

KOMPUTEROWE SYMULACJE EWOLUCJI

Michał Gocyla (Kraków)

Wstęp

Aby zaistniało zjawisko ewolucji biologicznej konieczne jest spełnienie wielu warunków. Po pierwsze musi istnieć środowisko, w którym żyć będzie populacja krzyżujących się osobników. Każdy z tych osobników powinien posiadać unikalny genotyp stanowiący podstawę dla fenotypu czyli zespołu cech jakimi charakteryzuje się dany osobnik. Taki opis jest właściwy dla świata organizmów żywych. Co otrzymalibyśmy próbując zdefiniować wymienione warunki w języku współczesnej techniki? Otóż środowisko stanowi źródło problemów, z którymi muszą sobie poradzić osobniki. Jego zadaniem jest również określenie czy i w jakim stopniu dany osobnik spełnił stawiane mu wymagania. Na podstawie tej właśnie oceny podejmowana jest decyzja o dalszym losie sprawdzanego. Może on istnieć dalej po to, by wziąć udział w krzyżowaniu lub zostać wyeliminowany. Ewentualnie może również przeżyć, pomimo iż jego cechy uniemożliwiły mu przekazanie swoich genów. Wszystko to sprawia, że środowisko można opisać jako układ składający się z dwóch składników. Pierwszym jest zbiór problemów. Drugi to funkcja oceniająca rozwiązania zaproponowane przez danego osobnika, czyli jego przystosowanie. Tak więc, jeśli przyjmiemy, że fenotyp (zbiór cech) osobnika stanowi rozwiązanie, to cała populacja nie będzie niczym innym, jak zbiorem mniej lub bardziej podobnych rozwiązań. Co się zaś tyczy pojęcia genotypu, to w biologii stanowi on opis, na podstawie którego tworzony jest nowy osobnik. W technice również stosowane są takie opisy – zazwyczaj nazywa się je projektami.

Algorytm Genetyczny – Teoria

Powyższy opis można wykorzystać do stworzenia modelu, którego zastosowanie w określonych warunkach będzie dawało wyniki bardzo podobne do tych generowanych przez proces ewolucji biologicznej. Do modelowania zjawiska ewolucji można wykorzystać algorytmy stworzone na podstawie obserwacji efektów jej działania. Takimi właśnie algorytmami są algorytmy genetyczne. Zasada ich działania jest podobna do praw rządzących ewolucją. Na początku mamy więc pewną populację losowo wybranych osobników. W przyrodzie populacja taka mogłaby powstać poprzez przeniesienie pewnej liczby

osobników danego gatunku do nowego środowiska. Nowe środowisko stawia przed osobnikami nowe wyzwania. Po pewnym czasie okazuje się, że tylko część osobników może dożyć wieku rozrodczego i wydać potomstwo. Tak potomstwo, jak również osobniki dorosłe są przez cały czas narażone na mutacje. W przypadku algorytmów owe potomstwo stanowią nowe unikalne projekty powstałe poprzez mieszanie fragmentów projektów z poprzedniego pokolenia. Przy czym warto tutaj zaznaczyć, iż istnieje kilka możliwości. W pierwszej źródłowa populacja może składać się z rozwiązań już znanych. Wtedy liczba odrzuconych projektów będzie stosunkowo niewielka. Druga możliwość to generacja rozwiązań całkowicie przypadkowych, której efektem jest zazwyczaj duży zbiór rozwiązań niespełniających stawianych wymagań. Rozwiązania satysfakcjonujące są wybierane dzięki działaniu funkcji oceniającej. Ważne jest, aby osobniki były poddawane ocenie po uwzględnieniu efektów mutacji. Następnie dochodzi do krzyżowania się wybranych osobników, czyli mieszania fragmentów rozwiązań. Efektem krzyżowania jest kolejna populacja lepszych i gorszych rozwiązań. Te lepsze uczestniczą w następnym krzyżowaniu, te gorsze są odrzucane. Mutacje, wybór i krzyżowanie powtarza się zazwyczaj wiele razy. Proces zatrzymywany jest w dwóch przypadkach. Pierwszy to sukces, czyli otrzymanie poszukiwanego rozwiązania. Porażka następuje gdy kolejne próby dają podobne, niesatysfakcjonujące rozwiązania.

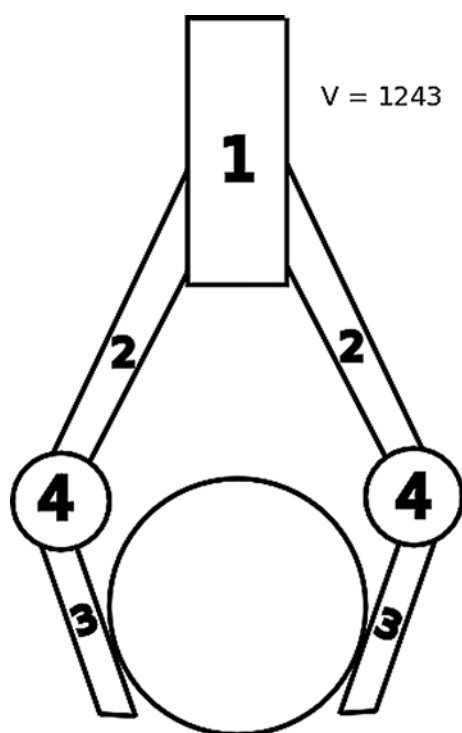
Algorytm Genetyczny – Realizacja

Tak przedstawia się teoria. W praktyce jednak ujawnia się kilka zasadniczych problemów. Po pierwsze trzeba wybrać metodę kodowania posiadanych informacji. Metoda ta musi umożliwiać zapisanie danych w postaci chromosomów stanowiących projekt każdego rozwiązania. Stworzone musi zostać środowisko i funkcja oceniająca. Niezbędne jest również zdefiniowanie operacji wykonywanych na chromosomach w czasie mutacji i krzyżowania chromosomów.

W informatyce operacje wykonuje się zazwyczaj na ciągach znaków. W algorytmach genetycznych wykorzystywać można ciągi różnych znaków. Mogą to być: zera i jedyńki (kodowanie binarne), cyfry, litery oraz ciągi mieszane zawierające wszystkie możliwe znaki. Przy czym różnorodność znaków

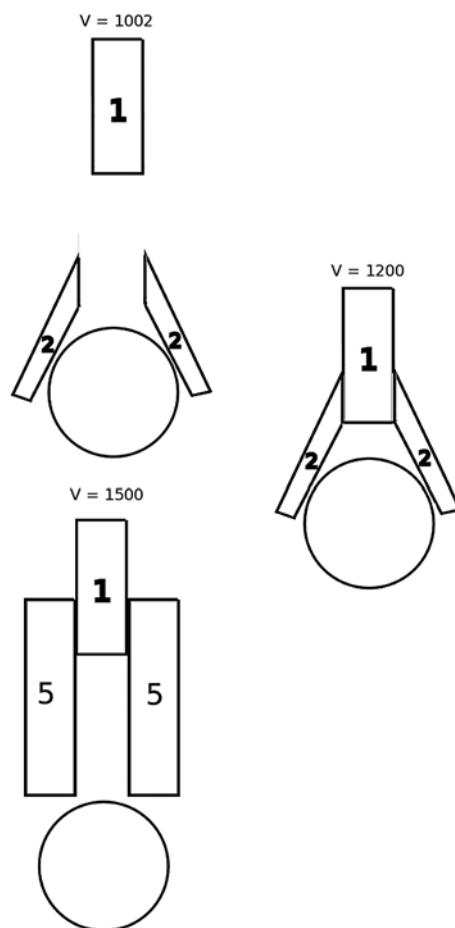
podlega zazwyczaj technicznemu ograniczeniu w postaci znaków nieobsługiwanych przez system lub język w którym powstaje program zawierający implementację algorytmu. Warto również zwrócić uwagę na fakt, iż komplikacja zapisu zwiększa czas potrzebny na wykonanie operacji, co zmniejsza wydajność procesu. Wreszcie wybór niewłaściwej metody kodowania może przyczynić się do utraty kluczowych informacji, co uniemożliwi otrzymanie poszukiwanego rozwiązania. Wybrawszy najwłaściwszą z metod, należy zastanowić się kształtem i ilością chromosomów. Często stosowaną zasadą jest przedstawianie każdego z projektów pod postacią jednego tylko chromosomu. Chromosom taki może mieć postać wektora. W tej postaci każdy gen jest reprezentowany przez znak, a znaki zapisywane są jeden po drugim.

Prawdziwe problemy zaczynają się jednak przy tworzeniu funkcji symulującej środowisko. I tak, jeśli celem algorytmu jest zaprojektowanie układu, który będzie wykonywał jakąś pracę, to miarą przystosowania będzie oczywiście skuteczność wykonywanej pracy. Dla przykładu rozważmy prosty przypadek chwytaka, którego zadaniem jest podniesienie kuli wykonanej z miękkiego materiału (ryc. 1). Aby powstał taki układ, konieczne jest matematyczne opisanie środowiska w którym istnieje ciężenie oraz określenie siły z jaką chwytak będzie naciskał na kulę



Ryc. 1. Chwytak, którego zadaniem jest podniesienie kuli. Poszczególne elementy oznaczone cyframi. Oznaczenia elementów zapisano w postaci wektora o stałej liczbie czterech współrzędnych. 0 oznacza brak elementu na danej pozycji.

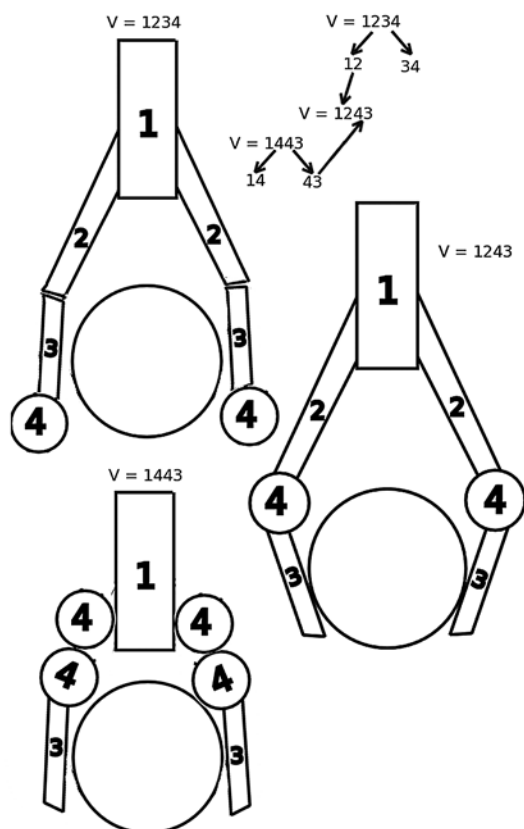
(siła ta ma wartość stałą). Wybrano również metodę kodowania. Chromosomy są tutaj wektorami składającymi się z cyfr. Każda cyfra jest numerem pewnego rodzaju części. Części są dodawane w kolejności wynikającej z pozycji genu (cyfry) w chromosomie (wektorze). Aby wykonanie zadania było w ogóle możliwe, konieczne było dodanie ruchomych połączeń pomiędzy kolejnymi prostopadłościanami (ruch następował zawsze tylko w jedną stronę). Oczywiście taki typ połączeń również mógłby powstać na drodze ewolucji, ale wtedy nasz przykład stałby się raczej skomplikowany. Niektóre rodzaje części układu mają również możliwość zbierania informacji. Jeśli zbierane będą tylko dane na temat powierzchni styku kuli i prostopadłościanu, to powierzchnia ta będzie stanowiła kryterium wyboru najlepszego z powstałych rozwiązań. Dodatkowym kryterium będzie obecność połączeń pomiędzy poszczególnymi elementami. Nowe projekty powstają na drodze mutacji i krzyżowania (ryc. 2, 3). Po kilku tysiącach „pokoleń” powstanie chwytak podnoszący kulę (ryc. 1). Jednakże jakość funkcji jest silnie zależna od danych na temat warunków



Ryc. 2. Kolejne pokolenia. Oba „osobniki” są efektem mutacji dodającej losowy element w losowym miejscu (bez zastępowania elementu 1). Po prawej wybrany, po lewej odrzucony. Powodami odrzucenia były: brak połączeń pomiędzy elementami (górny), zbyt mała powierzchnia kontaktu z podnoszoną kulą (dolny).

pracy układu takich jak na przykład waga podnoszonej kuli lub wytrzymałość połączeń. Brak uwzględnienia takich informacji w czasie projektowania zaowocowałaby powstaniem chwytaka niskiej jakości (ryc. 4).

Niełatwym zadaniem jest również wybór najlepiej przystosowanych osobników. Można tutaj zastosować

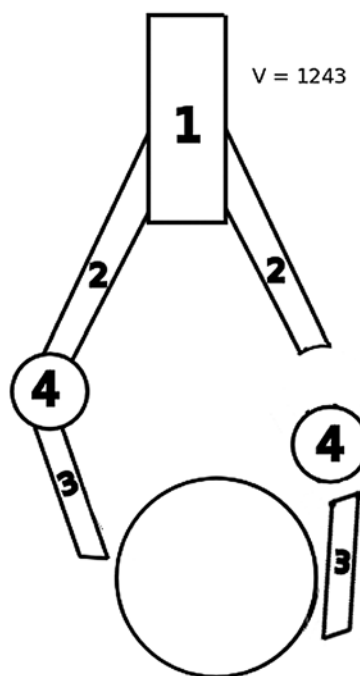


Ryc. 3. Krzyżowanie. Podział ciągu znaków na dwie równe części. Wymiana części pomiędzy podzielonymi ciągami. Krzyżowanie następuje tylko pomiędzy osobnikami spełniającymi określone kryteria.

wać metodę ruletki. W której miarą przystosowania jest ilość pól przynależnych każdemu z osobników (ryc. 5). Przydatność tej metody spada wraz ze wzrostem podobieństwa osobników – osobniki podobne otrzymują takie same wycinki koła. Można też zastosować metodę rankingową – każdy osobnik dostaje miejsce na liście. Problemem jest tutaj brak wrażliwości na różnice pomiędzy osobnikami – wszystkie osobniki spełniające określone kryteria są traktowane tak samo. Poprzez niewłaściwe stosowanie powyższych metod możemy otrzymać populację bardzo podobnych osobników, których wyniki będą oscylowały wokół pewnego ekstremum, które będzie jedynie lokalnym ekstremum badanej funkcji (ryc. 6)

Aby odzwierciedlić proces ewolucji biologicznej konieczne są również mutacje. Mutacje występują z określonym prawdopodobieństwem. Prawdopodobieństwo to jest zazwyczaj niskie, ponieważ duża ilość mutacji spowodowałaby nagły wzrost zmienności,

którego efektem byłoby zniszczenie wielu potencjalnie dobrych rozwiązań. Operacje wykonywane w czasie wprowadzania efektów mutacji są zależne od wybranej metody kodowania. Mogą to być odpowiedniki mutacji spotykanych w przyrodzie czyli: wstawianie pewnych fragmentów kodu w różne



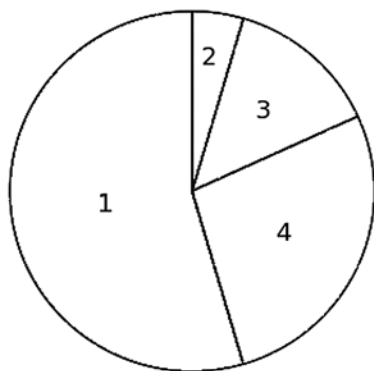
Ryc. 4. W czasie projektowania założono, iż wytrzymałość połączeń jest wystarczająca, a jej zwiększanie nie ma znaczenia – nie stanowi kryterium. Było to błędne założenie...

miejsca chromosomu, kasowanie jednego lub kilku genów, wymiana pewnej ilości genów. Można jednak również stosować inne operacje. Przykładowo zastosowanie kodowania binarnego umożliwia wykonywanie operacji logicznych, chromosom składający się z ciągu cyfr pozwoli na operacje algebraiczne. Dokładnie tak samo wygląda kwestia krzyżowania. Spotykane w przyrodzie rozcinanie i łączenie można w zależności od potrzeb zastąpić dowolną inną operacją. Jedynymi ograniczeniami są tutaj po raz kolejny wykorzystywane oprogramowanie – nie każdy język umożliwia proste dodawanie ciągów liter.

Przykłady Zastosowań Komputerowych Symulacji Procesu Ewolucji

W ten oto sposób można wykorzystać komputer do symulowania procesu ewolucji biologicznej. Wszystko to nie odpowiada jednak na zasadnicze pytanie. Jakie korzyści mogą wynikać z symulowania zjawiska ewolucji biologicznej. Otóż oprócz niezaprzeczalnej wartości poznawczej symulacje takie pozwalają osiągnąć znaczne korzyści materialne.

Jednym z przykładów może być rozwiązanie problemu komiwojagera. Polega on na odnalezieniu najkrótszej drogi łączącej wszystkie miasta tak, by przez



Ryc. 5. W tej próbie najlepsze rozwiązanie to 1. Najgorsze to 2.

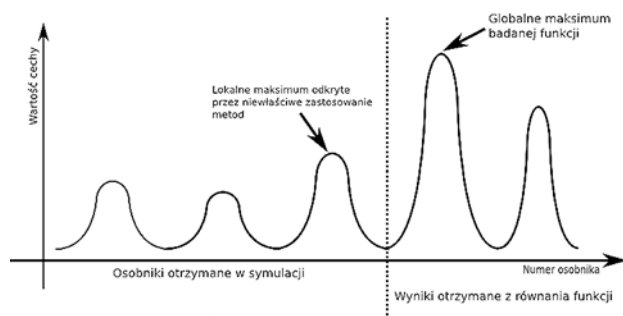
każde miasto przejść tylko raz. Oczywiście algorytm genetyczny nie zawsze da optymalne rozwiązanie. Jednak otrzymane rozwiązania sub-optymalne mogą okazać się wystarczające. Wyszukiwać można nie tylko drogę, ale również i dowolne informacje, co jest możliwe, ponieważ zasada działania algorytmów genetycznych umożliwia przeszukiwanie dużych zbiorów rozwiązań. Główną różnicą jest tutaj postać chromosomów. Przy zapisie drogi lepiej sprawdzają się wektory. Przy wyszukiwaniu i sortowaniu informacji używa się natomiast struktur drzewiastych.

Wielkie nadzieje daje również projektowanie genetyczne. Przy czym tutaj widoczny staje się problem różnic pomiędzy symulacją, a symulowanym środo-

wie w elektronice, gdzie algorytmy genetyczne łączy się z układami *Field Programmable Gate Array* (Bezpośrednio Programowana Macierz Bramek). Cechą tych układów jest zdolność do szybkiej zmiany struktury tak, iż do pomiarów używa się rzeczywistego układu. Jeszcze dalej poszli twórcy projektu GOLEM, którego celem jest automatyczne projektowanie i tworzenie robotów spełniających określone zadania. Jest to możliwe dzięki połączeniu metod projektowania genetycznego oraz druku trójwymiarowego. Roboty zaprojektowane w wirtualnym środowisku są automatycznie drukowane i sprawdzane w rzeczywistości. Dane o stopniu przystosowania wracają do komputera po to, by wybrać najlepsze z projektów i dać początek nowemu pokoleniu.

Podsumowanie

Możliwe jest stworzenie modelu opisującego zjawisko ewolucji biologicznej. Modele takie można wykorzystać do stworzenia symulacji tego procesu. Wielkie korzyści niosą również praktyczne zastosowania zdobytych informacji. Obecnie jedynym ograniczeniem wydaje się być moc obliczeniowa wykorzystywanych komputerów. Perspektywy rozwoju można natomiast podsumować jednym tylko pytaniem. Jak będzie wyglądał świat, w którym maszyna zaprojektuje się sama, a osoba projektanta stanie się zbędna?



Ryc. 6. Sprawdzenie metody selekcji. Zastosowanie algorytmu ewolucyjnego do odnalezienia ekstremum funkcji, której wzór był znany. Numery osobników przyporządkowano tak, aby zilustrować zjawisko – jedno maksimum dla każdej populacji. Przedostatnie maksimum (maksimum globalne) otrzymano podstawiając wyliczone wartości. Ostatnie maksimum to efekt podstawienia innych wartości.

wiskiem. Optymalne byłoby rozwiązanie, w którym tworzony jest prawdziwy obiekt podlegający działaniu środowiska naturalnego. Obecnie takie rozwiązanie jest już możliwe. Stosuje się je na przykład