cech był zbiór danych społeczno-ekonomicznych poszczególnych polskich województw. Ich nazwy są wypisane na prostokątach symbolizujących te obiekty w dolnej części rysunku, zaś dendryt powstały podczas ich grupowania z coraz większym dozwolonym dystansem narysowano powyżej.

Gdy przyjmie się małą wartość wysokości cięcia (czerwona strzałka) – powstanie 7 skupień, pokazanych na rysunku 8.



Rys. 8. Wynik analizy skupień przy małej wartości wysokości cięcia dendrytu

W poszczególnych skupieniach zebrane zostały obiekty rzeczywiście bardzo do siebie podobne, ale do pewnych zastosowań tak duża liczba skupień jest niewygodna. Możemy więc wysokość cięcia dendrytu podnieść (rys. 9).



Rys. 9. Wynik analizy skupień przy dużej wartości wysokości cięcia dendrytu

Otrzymujemy wtedy zaledwie trzy skupienia województw – ale zdecydowanie sensownie pogrupowanych.

Skupienia mogą grupować większe lub mniejsze ilości danych, co zilustrowano na rysunku 10, prezentując zamiast grup drzew wyższe albo niższe pojedyncze drzewa – właśnie reprezentujące owe skupienia i swoją wysokością sygnalizujące liczebność odpowiednich skupień.



Rys. 10. Metafora skupień gromadzących różne liczby obiektów

Korzyść, jaką można odnieść dzięki przejściu od rozpatrywania danych oryginalnych (rozproszonych) do rozpatrywania danych pogrupowanych w skupienia – zilustrowano na rysunku 11. Oczywiście warunkiem takiego korzystnego grupowania danych jest znalezienie takiej przestrzeni cech, w której dane należące do różnych kategorii będą tworzyły ładne skupienia. Ale warto do tego dążyć!



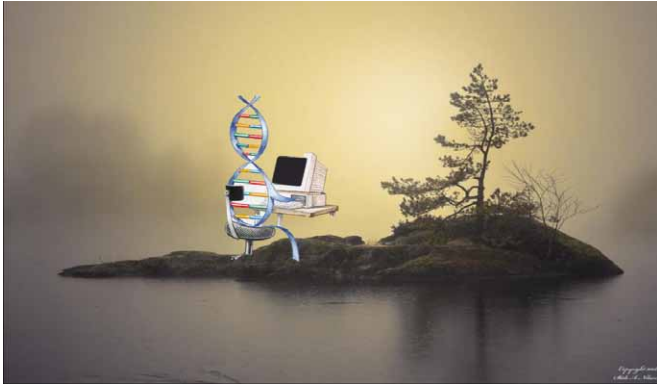
Rys. 11. Korzyść odniesiona dzięki zastosowaniu analizy skupień

Algorytmy genetyczne

Kolejna wyspa archipelagu sztucznej inteligencji omawiana w tym przeglądowym artykule związana jest z kolejną grupą metod wzorowanych na naturze. Poprzednio (grudzień 2020) omawiałem sieci neuronowe, czyli modele procesów biologicznych toczących się w skali mikro, natomiast teraz sięgamy do procesu przebiegającego w skali makro – czyli do ewolucji. Dziedzinę tę reprezentuje wyspa pokazana na rysunku 12.

Jak wiadomo, istotą procesu ewolucji jest stopniowe doskonalenie kolejnych pokoleń aż do wytworzenia gatunków optymalnych ze względu na wymagania środowiska. Ilustruje to żartobliwie rysunek 13, którego oczywiście nie trzeba brać dosłownie, bo człowiek zdecydowanie nie wywodzi się od dinozaurów – ale tak postanowiłem zaszyfrować proces ewolucyjnego doskonalenia.

Przeniesione do sztucznej inteligencji zasady ewolucji pozwalają na wygodne znajdowanie rozwiązań problemów różnorodnej **optymalizacji**. Jak wiadomo, optymalizacja polega na wyznaczeniu spośród dopuszczalnych rozwiązań jakiegoś problemu

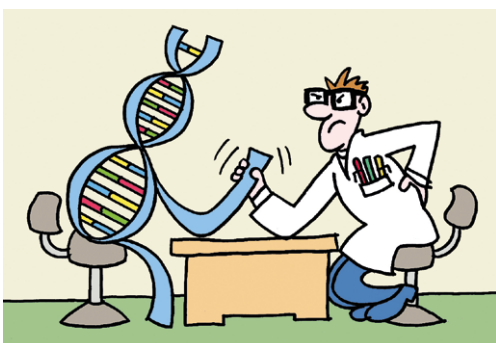


Rys. 12. Symbol genu (odcinek DNA) sterujący komputerem – to metafora algorytmów genetycznych



Rys. 13. Istotą procesu ewolucji jest stopniowe doskonalenie

rozwiązania najlepszego ze względu na przyjęte kryterium jakości (np. koszt, zysk, niezawodność). Użycie algorytmów genetycznych prowadzi do tego, że zamiast pracować poszukiwać najlepszego rozwiązania problemu informatycznego, lepiej pozwolić, żeby komputer sam sobie to rozwiązanie wyhodował [2]! W dodatku okazało się, że rozwiązania wytwarzane przez algorytm genetyczny bywają lepsze od tych wymyślonych przez ludzi (rys. 14).



Rys. 14. Rozwiązania znajdowane przez algorytmy genetyczne bywają lepsze od wymyślonych przez ludzi

Poznajmy więc podstawowe zasady, na jakich opierają się algorytmy genetyczne, żeby zrozumieć ich działanie.

Pierwszą rzeczą, jaką trzeba zrobić, gdy się planuje użyć algorytmu genetycznego do rozwiązania jakiegoś zadania, jest opisanie istoty tego zadania w postaci tak zwanego **chromosomu**.

W biologii chromosomy są nośnikami informacji genetycznej, sterującej rozwojem organizmu. W sztucznej inteligencji jest to wektor, w którym zapisane są wszystkie cechy poszukiwanego rozwiązania. Najprostszą formę chromosomu informacyjnego przedstawia rysunek 15. Jest to chromosom binarny, w którym zapisane są wyłącznie informacje o tym, że niektóre cechy występują w rozważanym rozwiązaniu (oznaczenie 1), a inne nie występują (oznaczenie 0). Chromosom będziemy utożsamiali z osobnikiem, a zbiorowość osobników z populacją.

Oczywiście chromosomy w algorytmach genetycznych mogą zawierać nie tylko zera i jedynki, ale także symbole oznaczające wybór jednej z kilku możliwości. Takie rozwiązanie pokazane jest na rysunku 16, na którym pokazano chromosom oraz osobnika (rybę) „zaprogramowanego” przez ten chromosom. Oczywiście w prawdziwych zastosowaniach algorytmów genetycznych osobnikiem nie jest ryba czy jakieś inne rzeczywiste stworzenie – tylko na przykład strategia inwestycyjna banku albo metoda sterowania zespołem zbiorników retencyjnych przy zapobieganiu powodzi – ale to trudniej narysować. Dlatego skorzystam z tych obrazków z rybkami (nie wiem, kto jest ich autorem, ale są bardzo fajne!), bo to ładnie pokazuje, o co chodzi, bez wchodzenia w skomplikowane abstrakcje.

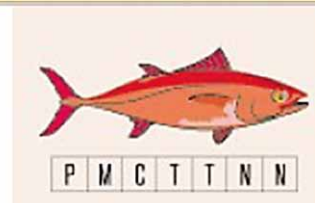


Rys. 15. Przykładowa budowa chromosomu binarnego używanego w algorytmach genetycznych

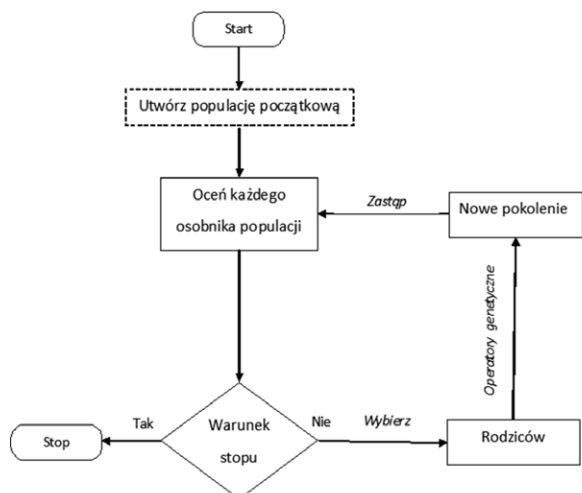
Na rysunku 16 u góry opisany jest sposób kodowania cech ryby za pomocą poszczególnych elementów chromosomu (genów), a na dole podany jest konkretny chromosom i obraz odpowiadającej mu ryby.

Ogólny schemat działania algorytmu genetycznego przedstawiono na rysunku 17. Przejdziemy przez kolejne bloki tego

kodowanie chromosomu							
kształt	ogon	color	plewy dółne	plewy górne	złoty	wąsy	
P - podkowy O - okrągły	M - mały D - duży	C - czerwony N - niebieski Z - zielony B - białe	T - są N - brak	T - są N - brak	T - są N - brak	T - są N - brak	



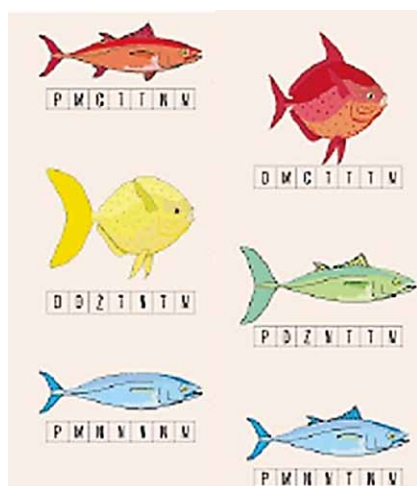
Rys. 16. Kodowanie właściwości osobnika (ryby) za pomocą chromosomu



Rys. 17. Struktura algorytmu genetycznego. Omówienie w tekście

schematu, cały czas ilustrując ogólne zasady – przykładem związanym z tymi pomysłowymi rybami. Jeszcze raz podkreślam, że nie ja ten przykład wymyśliłem, ale w żaden sposób nie mogę się dowiedzieć, kto jest autorem tego pomysłu rysunku, więc w dobrej wierze korzystam tu z cudzej własności intelektualnej – mając nadzieję, że może dzięki tej publikacji dowiem się, kogo powinienem prosić o zgodę na skorzystanie z tego rysunku.

Pierwszym krokiem algorytmu przedstawionego na rysunku 17 jest utworzenie populacji początkowej. Zwykle taką populację tworzy się, generując chromosomy zawierające geny, które mają losowo wybrane wartości. Przykład populacji początkowej dla rozważanej ewolucji ryb przedstawia rysunek 18.



Rys. 18. Przykładowa, losowo utworzona populacja początkowa ryb

Następny krok algorytmu nakazuje ocenić każdego osobnika populacji. Ta ocena w rzeczywistych zastosowaniach algorytmów genetycznych polega na tym, że każdy osobnik proponuje jakiś sposób rozwiązania postawionego zadania, a te rozwiązania dają większą albo mniejszą wartość przyjętego **kryterium jakości**. Takim kryterium może być na przykład uzyskiwany zysk, osiągnięta niezawodność, sprawność działania itp. Jeśli w skład kryterium wchodzi składniki, które uważamy

za niekorzystne (na przykład ponoszony koszt albo wielkość zanieczyszczenia środowiska) – to odpowiednie składniki można brać ze znakiem minus w formule wyznaczającej wartość kryterium albo korzystać z ich odwrotności.

Optymalizacja, będąca zawsze celem działania algorytmu genetycznego, zmierza do znalezienia takiego rozwiązania, które zapewni **największą** wartość kryterium jakości. Z tego punktu widzenia każdy osobnik może być oceniony jako lepszy (gdy zapewnia większą wartość kryterium) lub gorszy (w przeciwnym przypadku). Dla rozważanej przykładowo populacji ryb ta funkcja kryterialna może być ustalona dowolnie – na przykład w rzeczywistych warunkach mogłaby to być ilość pożywienia, jaką ryba zdołała zdobyć. W prezentowanej tu umownej zabawie w ewolucję ryb można by było przyjąć kryterium urody – ryba zdobywa tym większą liczbę punktów, im bardziej się obserwatorowi podoba.

Kolejną czynnością w rozważanym algorytmie (patrz rys. 17) jest sprawdzenie warunku stopu. Każdy algorytm musi mieć zagwarantowane, że po skończonym czasie się zatrzyma, a ewolucja może trwać w nieskończoność, więc ten warunek jest bardzo ważny. Decyzja o zatrzymaniu algorytmu może zostać podjęta po zaistnieniu jednej z dwóch okoliczności: albo osobnik z aktualnie rozważanej populacji, który osiągnął najlepszy wynik, jest już tak dobry, że więcej nie trzeba szukać, albo została wykonana założona z góry maksymalna liczba iteracji (wymian pokoleń – patrz dalej).

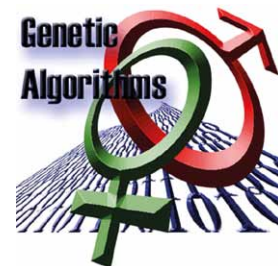
Zaistnienie pierwszej okoliczności stopującej algorytm oznacza sukces. Chromosom „zwycięskiego” osobnika zawiera przepis, jakie mają być cechy optymalnego rozwiązania. Niczego więcej nie potrzeba! Jednak ta sytuacja prawie nigdy nie zachodzi w rozważanej w tym momencie populacji początkowej.

Zatrzymanie algorytmu na skutek wyczerpania założonej liczby operacji oznacza niepowodzenie. Symulowana ewolucja nie dostarczyła akceptowalnego rozwiązania. Trzeba poszukiwać innych metod!

Jeśli warunek stopu nie nakazuje przerwania pracy – przechodzimy w prawo i przystępujemy do wyboru rodziców, którzy spłodzą osobniki, będące ich potomkami w następnym pokoleniu. Trzeba bowiem wiedzieć, że reprodukcja osobników w algorytmach genetycznych ma charakter płciowy, co symbolicznie przypomina rysunek 19.

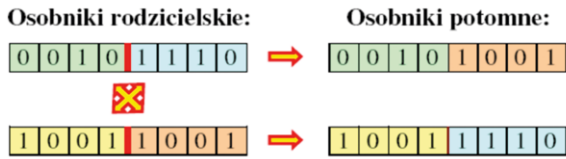
Jaki jest powód takiego wyboru?

Otóż w trakcie formowania chromosomów osobników potomnych chromosomy osobników rodzicielskich podlegają krzyżowaniu. Wybierany jest losowo punkt, w którym chromosomy osobników rodzicielskich zostają przecięte i zestawione w taki sposób, że każdy osobnik potomny otrzymał połowę chromosomu jednego rodzica i połowę chromosomu drugiego (rys. 20).

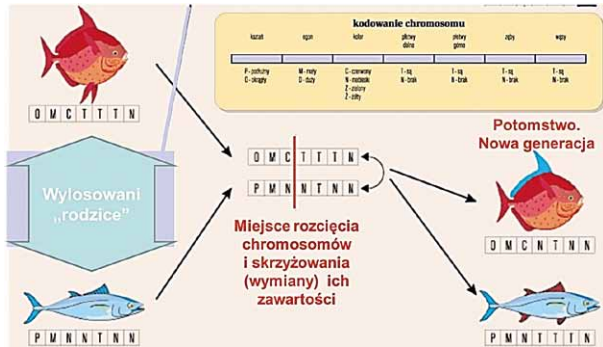


Rys. 19. Symbol przypominający płciowy charakter generacji nowych pokoleń w algorytmach genetycznych

W żartobliwym przykładzie ewolucji ryb wygląda to tak, jak przedstawiono na rysunku 21.



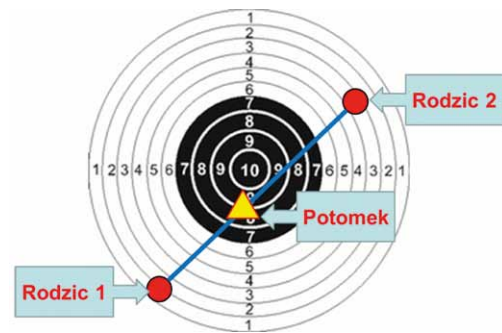
Rys. 20. Zasada krzyżowania



Rys. 21. Proces krzyżowania chromosomów

Po co to się robi?

Chodzi o to, by potomek – dziedzicząc cechy po obojgu rodzicach – mógł mieć większe szanse na to, by w następnym pokoleniu okazać się „zwyćskim”. Ilustruje to rysunek 22, na



Rys. 22. Korzystny efekt krzyżowania genów dwojga rodziców

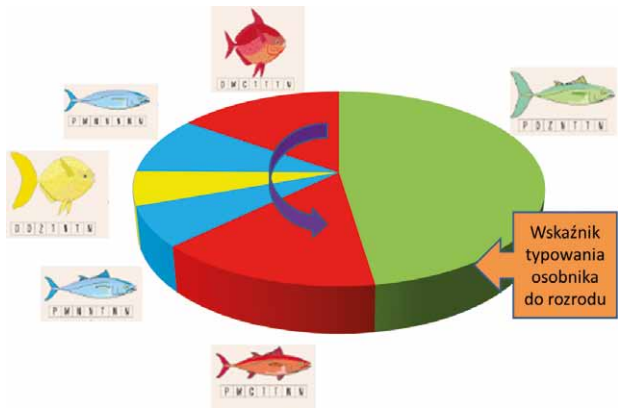
którym proces poszukiwania najlepszego osobnika zilustrowano jako strzelanie do tarczy. Obydwa osobniki rodzicielskie uzyskują nie najlepsze wyniki (daleko od centrum tarczy), natomiast potomek, mający chromosom powstały z krzyżowania chromosomów rodziców, może uzyskać lepszy wynik.

No dobrze, ale jak dobierać rodziców?

Logiczne jest, by promować chromosomy tych osobników, którzy osiągnęli w populacji lepsze wyniki. Dlatego w algorytmach genetycznych stosuje się zwykle zasadę ruletki. Każdemu osobnikowi z aktualnie rozważanej populacji przydziela się sektor na obwodzie symulowanego komputerowo „koła ruletki” o wielkości proporcjonalnej do uzyskanej przez niego wartości

reklama

reklama



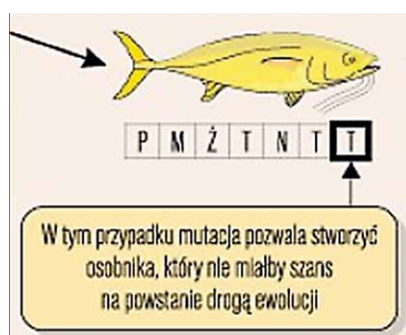
Rys. 23. Podział „koła ruletki” na sektory odpowiadające poszczególnym osobnikom

funkcji kryterialnej (rys. 23). Koło zostaje wprowadzone w ruch (a właściwie inicjowana jest akcja realizowanego algorytmicznie generatora liczb losowych), a gdy się zatrzyma – przy wskaźniku typowania osobnika do rozrodu jest punkt sektora należącego do określonego osobnika, więc to on zostaje wytypowany.

Potem w taki sam sposób typowany jest drugi osobnik rodzicielski – i tworzona jest para, która w sposób wyżej opisany wygeneruje potomstwo. Najczęściej do następnej populacji brane są dwa osobniki potomne o chromosomach wytworzonych metodą krzyżowania. Potem „koło ruletki” rusza jeszcze raz i jeszcze raz – aż wytworzona zostanie populacja potomków o takiej samej liczebności, jak populacja rodzicielska. Osobniki odnoszące sukcesy mogą mieć potomstwo wielokrotnie, a osobniki mające kiepskie wyniki mogą wcale nie doczekać się swojej okazji – ale o wszystkim decyduje mechanizm losowy. Nawet jeśli w wyniku losowania ta sama para osobników zostanie ponownie wylosowana do reprodukcji – potomstwo może być całkiem inne, bo miejsce rozcięcia chromosomów jest także wybierane losowo, więc w wyniku krzyżowania powstaną (zapewne – bo tu także działają czynniki losowe) osobniki potomne o odmiennych właściwościach.

Jak wynika z rysunku 17 – przy tworzeniu nowego pokolenia z pokolenia rodziców działają operatory genetyczne. Ważniejszy z tych operatorów już poznaliśmy – jest to operator krzyżowania. Dodatkowo, ale z mniejszym prawdopodobieństwem, działa operator mutacji.

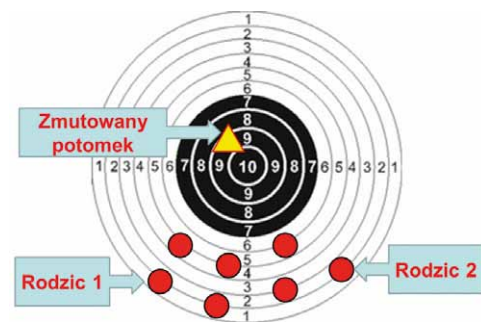
Mutacja polega na tym, że w chromosomie któregoś z osobników potomnych zmienia się losowo jeden z genów. Przykład mutacji przedstawiono na rysunku 24. W pokoleniu rodziców



Rys. 24. Ilustracja działania operatora mutacji

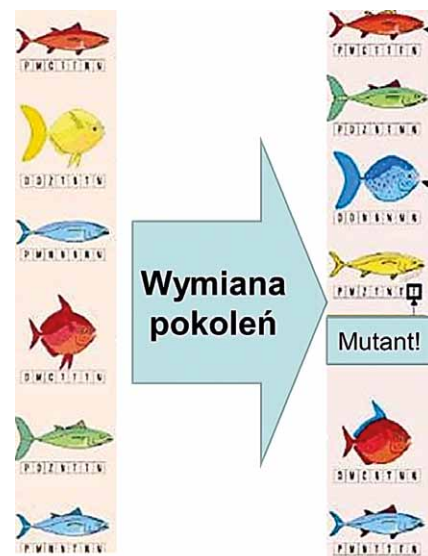
żadna ryba nie posiada wąsów. Natomiast jeden z potomków uzyskuje ten element swojej budowy na zasadzie losowej zmiany odpowiedniego genu. Oglądając rysunek 24, warto zwrócić też na rysunek 16, żeby sobie przypomnieć znaczenie poszczególnych genów.

Istota działania operatora mutacji polega na tym, że w wyniku jego działania pojawiają się w populacji potomnej osobniki mające cechy, których w populacji rodzicielskiej nie ma, na skutek czego żadne krzyżowanie w tej populacji nie pozwala na osiągnięcie znaczącego polepszenia osiąganych wartości kryterium. Natomiast mutant może mieć te możliwości. Skutek tego w dążeniu do osiągnięcia zamierzonego celu przedstawiony jest na rysunku 25.



Rys. 25. Efekt działania mutacji

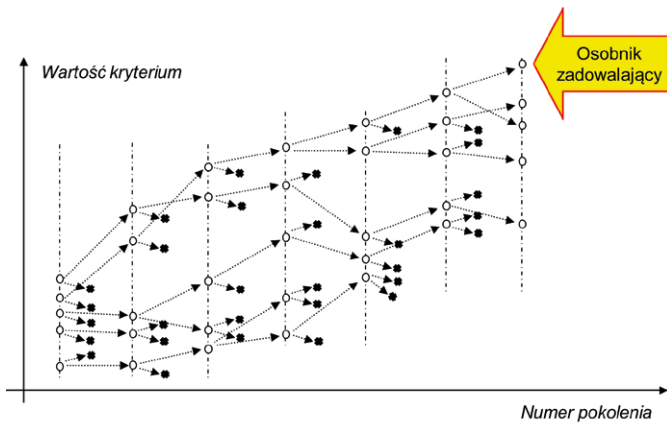
Po wykonaniu wszystkich operacji wyboru rodziców, krzyżowania genów i ewentualnie mutacji powstaje – zgodnie z rysunkiem 17 – nowe pokolenie. Przedstawiono to na rysunku 26.



Rys. 26. Pokolenie rodzicielskie i potomne w procesie przykładowej ewolucji

Po podmianie pokolenia rodzicielskiego na pokolenie potomne cały cykl się zamyka. Teraz to osobniki pokolenia potomnego będą oceniane, na podstawie ich ocen będzie się rekrutować rodziców, którzy będą generować nowych osobników z zastosowaniem operatorów genetycznych – aż powstanie kolejne pokolenie i ponownie cykl się zamknie.

Na rysunku 27 pokazano, jak wartość kryterium zmienia się w kolejnych pokoleniach. W każdym pokoleniu są osobniki



Rys. 27. Przebieg optymalizacji przy użyciu algorytmu genetycznego

lepsze i gorsze, ale w reprodukcji następnego pokolenia biorą udział głównie te lepsze, więc cała populacja regularnie podnosi swoje wyniki – aż wreszcie nadchodzi moment, kiedy najlepszy osobnik któregoś pokolenia osiąga taką wartość funkcji kryterium, która może być uznana za zadowalającą. W tym momencie działanie algorytmu można przerwać i wykorzystać uzyskane rozwiązanie. Warto zauważyć, że uzyskanie tego rozwiązania przy użyciu algorytmu genetycznego wiąże się z ogromnym nakładem obliczeń, czyli wymaga zaangażowania silnego komputera albo dłuższego oczekiwania na wyniki. Ale za to rozwiązanie uzyskuje się bez własnego wysiłku – po zdefiniowaniu chromosomu i funkcji obliczającej wartości kryterium wszystko dzieje się automatycznie. Nic dziwnego, że algorytmy genetyczne stale zyskują na popularności!

Algorytmy mrówkowe – podejście wąskie, ale owocne

Kolejną grupą metod sztucznej inteligencji – wzorujących swoje działanie na naśladowaniu w komputerach procesów zaobserwowanych w naturze – są algorytmy mrówkowe, należące do szerszej klasy metod określanych jako „inteligencja roju”. Metaforyczna wyspa (rys. 28) prezentująca tę grupę metod sztucznej inteligencji ma charakterystyczny kształt: jest raczej wąska, ale wysoka. Sygnalizuje to fakt, że obszar zastosowań tej grupy jest wąski, ale rozwiązywane problemy bywają naprawdę trudne. Wiążą się one głównie z wyszukiwaniem najkorzystniejszej ścieżki w grafie, co jest matematyczną i informatyczną



Rys. 28. Prezentacja alegorii algorytmów mrówkowych

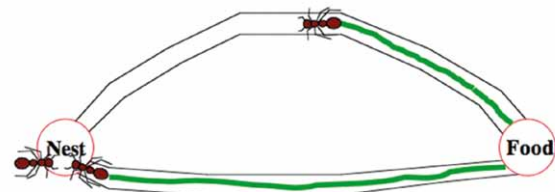
reprezentacją bardzo wielu praktycznych problemów. Kilka przykładów zastosowania algorytmów mrówkowych do rozwiązywania bardzo różnych zagadnień znaleźć można w publikacjach [3, 4, 5, 6, 7]. Algorytmy te bywają bardzo skuteczne, co symbolizuje potężne drzewo rosnące na wyspie, a rozpoczęcie pracy z nimi jest łatwe, co jest zasygnalizowane przez brzegi nadające się do łatwego wkroczenia. Przypominam, że we wcześniejszych artykułach przedstawiałem niektóre wyspy z bardzo nieprzyjawnymi brzegami (na przykład metody symboliczne), znamionującymi trudności dla początkujących użytkowników. W przypadku algorytmów mrówkowych takich trudności się nie napotyka.

Idea algorytmów mrówkowych opiera się na spostrzeżeniu, że pojedyncza mrówka nie cechuje się wysoką inteligencją, ale zbiorowość mrówek jest w stanie wytworzyć inteligencję kolektywną. Przejawem tej kolektywnej inteligencji jest fakt, że mrówki, dążąc do jakiegoś celu, nie poruszają się chaotycznie, ale tworzą ścieżki (rys. 29).



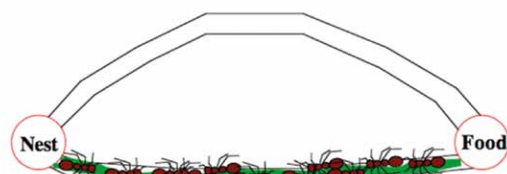
Rys. 29. Przykłady kolektywnej inteligencji mrówek

Opiera się to na bardzo prostej zasadzie: każda mrówka, idąc, pozostawia za sobą ślad w postaci substancji chemicznej (feromonu), która jest wyczuwana przez inne mrówki, które mają skłonność do podążania właśnie drogą zaznaczoną feromonem. Jednak feromon szybko paruje i ślad znika. Jeśli więc do jakiegoś celu są dwie drogi: krótsza i dłuższa – to pierwszy ślad feromonowy zostawi mrówka idąca krótszą drogą (rys. 30).



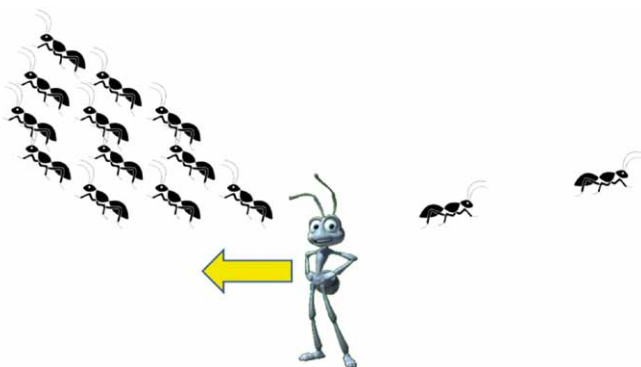
Rys. 30. Wczesniejszy ślad feromonowy na krótszej drodze

W efekcie tą krótszą drogą wybierze więcej mrówek (rys. 31).



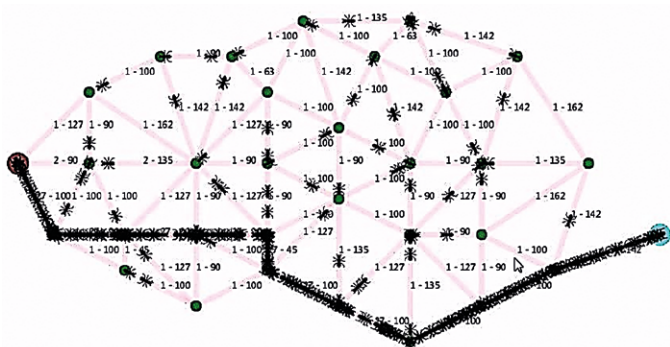
Rys. 31. Więcej mrówek podąża krótszą drogą

Każda kolejna mrówka, stając przed wyborem drogi – wybierze tę częściej uczęszczaną ze względu na silny ślad feromonowy (rys. 32).



Rys. 32. Zasada wyboru drogi przez pojedynczą mrówkę

Stosując tę prostą zasadę, można wygodnie (całkowicie automatycznie!) znajdować drogę w dowolnie rozbudowanym grafie (rys. 33). Odpowiednio pomysłowo stosując to podejście, można rozwiązywać mnóstwo praktycznych problemów, o czym świadczą przykłady opisane w publikacjach [3, 4, 5, 6, 7].



Rys. 33. Efekt działania algorytmu mrówkowego w bardziej rozbudowanym grafie

Uwagi końcowe

Chociaż prezentowany tu artykuł jest już trzecim w tej serii i jest dość obszerny – to jednak temat różnych metod sztucznej inteligencji (przedstawianych jako wyspy archipelagu) wciąż nie został całkiem wyczerpany. Zapowiadam zatem kontynuację cyklu w następnych numerach miesięcznika „Napędy i Sterowanie” – i mam nadzieję, że znajdą się Czytelnicy chętni do tego, żeby także i ten następny artykuł przeczytać.

Literatura

[1] TADEUSIEWICZ R., IZWORSKI A.: *Segmentation of Speech on Purpose of Recognition Using Cluster Analysis*. „Elektrotechnika” tom 7, zeszyt 3–4, 1988, https://www.academia.edu/27384037/Segmentation_of_Speech_on_Purpose_of_Recognition_Using_Cluster_Analysis.

- [2] Wyhoduj sobie rozwiązanie! (Algorytmy genetyczne, czyli jak stworzyć w komputerze sztuczną ewolucję) [*Breed your unique solution!*] Polska Gazeta Krakowska, 17.08.2011, https://www.academia.edu/39966084/Wyhoduj_sobie_rozwiazanie_; <http://www.gazetakrakowska.pl/artukul/439777,algorytmy-genetyczne-czyli-jak-stworzyc-w-komputerze-sztuczna-ewolucje,id,t.html>
- [3] PANCERZ K., LEWICKI A., TADEUSIEWICZ R., SZKOL J.: *Classification of Speech Signals through Ant Based Clustering of Time Series*, [IN] NGUYEN N.-T. ET AL. (EDS.): *Computational Collective Intelligence. Technologies and Applications, Part I*, LNAI 7653, Springer-Verlag, Berlin, Heidelberg, 2012, https://www.researchgate.net/publication/262388568_Classification_of_Speech_Signals_through_Ant_Based_Clustering_of_Time_Series.
- [4] LEWICKI A., PANCERZ K., TADEUSIEWICZ R.: *Ant Colony Inspired Clustering Based on the Distribution Function of the Similarity of Attributes*. Chapter in the book: NGUYEN N.T., TRAWINSKI B., KATARZYNIAK R., GEUN-SIK JO (EDS.): *Advanced Methods for Computational Collective Intelligence*, Studies in Computational Intelligence, vol. 457, Springer-Verlag Berlin Heidelberg 2013, https://www.researchgate.net/publication/278195962_Ant_Colony_Inspired_Clustering_Based_on_the_Distribution_Function_of_the_Similarity_of_Attributes.
- [5] PANCERZ K., LEWICKI A., TADEUSIEWICZ R., WARCHOL J.: *Rough Set Flow Graphs and Ant Based Clustering in Classification of Disturbed Periodic Biosignals*, Chapter in the book: POPOVA-ZEUGMANN L. (ED.): *Concurrency, Specification and Programming*. Informatik Berichte, Humboldt-Universität zu Berlin, 2012, https://www.researchgate.net/publication/289731888_Rough_set_flow_graphs_and_ant_based_clustering_in_classification_of_disturbed_periodic_biosignals.
- [6] PANCERZ K., LEWICKI A., TADEUSIEWICZ R., GOMULA J.: *Ant Based Clustering of MMPI Data – An Experimental Study*. Chapter in the book: YAO J.T, RAMANNA S., WANG G., SURAJ Z. (EDS.): *Rough Set and Knowledge Technology*, LNAI 6954, Springer Verlag, Berlin, Heidelberg, New York, 2011, https://www.researchgate.net/publication/226283168_Ant_Based_Clustering_of_MMPI_Data_-_An_Experimental_Study; https://www.researchgate.net/publication/221104687_Ant_Based_Clustering_of_MMPI_Data_-_An_Experimental_Study.
- [7] LEWICKI A. AND TADEUSIEWICZ R.: *The Recruitment and Selection of Staff Problem with an Ant Colony System* [IN:] PARDELA T. AND WILAMOWSKI B. M. (EDS.): 3rd International Conference on Human System Interaction, Rzeszow, IEEE-Press, 2010, https://www.researchgate.net/publication/251938024_The_recruitment_and_selection_of_staff_problem_with_an_Ant_Colony_system.

✉ prof. dr hab. inż. Ryszard Tadeusiewicz
AGH – Akademia Górniczo-Hutnicza w Krakowie
e-mail: rtad@agh.edu.pl, www.Tadeusiewicz.pl