

Czy roboty przyszłości będą ewoluować?

O robotach – czyli robotnych automatach, których pracą sterują programy komputerowe – pisaliśmy już sporo, zwłaszcza w eseju piątym. Wiedząc o tym, że są to istoty skomputeryzowane, ktoś mógłby stwierdzić: „*Po cóż pytać o ewolucję maszyn przyszłości, skoro już teraz, na naszych oczach, dokonuje się nieustanna ewolucja komputerowych technologii?*”.

Stwierdzenie to mógłby uzasadnić, podając te same przykłady, które my zebraliśmy w eseju szóstym. Przypomnijmy. W dziedzinie sprzętu, zwanego z angielska *hardware*, informatyczny postęp znaczą kolejne przełomy technologiczne: wynalezienie lamp elektronowych, tranzystorów, następnie zaś układów scalonych o wysokiej i bardzo wysokiej skali integracji (przy czym są to tylko wynalazki kluczowe, którym towarzyszy moc pomniejszych innowacji). Na polu drugim, znanym z kolei jako *software*, inwencja ludzka wykazuje aktywność nie mniejszą. Wytwarza bez ustanku nowe programistyczne idee, pośród których palmę pierwszeństwa dzierżą: kodowanie binarne, języki programowania (niskiego i wysokiego poziomu), idea interfejsu, sprzężenie zwrotne, przetwarzanie sieciowe. Jednym słowem: ewolucja informatyczna trwa, a jej wynikiem są między innymi coraz bardziej wyrafinowane roboty.

Choć to wszystko prawda, w niniejszym tekście nie o takiej ewolucji będziemy pisać. Będzie nam chodzić o proces autonomiczny, pozbawiony ludzkiej kontroli, przypominający to, co od czasów Karola Darwina zwykliśmy nazywać naturalną *ewolucją gatunków*. To właśnie tego rodzaju zjawisko, rozpatrywane jednak w dziedzinie układów sztucznych, będzie nas interesować dalej.

§1. Ewolucja naturalna

Skoro wyznaczyliśmy cel i powiązaliśmy go z osobą Karola Darwina, musimy zarysować idee, które uczony ten wprowadził do współczesnej nauki. Są to idee niezwykle wpływowe. Za ich sprawą niemal powszechne stało się przekonanie, że gatunki zwierzęce i roślinne ewoluują, a większość cech właściwych ludziom (łącznie ze świadomością i zdolnościami komunikacyjnymi) powstała w wyniku tego właśnie procesu. Skoro tak, to musimy wyjaśnić wstępnie, jak ów proces przebiega i jakie elementy obejmuje? Spróbujmy dać opis maksymalnie skrótowy, naginając jego kształt do dalszych uwag, powiązanych już z informatyką.

Zgodnie z wybraną na tytuł tego podrozdziału nazwą „evolucja naturalna”, ewolucji podlega pewien gatunek żyjący w swoim *naturalnym środowisku*. Przedstawiciele tego gatunku rozmnażają się, a czyniąc to, płodzą więcej potomstwa niż ma szansę przeżyć. Potomstwo to rywalizuje ze sobą o przetrwanie, co w post-darwinowskiej stylistyce określa się mianem *walki o byt*. W wyniku tejże walki dokonuje się *naturalna selekcja* osobników najlepiej przystosowanych, co rozumie się tak, że osobniki słabsze giną, silniejszym zaś udaje się przetrwać i wydać na świat potomstwo. W ten sposób do kolejnych pokoleń, czyli *populacji*, przechodzą organizmy coraz lepiej przystosowane do wymagań środowiska.

Elementem niezbędnym opisanego zjawiska jest *dziedziczenie* cech. Jeśli ograniczyć opis do gatunków dwupłciowych, to w ich obrębie osobniki potomne przejmują cechy po swoich rodzicach; nie w całości jednak i nie wszystkie. Chociaż część własności pochodzi od jednego z rodziców, a część od drugiego, to dokładna zawartość wynikowej „mieszanki” nie jest przesądzona z góry. Na jej powstanie i ostateczny kształt wpływa bowiem, i to niezwykle mocno, *przypadek*; od niego zależy, które osobniki dobiorą się w pary, w jakich proporcjach ich cechy się połączą, a także to, czy pewne cechy doświadczą *mutacji*, tj. niewielkich losowych zmian swojego kształtu. Przypadek zatem wprowadza do ewoluującego gatunku zmienność, a ta wskutek wspomnianego wyżej mechanizmu naturalnej selekcji (tj. promowania jednostek lepiej przystosowanych) przyczynia się do coraz ściślejszego zespolenia kolejnych populacji z ich naturalnym środowiskiem.

Nakreślony wyżej obraz trzeba dopełnić jednym jeszcze elementem. Otóż obraz ten ma naukowy sens pod warunkiem, że istnieje w przyrodzie jakiś uniwersalny nośnik dziedziczonych cech. Jak wiadomo, nośnik taki istnieje, a jest nim substancja organiczna, tzw. *kwasy DNA*, którego

najprostsze składniki tworzą coś na kształt alfabetu kodującego wszelkie cechy organizmów żywych. Kwas DNA i związany z nim biologiczny kod DNA wypełniają każdą komórkę organizmu, a w momencie zapłodnienia, czyli powstania pierwszej komórki nowego organizmu, kody rodziców łączą się, współtworząc kod potomka. W ten sposób powstaje nowy *genotyp*, czyli zbiór cech, które mogą (choć nie muszą) ujawnić się w przyszłym, rozwiniętym już, organizmie; o tym, które własności się ujawnią, decyduje źródło indywidualnych doświadczeń osobnika, czyli jego środowisko (zbiór cech uaktywnionych za pomocą środowiska nazywamy z kolei fenotypem).

Od strony naukowej badaniem kodu DNA, jak również procesów związanych z jego przekazywaniem, zajmuje się *genetyka*. Dyscyplina ta sprawia zatem, że teoria ewolucji jest możliwa. A mówiąc inaczej: genetyka i teoria ewolucji są ze sobą w sposób konieczny splecione i to one razem współtworzą naukowy obraz procesu, który zidentyfikował po raz pierwszy Karol Darwin.

§2. Od biologii do przetwarzania danych

Traktując *biologiczny darwinizm* jako teorię dobrze opisującą ewolucję organizmów w przyrodzie, możemy pokusić się o myśli nieco bardziej karkołomne. Karkołomne, bo wychodzące poza naturalną dziedzinę teorii ewolucji, tj. świat przyrody ożywionej. Dlaczego bowiem nie przyjąć, że w świecie maszyn będą zachodzić kiedyś te same zjawiska co w naturze: np. walka o byt, selekcja i dziedziczenie cech. Dlaczego nie rozwinąć narzucającej się obserwacji, że środowisko nasze staje się coraz bardziej nienaturalne, i już niebawem powstaną w nim nowe ekologiczne nisze, które wypełnią nasze wytwory, tj. nowe gatunki maszyn. Zgodnie z tym, co podpowiada teoria ewolucji, gatunki te będą żyły i ewoluowały obok nas.

2.1. Są to oczywiście myśli spod znaku *science fiction*. Grubą niezręcznością byłoby ich rozwijanie w książce, która ma ambicję trzymać się faktów, tj. bieżących dokonań informatyki.

Okazuje się jednak – co być może złagodzi surowy osąd niektórych Czytelników – że pośród informatycznych faktów są już dziś pewne techniki, które nawiązują wprost do teorii ewolucji. Nazywa się je *genetycznymi* lub szerzej: *ewolucyjnymi*. I choć stosuje się je w skali mikro, do rozwiązywania konkretnych problemów (a nie do kierowania ewolucją maszyn na szeroką

skale), to sam fakt ich istnienia świadczy o rodzącym się mariażu między informatyką i biologią.

Aby zrozumieć lepiej o czym mowa, wyobraźmy sobie sytuację następującą. Dostajemy do ręki program komputerowy, który ma rozwiązywać tzw. *problem komiwojażera*. Przypomnijmy, że problem ten polega na wyznaczeniu najkrótszej drogi objazdu n miast przy założeniu, że miasta te łączy zadana z góry siatka dróg. Wymieniony w nazwie problemu komiwojażer staje zatem przed niełatwym zadaniem: wyruszyć z punktu A, odwiedzić każde miasto dokładnie raz i wrócić do punktu A, pokonując możliwie najkrótszy dystans.

Z punktu widzenia informatyki problem komiwojażera należy do trudnych, a to dlatego, że wszelkie znane algorytmy do jego rozwiązywania mają *złożoność wykładniczą*. Znaczący to, że wraz ze wzrostem rozmiaru danych początkowych (tj. liczby miast) liczba operacji przewidzianych przez algorytm rośnie wykładniczo (tj. niezmiernie szybko). Wskutek tego dla odpowiednio dużych danych problem ten jest praktycznie nierozwiązywalny, a więc nie można stwierdzić, czy wskazana trasa objazdu miast jest rzeczywiście najkrótsza. To zaś powoduje, że dla uporania się z zadaniem komiwojażera informatycy poszukują wciąż nowych, niestandardowych strategii, w tym ewolucyjnych (zob. też esej drugi, p.t. „Czy komputery mogą być nieobliczalne”).

Wróćmy jednak do naszego rekwizytu, czyli programu rozwiązującego problem komiwojażera.. W jego instrukcji czytamy, że działa on w sposób **ewolucyjny**, a poszczególne kroki ewolucji można śledzić na ekranie. Zaintrygowani tą informacją uruchamiamy aplikację.

Na ekranie pojawia się obrazek wędrownego sprzedawcy na tle mapy gęstej od punkcików oznaczających miasta. Pod obrazkiem komputer wyświetla pytanie o „rozmiar problemu”, tj. liczbę miast. Załóżmy, że decydujemy się na 30 miast i opcję automatycznego generowania siatki połączeń. Po zatwierdzeniu wyboru przyciskiem OK w okienku programu pojawia się 30 kwadratowych punktów oraz sieć łączących je dróg z numerami i etykietami długości. W dolnej części okienka widzimy napis „POPULACJA POCZĄTKOWA”, a pod nim 50 sekwencji liczbowych; każda z nich reprezentuje jedno hipotetyczne rozwiązanie naszego problemu, tj. ciąg numerów kolejnych dróg, łączących ze sobą kolejne miasta. Przy każdym rozwiązaniu widnieje podsumowanie, czyli łączny dystans do pokonania

(oczywiście im dystans mniejszy, tym lepiej). Obok rozwiązań widzimy kolumnę o nazwie „Osobniki”, a w niej ciągi zer i jedynek; domyślamy się, że tak wyglądają binarne kody rozwiązań, czyli kolejnych sekwencji numerów dróg.

Obejrawszy to wszystko, klikamy przycisk „Dalej”. Tym razem na ekranie pojawia się zbiór par, a nad nim informacja: *„Przyjmując prawdopodobieństwo krzyżówki $p_k=0.3$, wybrano następujące pary osobników, które zostaną skrzyżowane i wydadzą potomstwo”*. Jej treść pasuje aż nadto do tego, co wyczytaliśmy w instrukcji: że program naśladuje reguły naturalnej ewolucji. Gdy zatwierdzamy informację kliknięciem, obok każdej pary pojawiają się dwa osobniki potomne; zwykły rzut oka wystarczy, by stwierdzić, że przypominają one rodziców: niektóre łańcuchy zer i jedynek pochodzą od jednego z rodziców, inne od drugiego.

Kolejny ekran ma nagłówek „Oto nowa populacja”, a wypełniają go wszystkie wygenerowane przed chwilą osobniki potomne. Nad ich zestawieniem komputer informuje: *„Przyjmując prawdopodobieństwo mutacji $p_m=0.04$, wybrano do mutacji następujące osobniki: 3, 15, 34, 49”*. I faktycznie, wewnątrz ciągów o podanych numerach dostrzegamy nieliczne (wyróżnione podświetleniem) zmiany zer na jedyneki i jedynek na zera.

Po dobrej chwili ekran znika, a w zamian ukazuje się zagadkowa podobna *koła ruletki*, którego brzeg tworzą różnokolorowe odcinki opatrzone numerkami kolejnych osobników. Napis nad obrazkiem głosi „ETAP SELEKCJI”. Domyślamy się zatem, że teraz nastąpi kluczowa dla teorii ewolucji procedura losowego wyboru osobników do nowej populacji. I tak właśnie się dzieje. Zagadkowe koło zaczyna wirować, a gdy zatrzymuje się, jego czarny wskaźnik dotyka zielonego odcinka na brzegu, opatrzonego numerem 41. Komputer oznajmia *„Do nowej populacji wybrano osobnika nr 41”*. Po chwili koło znów rusza...

Obrazowana przez ruch koła procedura selekcji powtarza się dokładnie 50 razy, czyli tyle razy, ile osobników mieści się w jednej populacji. W jej efekcie do nowego zbioru trafia dokładnie 50 osobników, które – przypomnijmy to – wcale nie są nowe, bo albo pochodzą z poprzedniej populacji, albo powstały już wcześniej przez krzyżowanie, a potem doznały losowych mutacji.

Nasz program pracuje oczywiście bez ustanku, generując kolejne zbiorowości potencjalnych rozwiązań. Począwszy od populacji drugiej rezygnujemy ze śledzenia poszczególnych etapów, zezwalając na ich automatyczny przebieg. Dotarłszy do populacji setnej, komputer zatrzymuje się, pozostawiając na ekranie jednego jedynego osobnika, który

został oceniony najwyżej. Osobnik ten reprezentuje najkrótszą znaną trasę objazdu 30 miast. Czy jest to faktycznie droga najkrótsza? Tego do końca nie wiemy, mamy jednak pewność, że komputer przedstawił nam najlepsze rozwiązanie spośród tych, które wyewoluowały.

2.2. I tak oto nasz wymaginowany program doprowadził komputer do rozwiązania, a nas do przybliżonej wiedzy o tym, jak działają *algorytmy genetyczne*. Jak można się było zorientować, nie są to algorytmy standardowe – prowadzące do rozwiązania krok po kroku, na podstawie zadanej z góry sekwencji instrukcji. Działają one wręcz przeciwnie: sterują generowaniem całej populacji potencjalnych rozwiązań, to znaczy ich doborem w parę, krzyżowaniem się, mutowaniem i nade wszystko promującą coraz lepsze rozwiązania procedurą selekcji. Ze względu na ten ostatni składnik programy ewolucyjne, przeciwstawia się często tradycyjnym mówiąc, że działają one nie na zasadzie instrukcji, lecz na *zasadzie selekcji*.

Aby nie pozostawić Czytelnika na poziomie jednego poglądowego przykładu i nabudowanego na nim opisu ogólnego, przedstawimy niżej dokładniejszy schemat przetwarzania danych na podstawie algorytmów genetycznych. Oto i on.

(0) DECYZJE WSTĘPNE

- Wybierz metodę kodowania osobników, tj. potencjalnych rozwiązań danego problemu, np. binarną czyli zerojedynkową. (Dalszy opis dostosujemy do reprezentacji binarnej)
- Wybierz rozmiar populacji (np. $n = 50$), prawdopodobieństwa krzyżówki (np. $p_k = 0.3$) i mutacji (np. $p_m = 0.01$)

(1) POPULACJA POCZĄTKOWA

- Wygeneruj losowo zerową populację $P_0=(O_1, O_2, \dots, O_m)$, tj. populację złożoną z osobników od O_1 do O_m .

(2) KRZYŻOWANIE

- Wybierz losowo, na podstawie prawdopodobieństwa p_k , kolejne pary osobników do skrzyżowania; włącz je do zbioru K .

Oto i procedura:

Dla każdego O_i generuj losowo liczbę L z przedziału $[0,1]$

- Jeśli $L < p_k$ wybierz O_i do skrzyżowania i dopisz go do zbioru K .

- *W przeciwnym razie pomiń O_i*

- W ramach każdej kolejnej pary (O_i, O_j) ze zbioru K wybierz losowo punkt cięcia. Utwórz osobniki potomne w ten sposób, aby pierwszy z nich zawierał początkowy fragment osobnika O_i (do punktu cięcia) i końcowy fragment osobnika O_j (od punktu cięcia); a drugi odwrotnie – początkowy fragment O_j i końcowy fragment O_i .
- W nowej populacji zastąp rodziców potomkami; resztę populacji pozostaw bez zmian.

(3) MUTACJA

- Wybierz losowo, na podstawie prawdopodobieństwa p_m , kolejne osobniki do zmutowania, włącz je do zbioru M .

Oto i procedura:

Dla każdego O_i generuj losowo liczbę L z przedziału $[0,1]$

- Jeśli $L < p_m$ wybierz O_i do zmutowania i dopisz go do zbioru M

- W przeciwnym razie pomiń O_i

- W każdym osobniku ze zbioru M zastąp losowo wybrane jedyнки zerami, a zera jedyнкami (z bardzo niskim prawdopodobieństwem).

(4) SELEKCJA

- Oblicz funkcje oceny poszczególnych osobników O_i , czyli $O_c(O_i)$
- Oblicz prawdopodobieństwa wyboru poszczególnych osobników p_i zgodnie z formułą $p_i = O_c(O_i) / F$, gdzie $F = \sum_{i=1..n} O_c(O_i)$
- Oblicz prawdopodobieństwa łączne q_i zgodnie z formułą $q_i = \sum_{j=1..i} p_j$
- Wybieraj losowo kolejne osobniki do nowej populacji

Oto i procedura:

- Generuj losowo liczby L z przedziału $[0,1]$.

- Za każdym razem, gdy liczba L mieści się w przedziale $[q_i, q_{i+1}]$, wybierz osobnika O_i do nowej populacji.

- Powtarzaj tę procedurę n razy (n – rozmiar populacji).

(5) PĘTLA LUB FINAŁ

- Mając nową populację (po selekcji), przejdź do kroku (2).
- Jeśli spełnione jest pewne kryterium finalne (np. wygenerowano wystarczająco dużo kolejnych populacji), zakończ sztuczną

ewolucję. W tym przypadku najlepszy osobnik z ostatniej populacji stanowi rozwiązanie problemu.

Powyższy opis nie pozostawia wątpliwości – zarówno w warstwie nazewniczej, jak i merytorycznej – że algorytmy genetyczne wywodzą się z biologicznej teorii ewolucji. Jądro algorytmu stanowi *selekcja*, w trakcie której sprawdza się, na ile dobrze tymczasowe rozwiązania spełniają warunki problemu. Warunki te odpowiadają środowisku naturalnemu (jeśli trzymać się terminów opisujących ewolucję w przyrodzie), a oceny poszczególnych osobników imitują stopnie przystosowania osobników do środowiska naturalnego (osobniki oceniane wyżej mają większe szanse przetrwać, tj. przejść do kolejnej populacji i wydać potomstwo).

Co ważne też, cała procedura ma charakter populacyjny; nie chodzi w niej o stopniową konstrukcję szukanego rozwiązania na podstawie danych wejściowych, lecz o generowanie całych populacji potencjalnych rozwiązań, przekształcaniu ich i wybieraniu spośród nich wariantów coraz bardziej obiecujących. Podsumowując zatem, algorytm genetyczny stanowi niestandardową, *populacyjno-selekcyjną* strategię rozwiązywania problemów, która wywodzi się z biologicznej teorii ewolucji.

W wielu opracowaniach technicznych (informatycznych) algorytmy genetyczne klasyfikuje się jako pewną *strategię heurystyczną*, to znaczy taką, która nie gwarantuje rozwiązania problemu, lecz wskutek użycia pewnej sprawdzonej gdzie indziej (tj. w przyrodzie) ogólnej metody daje nadzieję podążania w kierunku właściwego rozwiązania i znalezienia rozwiązań wystarczająco dobrych.

Innego rodzaju strategię heurystyczną stosuje się np. w algorytmach szachowych, gdzie do oceny jakości posunięć wykorzystuje się sprawdzone w rzeczywistej grze wskazówki ekspertów (np. nie poddawaj hetmana, gdy masz możliwość poświęcenia innej figury).

§3. Ku czemu zdają się prowadzić techniki ewolucyjne

Mając na uwadze fakt istnienia ewolucyjnych technik przetwarzania danych, a także ich prototypowy przykład (tj. algorytmy genetyczne), możemy podążyć myślą ku przyszłości i zastanowić się, jak techniki te mogłyby wzmocnić potencję naszych skomputeryzowanych wytworów czyli robotów. Rysują się tu co najmniej trzy, dopełniające się wzajem, scenariusze.

3.1. Scenariusz pierwszy nawiązuje najściślej do bieżących zastosowań informatyki. Zgodnie z nim roboty będą rozwiązywać problemy (przynajmniej niektóre) na *zasadzie selekcji*; będą zatem generować całe populacje próbnych rozwiązań, mieszać je ze sobą, zmieniać losowo pewne ich elementy, a następnie selekcjonować spośród nich rozwiązania coraz lepsze. Mówiąc obrazowo: zamiast zbliżać się do celu krok po kroku, z planem w ręku, wąską i dobrze wydeptaną ścieżką (pełną – rzecz jasna – nawrotów, zakrętów, pętli i rozgałęzień), będą nasi sztuczni naśladowcy iść szeroką ławą, bez planu, wypróbowując nowe kierunki i nowe ścieżki.

Jeśli dopełnić taki obraz zdolnością zapamiętywania wyników wiedzotwórczej ewolucji; jeśli dołączyć do tego nauczycieli i trenerów, którzy na podobieństwo środowiska naturalnego mogliby oceniać cząstkowe rozwiązania, to otrzymalibyśmy cyborgi uczące się niczym dzieci. Cyborgi takie byłyby zdolne poszerzać swoją wiedzę na podstawie specyficznej, bo opartej na teorii ewolucji, metody prób i błędów (zob. też esej 5).

Co ciekawe, naszkicowany obraz przewidywał mgliście już Alan Turing, a więc w pewnym sensie ojciec podejścia przeciwnego, tj. automatyzacji rozumowań na zasadach instrukcji, a nie selekcji (zob. pkt 2.2.).

Świadczy o tym następujący fragment głośnego artykułu *Computing Machinery and Intelligence* z roku 1956: „Każdą taką maszynę (tj. maszynę rozwiązującą problemy – P.S.) trzeba poddać próbie uczenia, ocenić jak dobre osiąga wyniki, i porównać to z wynikami innych maszyn. Przyjawszy następujące odpowiedniości: budowa mechanicznego dziecka – materiał dziedziczny; zmiany w budowie maszyny – mutacje; dobór naturalny – opinia eksperymentatora; cały ten proces możemy porównać do ewolucji naturalnej¹”.

Podkreślić warto, że intuicje Turinga idą nawet dalej niż powyższy scenariusz, bo postulują rywalizację maszyny z innymi maszynami (a nie tylko rywalizację rozwiązań w maszynowej pamięci).

Posłużyliśmy się wyżej dziwnym i raczej mało spotykanym terminem „ewolucja wiedzo-twórcza”. Wydaje on się ze wszech miar trafny, ponieważ domysły nasze nie dotyczą ewolucji w tradycyjnym rozumieniu, tj. ewolucji w wymiarze gatunkowym, która polega na selekcji naturalnej tworzących gatunek osobników; dotyczą natomiast procesu rozwiązywania problemów.

¹ Cytat pochodzi z książki *Filozofia umysłu* (pod red. B. Chwedeńczuka), w której zamieszczono cały przekład oryginalnego artykułu Alana Turinga (Wydawnictwo Spacja, Warszawa 1995, s. 295).

Znaczy to, że sytuują ewolucję wewnątrz pojedynczych sztucznych osobników, a konkretnie wewnątrz sterujących nimi programów. Ponadto zaś upatrują w niej pewnej metody budowania wiedzy, która to wiedza może być użyteczna przy podejmowaniu decyzji, interakcji z otoczeniem, działaniu itp.

Zauważmy na koniec, że postulowana na poziomie sterujących robotami programów wiedzo-twórcza ewolucja ma tę przewagę nad innymi metodami, że wprowadza do sztucznego robociego myślenia elementy *inwencji*. Dlaczego? Otóż, zgodnie z wyjaśnieniami w punkcie 2.1, opiera się ona na wyborach losowych – przypomnijmy, że w przypadku algorytmów genetycznych to los decyduje o mutacjach, krzyżówkach i przebiegu selekcji. Losowość zaś, a razem z nią nieprzewidywalność pomysłów, rozwiązań i decyzji, są cechami swoistymi ludzkiej inwencji.

Czy jednak hipotetyczna, oparta na regułach ewolucji, inwencja robotów ma szansę być tak samo pozytywnie ukierunkowana jak inwencja ludzka? Czy będzie zatem prowadzić do wartościowych wyników, a nie jakichś posunięć zupełnie chaotycznych? Wydaje się, że TAK, a decyduje o tym wpisana w każdą ewolucję – i naturalną, i sztuczną – reguła selekcji. Reguła ta promuje osobniki (tj. rozwiązania) coraz lepiej przystosowane do warunków środowiska (tj. wymagań postawionego problemu), a w dłuższym horyzoncie czasowym może doprowadzić do narodzin osobnika przystosowanego perfekcyjnie (tj. najlepszego rozwiązania).

3.2. Pora nakreślić scenariusz drugi, który zawdzięcza swój kształt obserwacjom neurobiologów, próbujących rozwikłać zagadkę ludzkiego mózgu. Niektórzy z nich twierdzą, że wewnętrzne struktury ludzkiego mózgu nie tylko rozrastają się i rozwijają, lecz także ewoluują. Proces ten miałby zachodzić na poziomie najmniejszych cegiełek układu nerwowego, tj. neuronów, a jego największe nasilenie miałyby przypadać na okres życia płodowego i niemowlęcego. To wtedy właśnie pojedyncze neurony i ich grupy miałyby walczyć ze sobą o przetrwanie – te silniejsze zajmowałyby miejsca w najważniejszych strukturach wewnątrz-mózgowych, te słabsze ginęłyby lub trafiały w obszary mniej eksploatowane. W bardzo grubym zarysie tak właśnie przedstawia się idea *selekcji grup neuronowych* w mózgu, propagowana przez G.M. Edelmana.

Opartą na tej idei teorię przybliżył w sposób popularno-naukowy książka p.t. „Przenikliwe powietrze, jasny ogień. O materii umysłu” (PIW 1998). W książce tej G. M. Edelman niezwykle mocno podkreśla, że ludzki mózg jest systemem selekcyjnym (rozwijającym się i

działającym na zasadach ewolucji), a nie algorytmicznym (realizującym gotowe programy). Odróżnia przy tym ewolucję *gatunkową*, która określa ogólno-gatunkowe cechy mózgu, ingerując stopniowo w kształt kodu genetycznego człowieka, od ewolucji *osobniczej*, która ma odpowiadać za kształtowanie się struktur konkretnego mózgu (nie zmienia natomiast kodu genetycznego). Właśnie ten drugi proces – proces zachodzący wewnątrz indywidualnych mózgów – jest przedmiotem teorii selekcji grup neuronowych.

Przenosząc koncepcje neurobiologów w dziedzinę układów sztucznych, do czego motywują nas dodatkowo pewne idee informatyków (o tym za chwilę), możemy wysnuć myśli następujące. Sterujące robotami oprogramowanie będzie projektowane w postaci zarodkowej, następnie zaś będzie rozwijało się – podobnie jak ludzki mózg – na drodze *mikro-ewolucji* swoich składników. Znaczący to, że pewne elementy opracowanego wstępnie programu (np. jakieś parametry liczbowe procedur czy całe procedury) będą „walczyły” ze sobą o przetrwanie, a więc albo o pozostanie wewnątrz właściwego programu, albo o zajęcie ważnego miejsca w jego wnętrzu.

Do wiary w taki scenariusz skłania nas dodatkowo wiedza o pewnej, stosowanej już dziś, strategii informatycznej o nazwie **COGANN** (z ang. *Combinations of Genetic Algorithms and Neural Networks*). Zgodnie z brzmieniem przytoczonej nazwy strategia ta polega na łączeniu algorytmów genetycznych ze sztucznymi sieciami neuronowymi, a jeden z jej wariantów przewiduje taki oto kształt sztucznej ewolucji. Dana jest wstępnie zdefiniowana sieć neuropodobna SN, która ma realizować w przyszłości konkretne zadanie, np. rozpoznawać pisane odręcznie litery (więcej informacji o sieciach zawiera esej 4). Sieć SN poddaje się treningowi ewolucyjnemu, to znaczy pozwala się, by pewne jej podsieci zmieniały losowo swój kształt (tj. strukturę połączeń między sztucznymi neuronami), rywalizując w ramach pewnej populacji podsieci konkurencyjnych. Na kolejnych krokach ewolucji sprawdza się, na ile dobrze cała sieć realizuje swoje przyszłe zadanie. Oceny takie stają się podstawą generowania kolejnych populacji rywalizujących ze sobą wewnątrz-sieciowych struktur. Przyjmuje się, że po dostatecznie wielu cyklach ewolucji sieć SN zostanie uformowana ostatecznie i będzie mogła dobrze wypełniać swoją funkcję.

W informatyce stosowanej strategia COGANN ma wiele szczegółowych wariantów, które rozróżnia się ze względu na „podział ról” między łączonymi ze sobą – w ramach realizacji danego zadania – technikami, tj. algorytmem genetycznym (AG) i sztuczną siecią

neuronową (SSN). Wypunktujmy: 1) SSN i AG realizują to samo zadanie, ale w inny sposób, 2) AG realizuje zadanie, a SSN jest używana wewnątrz AG jako narzędzie wspomagające, 3) SSN realizuje zadanie, ale AG przygotowuje SSN do działania, 4) AG i SSN współdziałają ze sobą w realizacji jednego złożonego zadania.

Przytoczone wyżej zastosowanie strategii COGANN stanowi coś pośredniego między opisaną w punkcie 3.1 ewolucją wiedzo-twórczą (wszak chodzi tu o przygotowanie sieci do rozwiązywania konkretnego problemu, tj. rozpoznawania liter), a postulowaną w tym punkcie mikro-ewolucją. Ten drugi czynnik jednak przeważa. Decyduje o tym, że jakkolwiek sieć uczy się wypełniać konkretne zadanie, to czyni to w drodze mikro-ewolucji swojej struktury, w trakcie której ewoluują konkurujące ze sobą podsieci.

Powtórzmy zatem – tym razem w stylistyce „sieciorowej” – główną myśl scenariusza ewolucyjnego nr 2. Jeśli hipotetycznymi robotami przyszłości będą kierować złożone wielofunkcyjne sieci neuropodobne, to sieci takie będą się rozwijać stopniowo, od postaci zarodkowej do dojrzałej, a proces ten będzie polegał na ewolucji ich wewnętrznych struktur, nazwanej przez nas mikro-ewolucją..

3.3. Mając świeżo w pamięci strategię COGANN, wyobraźmy sobie – w formie ćwiczenia przybliżającego kolejny scenariusz robociej ewolucji – taką oto sytuację. Pewien zespół informatyków zмага się od lat z problemem wynalezienia maksymalnie efektywnego programu do gry w szachy (nawiasem mówiąc jest to typowe pole badawcze specjalistów od sztucznej inteligencji). Na pewnym etapie badań zespół testuje technikę rywalizujących ze sobą sztucznych sieci neuronowych. W tym celu sprzęga ze sobą – w myśl idei COGANN – dwa algorytmy: sieciowy i genetyczny. Ujmując rzecz operacyjnie, daje to wynik następujący: wstępnie zaprojektowane sieci grają ze sobą w ramach kolejnych populacji; te, które grają lepiej, krzyżują się, dają potomstwo i zyskują większą reprezentację w kolejnych populacjach; te, które grają słabiej, odpadają. Zespół ma nadzieję, że odpowiedni dobór typu sieci oraz parametrów sterującego ewolucją genetycznego algorytmu, doprowadzi do powstania układu grającego perfekcyjnie.

Zauważyć trzeba, że w przeciwieństwie do scenariuszy 1 i 2 w powyższym przykładzie mamy do czynienia z *ewolucją właściwą*, tj. gatunkową. Mówiąc obrazowo: nasz wyimaginowany zespół pragnie wyhodować gatunek sieci dobrze grających w szachy, hodowla zaś rozwija

się na podobieństwo gatunków naturalnych, których przedstawiciele rywalizują ze sobą w realnym środowisku.

Przenosząc ten obraz daleko poza dziedzinę wyspecjalizowanych sieci i daleko poza dzisiejsze laboratoria informatyków, wkraczamy w orbitę spekulacji *futurologicznych*. Spekulacji śmielszych i bardziej ryzykownych niż dotychczasowe. Mając tego świadomość, popuśćmy jednak wodze fantazji. Wyobraźmy sobie żyjące obok nas (a w wariacie pesymistycznym: bez nas) gatunki robotów, które rodzą się, rozmnażają, konkurują ze sobą i giną. Być może znajdą się wśród nich tylko cyborgi wyspecjalizowane (np. jakieś spec-roboty do gry w szachy), a być może pojawią się automaty wielofunkcyjne, podobnie wszechstronne jak człowiek.

To właśnie taki fantastyczno-naukowy scenariusz stanowi ostateczne dopełnienie wizji sztucznej ewolucji, która – przypomnijmy to – miałyby zachodzić na *trzech poziomach*: po pierwsze, na poziomie rozwiązywania problemów i budowania wiedzy (ewolucja wiedzo-twórcza), po drugie, na poziomie autonomicznego rozwoju indywidualnych programów (mikro-ewolucja), i po trzecie, na poziomie rozwoju całych populacji programów (ewolucja gatunkowa).

Mimo wykończenia naszej informatyczno-futurologicznej konstrukcji, nie stawiajmy jednak kropki nad „i”. Wsuńmy na koniec pewne zagadnienia otwarte, które skierują uwagę czytelnika ku drugiej części książki.

Oto zagadnienie pierwsze. Czy strategia *antropomorficzna*, polegająca na rzutowaniu obserwacji dotyczących nas samych w dziedzinę naszych przyszłych wytworów (tj. robotów i sterujących nimi programów) jest jedynie możliwa? Czy wsparte autorytetem biologii i psychologii spostrzeżenie, że to zasady ewolucji kierują naszym rozwojem (gatunkowym), dojrzewaniem (osobniczym) i rozwiązywaniem problemów, musi stanowić jedyną podporę naszych przewidywań? Czy nie należy sądzić, że autonomicznie myślący matematycy, logicy, inżynierowie i wynalazcy nie obmyślą innych – lepszych i lepiej matematycznie uzasadnionych – metod niż ewolucyjne (np. odwołujących się do logiki)?

A oto i zagadnienie drugie. Czy prognozowane przez nas zastosowania technik ewolucyjnych mogą obyć się bez *organicznego budulca*? Czy nie jest istotny fakt, że ewolucyjny sukces człowieka wiąże się z konkretnym materiałem (białkowym), na którym dokonywały się i dokonują się ewolucyjne przeobrażenia? Czy własności tego materiału, a nie tylko struktura jego składników i struktura ich powiązań ze środowiskiem, nie są najbardziej istotne w inżynierskim naśladowaniu Natury? Mówiąc ogólniej:

czy projektując roboty, wystarczy naśladować zasady Natury (w tym ewolucji naturalnej), czy też trzeba jeszcze posługiwać się materią Natury?

Z wieńczącymi esej pytaniami nie pozostawiamy Czytelnika sam na sam. Niektóre z nich znajdują swoje rozwinięcia w części drugiej, zwłaszcza w eseju 12-ym p.t. „*Rzut oka na rozwój informatyki. Zagadka sztucznej inteligencji*”. Gdyby jednak Czytelnik oczekiwał, że rozwinięcia tam zawarte przejdą w rozstrzygnięcia, to musimy jego zapal ostudzić: zagadki umysłu wciąż bowiem będą czekać na przekonujące rozwiązanie.

WARTO PRZECZYTAĆ

Edelman G. M., *Przenikliwe powietrze, jasny ogień. O materii umysłu*, PIW, Warszawa 1998.

Rutkowska D., Piliński M., Rutkowski L., *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, PWN, Warszawa-Łódź 1997.

Stacewicz P., *Umysł a modele maszyn uczących się. Współczesne badania informatyczne w oczach filozofa*, Wydawnictwo EXIT, Warszawa 2010.