

ANDRZEJ STĘPNIK

O komputerowym wspomaganiu twórczego rozwiązywania problemów

Twórczość artystyczna, literacka, naukowa i wynalazczość w powszechnym mniemaniu uchodzą za dziedziny niepoddające się algorytmizacji¹. Stanowczym wyrazem tego poglądu w odniesieniu do odkrycia naukowego są słowa Poppera:

Uważam jednak – słusznie czy niesłusznie – że nie istnieje nic takiego jak logiczna metoda wpadania na owe [twórcze] pomysły lub logiczna rekonstrukcja owego procesu. Swoje stanowisko mogę ująć, mówiąc, że każde odkrycie kryje „element irracjonalny” albo „intuicję twórczą” w sensie Bergsona. [Popper, 2002, s. 26]

Na potwierdzenie przywołuje Popper stanowisko Einsteina, zaprzeczającego istnieniu „logicznej ścieżki” wiodącej do odkrycia praw naukowych i uznającego, że intuicja jest jedyną drogą do poznania owych praw [Popper, 2002, s. 26]. Takie ujęcie powoduje uznanie procesu odkrycia za irracjonalny, a przez to niemożliwy do racjonalnej rekonstrukcji, i przerzucenie uwagi na analizę wytworów tego procesu, czyli na logiczno-metodologiczną analizę uzyskanych rozwiązań.

¹ Algorytm można zdefiniować jako dokładny schemat postępowania, którego mechaniczne wykonanie prowadzi do rozwiązania danego problemu [Marciszewski, 1998, s. 64]. Zatem dany problem podlega algorytmizacji, gdy można skonstruować algorytm, który prowadzi do jego rozwiązania. I ostatecznie dana dziedzina podlega algorytmizacji, gdy (wszystkie, większość, niektóre – możliwa jest różna kwantyfikacja) jej problemy dają się zalgorytmizować.

Wydaje się jednak, że opierając się na psychologicznych badaniach dotyczących rozwiązywania problemów, twórczości i intuicji, możemy powiedzieć o wiele więcej na temat tego, co dzieje się podczas dochodzenia do twórczych rozwiązań. W związku z czym nasuwają się następujące pytania: Czy wiedza płynąca ze wspomnianych badań jest wystarczająca do tego, by procesy twórcze, a przynajmniej niektóre ich składniki, dało się zalgorytmizować? Czy i w jakim stopniu programy komputerowe mogą wspomagać nas w twórczym rozwiązywaniu problemów? Czy przynajmniej niektóre programy komputerowe zasługują na miano twórczych?

Niniejszy artykuł stanowi próbę odpowiedzi na te pytania. W części pierwszej przyjrę się wynikom badań nad twórczością i zastanowię się, na ile ich wyniki są pomocne przy projektowaniu programów wspomagających twórcze rozwiązywanie problemów. W drugiej przybliżę niektóre narzędzia informatyczne pomocne w tworzeniu programów generujących twórcze rozwiązania i spróbuję określić zakres ich stosowania. W podsumowaniu odpowiem na postawione pytania i zarysuję perspektywy rozwoju oprogramowania wspomagającego twórczość.

Zastosowania psychologicznych badań nad twórczością w informatyce

Rozważania z zakresu psychologii twórczości powinniśmy zacząć od przybliżenia samego pojęcia twórczości. Choć w psychologii wyróżniamy różnorakie definicje twórczości² odnoszące się zarówno do cech osobowych, procesów twórczych, jak i do wytworów tych procesów, to wyjściowa intuicja, leżąca u podstaw tych definicji, pozostaje taka sama: twórczy wytwór posiada jednocześnie dwie cechy – jest nowy (oryginalny) i wartościowy [Nęcka, 2003, s. 11–34]. Nęcka zdaje sobie sprawę z tego,

² Tu i dalej będzie mowa o twórczości, ale warto zaznaczyć, że w tego typu kontekstach używa się też terminu „kreatywność”. Traktuję wyrażenia „kreatywność” i „twórczość”, „kreatywny” i „twórczy” jako synonimy (oczywiście w ograniczeniu do pewnych użyć, gdyż np. słowo „twórczość” stosuje się na oznaczenie zbioru wytworów, stanowiących dokonania danego twórcy, podczas gdy słowa „kreatywność” nie używa się w tym kontekście).

że nowość jest cechą stopniowalną (coś może być mniej lub bardziej nowe, oryginalne, i najczęściej przynajmniej do pewnego stopnia bazuje na tym, co stare) i względna, ponieważ jest orzekana w odniesieniu do aktualnego stanu (coś może być oryginalne dzisiaj, ale już nie za parę lat). Co więcej, ów stan można rozumieć co najmniej dwojako: jednostkowo i psychologicznie, ale też ponadjednostkowo i historycznie. Z tego względu Boden wyróżnia dwa rodzaje nowości: psychologiczną, sprowadzającą się do tego, że coś jest nowe dla danej jednostki, oraz historyczną, dotyczącą tego, że coś nie miało wcześniej miejsca w historii [Boden, 2009, s. 24]³. Podobnie jest z cechą bycia wartościowym: jest ona zarazem stopniowalna (coś jest mniej lub bardziej wartościowe), jak i względna, gdyż wytwór jest wartościowy tylko w odniesieniu do szeroko rozumianego problemu, na który ma być odpowiedzią (coś może sprawdzać się w odniesieniu do danego problemu lub pewnej klasy problemów, lecz nie mieć zastosowania do innego rodzaju problemów)⁴. Dodajmy, że obie nazwy – „nowy” i „wartościowy” – są nieostre, co przekłada się na nieostrość nazwy „twórczy wytwór”. Ponadto wyrażenie „twórczy wytwór” domaga się formalnie poprawnej i merytorycznie trafnej operacjonalizacji. Mimo to charakterystyka twórczego wytworu w kategoriach nowości i wartości w kontekście rozwiązania danego problemu wydaje się dobrym punktem wyjścia. Po-

³ Warto by jeszcze wprowadzić trzecie, pośrednie rozumienie: nowość społeczną jako coś nowego dla danej społeczności, grupy czy populacji.

⁴ Trzeba wyraźnie zaznaczyć, że artykuł koncentruje się na twórczym rozwiązywaniu problemów i w tym kontekście ujmuje się cechę bycia wartościowym. Można mieć bowiem wątpliwości, czy w każdym wypadku posiadanie przez wytwór wartości jest relatywizowane do pewnego problemu. Dla przykładu weźmy wartości estetyczne. Nic nie stoi na przeszkodzie, by walory estetyczne wytworów były uwzględniane w ocenie ich wartości zrelatywizowanej do rozwiązania problemu, np. w odniesieniu do problemu stworzenia wytworu wywołującego u odbiorców przeżycia estetyczne, czy problemu wytworzenia przedmiotu uznawanego przez odbiorców za piękny, pożądany, cenny itp. W kontekście wymienionych problemów rozwiązania uwzględniające przynajmniej pewne walory estetyczne wytworów są bardziej wartościowe – oczywiście *ceteris paribus* – od rozwiązań ich nieuwzględniających. Czy jednak dzieje się tak w każdym wypadku i dla każdego rodzaju wartości? Pozostawmy ten problem otwarty, zwracając uwagę z jednej strony na cel artykułu, a z drugiej na fakt, iż konieczność relatywizowania wartości wytworu do problemu, na który wytwór ma być odpowiedzią, w dużej mierze zależy od szerokości zakresu nazwy „problem”.

krywa się też przynajmniej z niektórymi ujęciami twórczości w odniesieniu do informatyki⁵.

Posiadając roboczą definicję twórczego wytworu, możemy przyjąć, że proces twórczy to taki proces, który prowadzi do wytworzenia czegoś nowego i wartościowego, a osoba twórcza to osoba zdolna do generowania wytworów nowych i wartościowych. Nasuwa się pytanie, czy analogicznie możemy mówić o twórczości w odniesieniu do komputerów, programów i algorytmów. Bezpośrednie zastosowanie psychologicznego rozumienia twórczości do informatyki skutkowałoby przyjęciem następujących definicji:

– dany wytwór programu komputerowego lub algorytmu jest twórczy wtedy i tylko wtedy, gdy stanowi nowe (w rozumieniu historycznym, a nie psychologicznym) i wartościowe rozwiązanie pewnego problemu;

– dany algorytm lub program⁶ jest twórczy wtedy i tylko wtedy, gdy jest w stanie wygenerować nowe i wartościowe rozwiązania danego problemu.

Jakkolwiek nie mam większych obiekcji przy uznaniu wytworów algorytmów i programów komputerowych za twórcze, to mam wątpliwości w przypadku określania algorytmów i programów mianem twórczych na podobieństwo uznawania pewnych osób za twórcze. Co stanowi tu o różnicy między osobą a odpowiednio zaprogramowaną maszyną? Pierwsza różnica może dotyczyć świadomości: człowiek jest świadomy, natomiast maszyna nie. Dokładnie rzecz ujmując, przynajmniej część ludzkich procesów i stanów psychicznych jest świadoma, natomiast obliczeniowe procesy i stany maszyny nie są przez nią uświadamiane (w każdym razie nic na to nie wskazuje). Nie wdając się w skomplikowaną dyskusję na temat natury świadomości, zauważmy, że w wypadku wielu odkryć procesy twórcze do nich prowadzące są nieświadome i dopiero ich rezultat przedostaje się do

⁵ Dobrym przykładem jest klasyczny tekst Boden [2009]. Czasami autorka dodaje trzecią cechę, a mianowicie bycie zaskakującym (*surprising*) – wydaje się jednak, że cecha ta stanowi „psychologiczną” pochodną cechy bycia nowym (oryginalnym).

⁶ Przypisywanie twórczości algorytmom i programom komputerowym jest o wiele bardziej adekwatne niż przypisywanie jej samym komputerom, gdyż tak naprawdę potencjalne twórcze działanie komputera (który stanowi *hardware*) jest w większości zależne od jego oprogramowania (od *software*’u).

świadomości odkrywcy. Idąc tym tropem, dlaczego nie uznać kogoś za twórczego, jeśli nawet cały proces twórczy, łącznie z analizą sytuacji problemowej, z oceną i wykorzystaniem uzyskanego rezultatu, przebiegałby poza świadomością? Z tego względu posiadanie świadomości nie wydaje się dobrym kryterium odróżniania twórców od nietwórców. Druga różnica wiąże się z intencjonalnością, czyli z własnością umysłu, dzięki której skierowany jest on na przedmioty i stany rzeczy niezależnie od istniejącego świata, oraz dzięki której stany umysłu dotyczą tychże przedmiotów i stanów rzeczy [Searle, 2010, s. 176]. Co więcej, intencjonalne mogą być nie tylko świadome stany umysłowe, ale również nieświadome. Z kolei istnieją silne racje przemawiające za tym, że komputer nie dysponuje intencjonalnością i działa na poziomie syntaktycznym, a nie semantycznym jak umysł ludzki [Searle, 1995]. Jeśli zatem, aby być twórczym, trzeba móc odnieść się – intencjonalnie – do problemu, to kryterium to sprawiałoby, że komputery nie mogą być twórcze. Komputery, operując na symbolach i nie odnosząc ich do niczego poza nimi, same nie są intencjonalne, gdyż intencjonalność zapewnia z jednej strony programista, a z drugiej użytkownik programu. Argumentem przeciw kryterium intencjonalności mogłoby być to, że czasem używamy określenia „twórczy” w stosunku do „twórców”, którzy nie odznaczają się intencjonalnością (mówimy np., że przyroda jest twórcza). Z kolei trzecia różnica dotyczy wkładu w proces twórczy. W wypadku komputera nasuwa się myśl, że tak naprawdę twórczy nie jest komputer ani nawet program, ale człowiek, który go stworzył. Ten argument nie sprowadza się do tego, że wkład twórczy rozkłada się – nawet nierówno – na programistę i maszynę (w przypadku twórczości ludzkiej też często mamy do czynienia z twórczymi działaniami grup, a nie pojedynczych osób), lecz do tego, że cały wkład twórczy w wypadku „twórczości” komputerów należy do człowieka (inżynierów, programistów, czasem użytkowników określonych programów). Tym samym mówienie o twórczych programach tak naprawdę byłoby potwierdzeniem twórczego potencjału ich twórców. Podsumowując, uważam, że adekwatne jest określanie wytworów pewnych algorytmów i programów mianem twórczych, ale to nie przekłada się automatycznie – jak w wypadku ludzi – na stwierdzenie, że owe algorytmy czy programy są twórcze.

Po dyskusji zastosowania psychologicznej kategorii twórczości do informatyki przejdźmy do nakreślenia wyników badań nad procesami twórczymi. Zaczniemy przegląd od klasycznego podejścia do procesów twórczych. Według reprezentatywnej dla tego nurtu koncepcji Wallasa na proces twórczy składają się cztery fazy [Nęcka, 2003, s. 42–43]:

1. przygotowawcza (preparacji) – zbieranie i porządkowanie informacji, definiowanie problemu i celów, a także pozostałe czynności wstępne;
2. inkubacji – nieświadoma praca nad problemem, podczas gdy świadomość zajęta jest czymś innym;
3. olśnienia (iluminacji) – problem zostaje rozwiązany, rozwiązanie zostaje uświadomione, co skutkuje nagłym olśnieniem (tzw. doświadczeniem Aha!);
4. sprawdzenia (weryfikacji) – rozwiązanie jest testowane pod kątem sensowności, poprawności i użyteczności w kontekście postawionego problemu.

Na pierwszy rzut oka takie ujęcie procesu twórczego jawi się jako mało użyteczne i to nie tylko w kontekście informatyki. Łatwo ulec wrażeniu, że kluczowa część procesu twórczego – faza inkubacji – jest czymś, na co mamy znikomy wpływ. Jak się jednak okazuje, dzięki podejmowanym działaniom możemy zwiększyć lub zmniejszyć prawdopodobieństwo jej wystąpienia [Kolańczyk, 1991, s. 61]. Po pierwsze, trzeba dokonać jak najdokładniejszej analizy problemu, uwzględniając ukryte założenia i wszelkie uwarunkowania mogące stanowić bariery na drodze do rozstrzygnięcia problemu. Po drugie, istotne wydaje się zainteresowanie problemem i silna motywacja wewnętrzna⁷ do jego rozstrzygnięcia, a nawet osiągnięcie sta-

⁷ „Motywacja wewnętrzna pochodzi z wnętrza: jednostka angażuje się w działanie dla samego działania, także pod nieobecność zewnętrznej nagrody. Sposoby spędzania czasu wolnego, jak jazda na rowerze, pływanie kajakiem czy gra na gitarze, są zwykle motywowane wewnętrznie. Ten rodzaj motywacji ma swoje źródła w wewnętrznych właściwościach, na przykład cechach osobowościowych czy szczególnych zainteresowaniach. Natomiast motywacja zewnętrzna pochodzi z zewnątrz i opiera się na nagrodach i karach. Obejmuje zachowania nakierowane na zewnętrzne skutki, przykładowo pieniądze, oceny szkolne, pochwały, a nie zachowania nastawione na likwidację wewnętrznego napięcia” [Zimbardo, Johnson, McCann, 2011, s. 62].

nu, który psychologowie nazywają *flow*⁸ (przepływem, uskrzydleniem). I wreszcie po trzecie, musi dojść do zmiany w zakresie uwagi – uwaga silnie skoncentrowana na problemie (można ją przyrównać do reflektora punktowego) powinna zostać zastąpiona przez uwagę swobodnie rozlewającą się na różne elementy (podobną do światła żarówki), przy czym tego typu fluktuacje uwagi mogą mieć zarówno charakter spontaniczny, jak i być wynikiem celowego działania. Niestety, zalecenia odnoszące się do motywacji i do zmiany w zakresie uwagi nie przydadzą się w kontekście tworzenia algorytmów i programów generujących twórcze rozwiązania. Istotne wydaje się natomiast podkreślenie znaczenia dokładnej analizy sytuacji problemowej. To właśnie na podstawie reprezentacji problemu w fazie inkubacji dochodzi do wygenerowania rozwiązania. Kluczowe zdają się tu procesy, które przekształcają reprezentację sytuacji problemowej w rozwiązanie problemu. Dlatego też przyjrzyjmy się temu, co dzieje się podczas fazy inkubacji⁹.

Fenomen inkubacji i olśnienia doczekał się wielu konkurencyjnych wyjaśnień [Dobrołowicz, 1995, s. 83–85]. Poincare przyjął, że w fazie inkubacji dokonywane są nieświadome procesy myślowe, których rezultat uświadomiony zostaje w fazie olśnienia. Z kolei inni badacze uznają inkubację za okres potrzebnego odpoczynku, po którym umysł znów może pracować wydajnie, dzięki czemu jest w stanie wygenerować nowe pomysły. Nie wyjaśnia to jednak, dlaczego okresy inkubacji trwają znacznie dłużej, niż byłoby to potrzebne do regeneracji, a także nie wyjaśnia inku-

⁸ Przepływ (*flow*) jako stan optymalny opisywany jest następująco: „Wiemy już, jak ludzie opisują wspólne cechy doświadczenia optymalnego: poczucie, że nasze umiejętności są odpowiednie do stawienia czoła bieżącym wyzwaniom w kierowanym celami i określonym zasadami systemie, który daje jasne informacje dotyczące jakości naszych działań. Stan koncentracji jest tak intensywny, że nie jesteśmy w stanie myśleć o czymkolwiek innym ani martwić się naszymi problemami. Znika samoświadomość, a poczucie czasu ulega zachwianiu. Czynność, która daje takie doświadczenia, jest tak zadowalająca, że ludzie chcą wykonywać ją dla niej samej, nie zważając na to, co mogą dzięki niej osiągnąć, nawet jeżeli jest trudna lub niebezpieczna” [Csikszentmihalyi, 2005, s. 135].

⁹ Współcześnie zwraca się uwagę na podobieństwo uczenia się mimowolnego i intuicji. W wyjaśnianiu intuicji wykorzystuje się też wprowadzone przez Damasio [Damasio, 1999] pojęcie markera somatycznego [Balas, 2003]. Nie znajduje to jednak zastosowania w odniesieniu do informatyki.

bacji, w których umysł nie odpoczywa, lecz pracuje nad rozwiązaniem innego problemu. To stanowi też trudność dla koncepcji Woodwortha, który istotę inkubacji upatruje w uwolnieniu się od barier psychicznych i błędnych nastawień, przeszkadzających w rozwiązaniu problemu. Podobną teorię proponuje Kozielecki. Z kolei dla Guilforda inkubacja sprowadza się do przekształcenia posiadanych informacji z wykorzystaniem wiedzy leżącej „głębiej” w magazynach pamięci i powiązanej w mniej oczywisty sposób z problemem. Dla Schuberta inkubacja wiąże się z przeglądem dotychczasowych rozwiązań i odrzucenia trywialnych, nietwórczych pomysłów. Natomiast Govan upatruje w inkubacji częściowego zahamowania aktywności lewej półkuli mózgowej i dopuszczeniu obrazów wytwarzanych przez prawą półkulę.

Moim zdaniem teorie sprowadzające fazę inkubacji wyłącznie do regeneracji czy pozbycia się niektórych barier nie wyjaśniają w pełni tego, co się wtedy dzieje. Istnieją poważne racje, by uznać, iż podczas omawianej fazy dokonują się operacje na reprezentacjach sytuacji problemowej, pytanie tylko, jakie. W ich dookreśleniu mogą nam pomóc wyniki badań nad wglądem.

Mayer definiuje wgląd jako proces, dzięki któremu podmiot nagle przechodzi od stanu, w którym nie potrafił rozwiązać problemu, do stanu, w którym potrafi to zrobić. Wgląd najczęściej pojawia się w przypadku myślenia kreatywnego (*creative thinking*)¹⁰, gdy podmiot wymyśla nowe rozwiązanie danego problemu [Mayer, 1996, s. 3]. W historii psychologii rozmaicie próbowano wyjaśnić mechanizm wglądu, między innymi traktowano go jako [Mayer, 1996, s. 8–25]:

– wypełnienie schematu (Otto Selz), polegające na uzupełnieniu problemu rozumianego jako koherentny system informacji zawierający pewne luki;

¹⁰ Niektórzy psychologowie odróżniają analityczną kreatywność (*analytical creativity*) od intuicyjnej kreatywności (*intuitive creativity*), czy też konceptualną kreatywność (*conceptual creativity*) od kreatywności pierwotnej (*primordial creativity*). Z fenomenem wglądu i intuicji związane są drugie człony owych podziałów [Csikszentmihalyi, Sawyer, 1996, s. 354–355].

- reorganizację informacji wizualnej (Wolfgang Kohler), sprowadzającą się do nowego uporządkowania elementów wizualnych do takiej postaci, która nasuwa rozwiązanie problemu;
- nowe sformułowanie problemu (Karl Duncker), w tym redefinicję celów i uporządkowanie posiadanych informacji na nowo;
- usunięcie umysłowych blokad (Karl Duncker), związanych z powszechnymi czy nawykowymi sposobami podejścia do problemu;
- znalezienie analogii do rozwiązywanego problemu (Max Wertheimer), która pozwala odkryć jego właściwą strukturę na podstawie już uchwyconej struktury analogonu.

Zwróćmy uwagę na to, że większość wyjaśnień istotę wglądu upatruje w reorganizacji reprezentacji sytuacji problemowej. Operacje na reprezentacji problemu mają na celu jej nowe uporządkowanie, uzupełnienie, redefinicję celów i wydobywanie struktury problemu, dzięki czemu możliwe będzie zastosowanie rozwiązania analogicznego problemu należącego do innej dziedziny¹¹. Jak się wydaje, dysponujemy narzędziami, które pozwalają na taką reorganizację.

Jednym z nich jest analiza morfologiczna zaproponowana przez Zwicky'ego [Zwicky, 1969; Nęcka, 2003, s. 209; Dobrołowicz, 1993, s. 175–175; Góralski, 1990]. Polega ona na wyróżnieniu podstawowych parametrów (wymiarów) sytuacji problemowej, a następnie wypisaniu wszystkich możliwych kombinacji uwzględniających różne wartości tych parametrów i dokładnym przyjrzeniu się każdemu z uzyskanych wariantów. Dzięki temu może się okazać, że pojawiły się wartościowe kombinacje, których pierwotnie nie braliśmy pod uwagę lub które – chociażby ze względu na różnorakie bariery twórczości – uważaliśmy za niewarte rozważenia.

Analiza morfologiczna należy do narzędzi często stosowanych i przynoszących wartościowe rozwiązania w rozmaitych dziedzinach. W celu przybliżenia metody pokażemy jej użycie marketingowe: w jednym z artykułów Anna Ujwary-Gil pokazuje wykorzystanie analizy morfologicznej do udoskonalenia opakowania jednego z produktów firmy Roleski [Ujwa-

¹¹ Doskonałym tego przykładem jest rozwiązywanie problemów inżynierskich przez odwołanie się do rozwiązań naturalnych, wypracowanych w toku ewolucji w odpowiedzi na podobne zadania.

ry-Gil, 2006]. Wyróżnione zostają następujące wymiary zadania, jakim jest stworzenie odpowiedniego opakowania produktu (w nawiasie podano wartości tych wymiarów):

- funkcjonalność opakowania (poręczność, dozowanie, informacja, ozdoba, konserwacja);
- forma opakowania (słoik, butelka, wiaderko, kubek, saszetka, pojemnik);
- wartość dodana (frajda, wiedza, wtórne użycie jako pojemnik na przetwory, zabawka, akwarium, kolekcjonowanie, prestiż);
- materiał (szkło, plastik, folia, karton, metal, porcelana, glina/skóra);
- rodzaj/styl (eleganckie, wysokiej jakości, okolicznościowe, promocyjne, proste, masowe);
- adresat produktu (catering, średnio zamożny, zamożny, w zależności od oferty, fast-foody, restauracje, dziecko).

Na podstawie tak skonstruowanej tablicy morfologicznej utworzono kombinacje parametrów (użyto słabszej wersji analizy morfologicznej w postaci macierzy odkrywczej Molesa), a następnie wybrano cztery rozwiązania, które dodatkowo rozwinięto z wykorzystaniem dodatkowej wiedzy o rynku i preferencjach konsumentów:

- okolicznościową (z okazji znaczących wydarzeń sportowych), szklaną butelkę o charakterze kolekcjonerskim i informacyjnym dla średnio zamożnych klientów;
- eleganckie, kolekcjonerskie, szklane kubki ozdobne dla dziecka, emitowane w seriach;
- eleganckie, porcelanowe kubki ozdobne dla zamożnych klientów, emitowane seryjnie;
- informacyjny kubek-zabawka wykonany z porcelany, wydawany okolicznościowo dla średnio zamożnych klientów.

Jak łatwo zauważyć, główna trudność w stworzeniu komputerowego programu do analizy morfologicznej leży w takiej reprezentacji sytuacji problemowej, która umożliwiłaby wyróżnienie podstawowych wymiarów i określenie dopuszczalnych ich wartości, a także w skonstruowaniu procedury, która wyłoni z ogromnej liczby rozwiązań te najlepiej rokujące. Sama procedura kombinatoryczna jest banalnie prosta do zaprogramowania.

Innymi słowy, to nie algorytmizacja tworzenia rozwiązań na bazie tablicy morfologicznej, ale utworzenie samej tablicy i wybór najlepszych kombinacji stanowi programistyczne wyzwanie.

Kolejnym niezwykle pomocnym narzędziem jest metoda tak zwanych pytań kontrolnych Osborne'a (inaczej pytań stymulujących heurystę), pozwalających na lepsze zrozumienie problemu i pokonanie barier twórczości. W 1977 roku Bob Eberle ułożył listę pytań kontrolnych w łatwy do zapamiętania akronim SCAMPER¹² [Szmidt, 2013, s. 539–540]:

S – Substitute (Zastąp): Czym można to zastąpić? Kim? Wprowadzić nowe składniki? Materiały? Procesy? Funkcje? Motywacje? Inne miejsce? Inne podejście? Jak rozwiązać inaczej?

C – Combine (Łącz): Połączyć z czymś innym? Zastosować mieszaninę? Połączyć cele, pomysły, funkcje, osoby, przedmioty? Dwa lub więcej w jednym?

A – Adapt (Zapożyczaj): Do czego to jest podobne? Od czego można zapożyczyć? Co można skopiować?

M – Modify, Minify, Magnify (Modyfikuj, Zmniejszaj, Powiększaj): Ulepszyć? Nadać nową formę? Zmienić barwę, ruch, dźwięk, zapach itp.? Coś dodać, o coś uzupełnić? Zwiększyć wymiary, czas, częstotliwość, wytrzymałość? Podwoić lub zwielokrotnić? Zrobić grubszym, większym itp.? Coś odjąć? Zmniejszyć wymiary, czas, częstotliwość, wytrzymałość? Zminiaturyzować? Uczynić mniejszym, niższym, lżejszym? Coś pominąć? Zmniejszyć straty? Osłabić? Skrócić?

P – Put to other uses (Nowe zastosowania): Użyć do innych celów?

E – Eliminate (Eliminuj): Co wyeliminować lub pominąć? Czy wszystko jest niezbędne?

R – Rearrange, Reverse (Przearanżuj, Odwróć): Wziąć pod uwagę przeciwieństwo. Przekręcić? Postawić do góry nogami? Zmienić kierunek? Przebieg? Wykorzystać paradoks?

¹² Angielskie słowo *scamper* oznacza trucht.

Przywołane pytania dają się przekształcić w procedury operujące na reprezentacjach sytuacji problemowej. W niektórych przypadkach jest to trudniejsze (np. S, A, P), gdyż wymaga odniesienia się do szerszej bazy wiedzy, z kolei w innych – szczególnie w M – komputery mogą być ogromnie pomocne, prezentując graficzne lub dźwiękowe reprezentacje zmodyfikowanych obiektów. Wydaje się, że skonstruowanie programu wspomagającego w ten sposób ludzką twórczość jest względnie łatwe, natomiast problematyczne jest stworzenie programu, który bez pomocy człowieka reprezentowałby dowolny problem, następnie stosował pytania kontrolne w postaci procedur działających na reprezentacjach problemu, a na koniec sam oceniał uzyskane w ten sposób rozwiązania.

Na koniec warto przyrzeć się dokonaniom wybitnego polskiego badacza twórczości – Edwarda Nęcki. Przedstawił on podstawowe metody heurystyczne, przydatne w twórczym rozwiązywaniu problemów [Nęcka, 1994, s. 57–69]. Jakkolwiek są one niezwykle ważne w kontekście twórczości ludzkiej, zdają się mieć niewielkie zastosowanie w przypadku komputerów. Mimo to podejście Nęcki jest możliwe do wykorzystania w informatyce. Jego zdaniem, twórczość jest wypadkową wielu elementów, w tym określonych zdolności, umiejętności, postaw, motywacji, które za pomocą odpowiednio dobranych ćwiczeń dają się rozwijać [Nęcka, 2003; Nęcka i in., 2005]. Nas najbardziej interesują zdolności poznawcze¹³, do których należą:

- abstrahowanie – umiejętność ta służy m.in. klasyfikowaniu obiektów w sposób nowatorski;
- dokonywanie skojarzeń – bardzo istotna jest umiejętność dokonywania nietypowych, odległych skojarzeń; niektórzy widzą w tym nawet podstawę twórczości;

¹³ Oprócz nich w programie rozwoju twórczości znajdują się ćwiczenia umiejętności społecznych (niezbędnych przy twórczej pracy w grupie), ćwiczenia wyrabiające odpowiednią motywację i ćwiczenia na przezwyciężenie psychologicznych barier [Nęcka i in., 2005]. Nie omawiam ich z tego powodu, że wyżej wymienione obszary, choć odgrywają ogromną rolę w wypadku twórczości ludzkiej, to nie znajdują zastosowania w odniesieniu do twórczości komputerowej (przynajmniej na obecnym etapie rozwoju komputerów i informatyki).

– rozumowanie dedukcyjne – umiejętność wyprowadzania wniosków, szczególnie z przesłanek kontrfaktycznych lub wniosków dotyczących sytuacji kontrfaktycznych;

– rozumowanie indukcyjne, rozumiane dość specyficznje, głównie w kontekście dostrzegania i tworzenia analogii – ułatwia tworzenie tzw. analogii trafnych, przydatnych w fazie analizy i rozumienia problemu, i tzw. analogii nośnych, wykorzystywanych do generowania pomysłów (np. dzięki zbudowaniu pomostu między pozornie różnymi dziedzinami czy sytuacjami problemowymi);

– transformowanie umysłowych reprezentacji – niezwykle ważne przy analizie sytuacji problemowej, gdyż pozwala na wyjście poza utarte schematy i silnie zakorzenione założenia.

Zauważmy, że przynajmniej część wymienionych zdolności daje się komputerowo zasymulować. Abstrahowanie i dokonywanie skojarzeń można uzyskać dzięki stworzeniu odpowiednich sieci semantycznych. Dysponujemy algorytmami pozwalającymi wyprowadzać logiczne wnioski ze zbioru przesłanek. Najwięcej trudności może sprawić dostrzeganie i tworzenie analogii, a także transformowanie, gdyż wymaga wyjścia poza reprezentację problemu i daną dziedzinę. Algorytmami, które przynajmniej dają na to nadzieję, zajmiemy się w następnym podrozdziale.

Użyteczne narzędzia programistyczne

Tworzenie programów wspomagających twórcze rozwiązywanie problemów nie jest łatwe. Aby tego dokonać, należy poradzić sobie z następującymi zadaniami:

- 1) wykreowaniem adekwatnej i elastycznej reprezentacji problemu, przy czym adekwatność sprowadza się do wyróżnienia podstawowych parametrów sytuacji problemowej, określenia ich możliwych wariantów i oceny istotności poszczególnych parametrów dla rozwiązywanego problemu, natomiast elastyczność do wyjścia poza posiadane założenia odnoszące się do wyróżniania parametrów, ich wariantów i istotności w kontekście rozwiązywanego problemu;

2) stworzeniem funkcji przekształcającej reprezentację sytuacji problemowej w rozwiązanie problemu przy wykorzystaniu:

– istniejących rozwiązań problemów (np. z innych dziedzin) wykazujących podobieństwo do analizowanej sytuacji problemowej;

– skojarzeń (np. w kontekście użycia odpowiednich sieci semantycznych);

– nietypowej klasyfikacji obiektów;

– wnioskowań dedukcyjnych i niededukcyjnych (np. o charakterze redukcyjnym: od poszukiwanego idealnego rozwiązania do warunków, jakie muszą być spełnione, by urzeczywistnić to rozwiązanie);

– metod kombinatorycznych;

3) utworzeniem procedury oceniającej wygenerowane rozwiązania pod kątem wyszukiwania tych najbardziej twórczych.

Każde z wymienionych zadań nastęca problemów w realizacji. Trudności zaczynają się już na poziomie reprezentacji sytuacji problemowej. Flasiński wręcz twierdzi, że nie da się stworzyć programu, który potrafiłby samodzielnie rozwiązywać dowolne problemy z tego względu, że człowiek jest niezbędny do opracowania modelu sytuacji problemowej.

Wracając natomiast do zagadnienia konstrukcji „rozwiązywacza problemów wszelakich” zauważmy, że jedyną sprawdzającą się tutaj strategią jest (heurystyczne) *przeszukiwanie przestrzeni stanów* i jego rozwinięcie w postaci *obliczeń ewolucyjnych*. Jednakże systemy oparte na tej strategii nie rozwiązują problemów ogólnych samodzielnie (autonomicznie), ale we współpracy z człowiekiem [...]. Wydaje się, że opracowanie metod pozwalających systemom sztucznej inteligencji na automatyczną konstrukcję *modelu opisu problemu* na podstawie percepcji tego problemu jest największym wyzwaniem w zakresie symulacji zdolności umysłowo-poznawczych. [Flasiński, 2011, s. 231]

Na tej podstawie moglibyśmy wyróżnić dwa wymiary odnoszące się do konstrukcji reprezentacji sytuacji problemowej:

– wielkość obszaru działania (od programów nastawionych na rozwiązywanie problemów z dobrze określonej, wąskiej dziedziny do programów rozwiązujących dowolne problemy);

– stopień pomocy ze strony człowieka (od programów, w których człowiek w pełni definiuje problem i jego podstawowe wymiary, a nawet

wartości tych wymiarów, do programów, które robią to w sposób całkowicie samodzielny).

Obecnie dysponujemy programami dość dobrze radzącymi sobie z rozwiązywaniem problemów w stosunkowo wąskich dziedzinach (np. w zakresie diagnostyki medycznej czy gry w szachy), przy czym opracowanie sytuacji problemowej i reguł działania dokonywane jest przy pomocy człowieka (programistów i ekspertów z danej dziedziny), czego przykładem są systemy eksperckie (czy ekspertowe) [Ficoń, 2013, s. 273–334]. Jednakże punktem dojścia powinien być program w rodzaju – cytując Flasińskiego – „rozwiązywacza problemów wszelakich”, który bez pomocy człowieka (lub z minimalną pomocą) potrafi utworzyć reprezentację dowolnej sytuacji problemowej i na tej podstawie wygenerować jej rozwiązania. Jak na razie jesteśmy daleko od stworzenia takiego programu¹⁴.

Pewne nadzieje można wiązać z powstaniem narzędzi informatycznych – takich jak algorytmy genetyczne i sieci neuronowe – które za sprawą symulacji naturalnych procesów radzą sobie w dziedzinach, w których standardowe algorytmy wypadają słabo.

Zacznijmy od algorytmów genetycznych, czy szerzej – algorytmów ewolucyjnych¹⁵ – stanowiących przeniesienie procesu ewolucji darwinowskiej na proces generowania rozwiązań danego problemu. Algorytmy genetyczne mają następującą konstrukcję¹⁶:

0) Etap wstępny – kodowanie problemu, czyli przestrzeni stanów (wszystkich potencjalnych rozwiązań) w języku binarnym; w tym kontek-

¹⁴ W 1984 roku zainicjowano projekt uniwersalnego systemu eksperckiego CYC, którego celem było wyposażenie komputera w „zdrowy rozsądek” i możliwość rozwiązywania na tej bazie różnorodnych problemów. Mimo ponad 20 lat pracy system ten nie spełnia pokładanych w nim nadziei i jak twierdzą niektórzy (np. H. Dreyfus) – nie spełni ich nigdy [Ficoń, 2013, s. 330–333]. Porażkę poniósł także stworzony w 1957 roku *General Problem Solver*, który słabo radził sobie z rzeczywistymi problemami [Ficoń, 2013, s. 102–103].

¹⁵ O relacji między algorytmami genetycznymi, ewolucyjnymi a programami genetycznymi można przeczytać w: Michalewicz, 2003, s. 17; Rutkowski, 2006, s. 229–299; Ficoń, 2013, s. 218–219.

¹⁶ Rekonstrukcja algorytmu genetycznego na podstawie: Leciejewski, 2014, s. 268–269. Ogólny schemat algorytmu genetycznego można znaleźć w: Gwiazda, 2007, s. 14; Michalewicz, 2003, s. 26.

ście dokonuje się także przedstawienie genotypu osobnika w postaci ciągu binarnego o stałej długości.

1) Stworzenie populacji początkowej składającej się z N osobników całkowicie losowych (randomizacja) lub N osobników wyznaczonych za pomocą pewnego algorytmu heurystycznego¹⁷.

2) Zastosowanie operatorów genetycznych – mutacji i krzyżowania – przekształcających kod genetyczny osobników; mutacja polega na zastąpieniu wylosowanego bitu wylosowanego osobnika wartością przeciwną, z kolei krzyżowanie na losowym łączeniu osobników w pary i zamianie ich fragmentów genotypów.

3) Policzenie wartości funkcji celu (funkcji przystosowania czy dopasowania) osobników – interpretacja genotypu każdego osobnika jako elementu przestrzeni stanów, czyli przykładowego rozwiązania problemu, a następnie ocena tego rozwiązania za pomocą funkcji celu¹⁸.

4) Dokonanie selekcji osobników najlepiej dostosowanych do środowiska, czyli najlepiej rozwiązujących dany problem.

5) Uczynienie populacji końcowej populacją bieżącą i powrót do punktu 2; wyjście z pętli następuje po określonym czasie lub po spełnieniu kryteriów odnoszących się do rozwiązania.

Bez wątpienia algorytmy genetyczne są w stanie generować rozwiązania nowe i wartościowe.

Często program komputerowy oparty na algorytmach genetycznych (programowanie genetyczne) wykazuje pewne cechy innowacyjności i działalności twórczej, co wynika z dość losowego i przypadkowego poszukiwania rozwiązania optymalnego. W odróżnieniu od człowieka i zwykłych algorytmów numerycznych nie działają one schematycznie i operują niekiedy modelami przybliżonymi, które generują też przybliżone rozwiązanie końcowe, ale spełniające kryterium dopuszczalności i pewne warunki brzegowe. [Ficoń, 2013, s. 224]

Algorytmy genetyczne znajdują szerokie zastosowanie: przy rozwiązywaniu problemów NP-zupełnych (np. problemu komiwojażera czy zada-

¹⁷ Leciejewski pisze tylko o losowym sposobie doboru populacji początkowej. Uzupełniłem jego opis na podstawie Michalewicz, 2003, s. 343–344.

¹⁸ Czasami zamiast pełnego wyznaczania funkcji dopasowania rozwiązań można porównać dwa rozwiązania i wybrać lepsze [Michalewicz, 2003, s. 355–356].

nia plecakowego), projektowaniu, optymalizacji, prognozowaniu i symulacji [Ficoń, 2013, s. 223–226]. Przykładem jest projekt Golem, gdzie przy użyciu algorytmów genetycznych projektuje się i programuje roboty przemysłowe, a wiele z tych projektów daje się określić jako nowe i wartościowe, a więc twórcze. Warto przy tym pamiętać, by algorytmów genetycznych nie stosować wąsko¹⁹. Mogą one bowiem służyć nie tylko do znajdowania funkcji przekształcających reprezentacje sytuacji problemowej w rozwiązania problemu, ale także do generowania nowych reprezentacji problemu, a nawet nowych kryteriów oceny uzyskanych rozwiązań. Można je także stosować do poszukiwania optymalnej architektury sieci neuronowej i uczenia jej wag [Rutkowski, 2006, s. 280–287].

Z algorytmami genetycznymi wiążą się pewne trudności.

Algorytmy genetyczne oprócz wielu zalet mają także swoje wady i ograniczenia epistemologiczne. Po pierwsze, aby uzyskać wynik, trzeba określić funkcję celu, a nie ma algorytmu mówiącego, jak to zrobić. Zwykle wynik zależy od doświadczenia danego programisty. Po drugie, nigdy nie mamy pewności, że rozwiązanie problemu jest optymalne (zwykle jest to rozwiązanie przybliżone, gdyż algorytm genetyczny jest algorytmem randomizowanym). [Leciejewski, 2014, s. 270]

Druga trudność jest mniej dla nas interesująca, gdyż rozwiązanie twórcze nie musi być zarazem optymalne (i bardzo często nie jest). Istotny jest natomiast problem braku algorytmów tworzenia funkcji celu, ponieważ z jednej strony stanowi kolejny argument za tym, że procesy twórcze nie dają się w pełni zalgorytmizować, a z drugiej wskazują na twórczy wkład człowieka jako niezbędny element komputerowej twórczości.

Wspomnieliśmy o sieciach neuronowych, warto więc je przybliżyć. Są to programowe lub sprzętowe symulatory modeli matematycznych, opisujących działanie sieci neuronowych, składające się z wielu wzajemnie połączonych (połączeniom przypisane są wagi) sztucznych neuronów, ułożonych w warstwy (wejściową, obliczeniową, wyjściową) [Ficoń, 2013,

¹⁹ Warto dodać, że algorytmy ewolucyjne są stale rozwijane; np. w postaci algorytmów kulturowych uwzględnia się wzajemne oddziaływanie ewolucji biologicznej i kulturowej [Reynolds, Sverdlík, 1993].

s. 134–172]. Niezwykle istotne jest to, że sieci mogą się uczyć²⁰. Wyróżnia się uczenie z nadzorem (z nauczycielem), podczas którego nauczyciel prezentuje właściwą odpowiedź na dane wejściowe, dzięki czemu sieć koryguje wagi połączeń synaptycznych, oraz uczenie bez nadzoru (bez nauczyciela), gdzie sieć nie otrzymuje właściwych odpowiedzi, lecz musi sama do nich dojść²¹.

Sztuczne sieci neuronowe wydają się obiecującym narzędziem, gdyż przejawiają własności podobne do ludzkiego mózgu. Według Kosińskiego [2014, s. 188] należą do nich:

- zdolności pamięciowe, zwłaszcza pamięci adresowanej kontekstowo, skojarzeniowej;
- umiejętność uczenia się na przykładach;
- umiejętność generalizacji²²;
- odporność na uszkodzenia sieci i zdolność do poprawnej pracy przy pewnym poziomie uszkodzeń sieci;
- umiejętność równoległego przetwarzania informacji;
- umiejętność pracy niealgorytmicznej;
- możliwość heurystycznego rozwiązywania problemów.

Z tej racji sieci neuronowe mogłyby zostać wykorzystane do rozpoznawania wzorów (w odniesieniu do sytuacji problemowej), prób zastosowania rozwiązań problemów mających podobną strukturę do danej sytuacji problemowej i szukania istotnych aspektów problemu.

Wykorzystanie sieci neuronowych do twórczości wiąże się jednak z co najmniej dwoma problemami. Pierwszy dotyczy uczenia: przy twórczym

²⁰ „Pod pojęciem uczenia sieci będziemy rozumieli wymuszenie na niej określonego reagowania na zadane sygnały wejściowe” [Żurada, Barski, Jędruch, 1996, s. 57].

²¹ „Przy uczeniu się bez nadzoru nie ma nauczyciela. W dalszym ciągu zajmujemy się siecią, która ma wejścia i wyjścia, ale nie istnieje sprężenie zwrotne od środowiska, mówiące, jakie te wejścia powinny być lub czy są prawidłowe. Sieć musi odkrywać sama dla siebie wzorce, właściwości, regularności, kategorie w danych wejściowych i kodować je w celu otrzymania wyjścia. Jednostki i połączenia muszą zatem okazywać pewien poziom **samoorganizacji**. Uczenie bez nadzoru może okazać się pożyteczne tylko wówczas, gdy istnieje **redundancja** w danych wyjściowych” [Hertz, Krogh, Palmer, 1993, s. 245].

²² Jak pisze Osowski: „Podstawową cechą sieci neuronowej jest jej zdolność do uogólniania, a więc generowania właściwego rozwiązania dla danych, które nie pojawiły się w zestawie danych uczących” [Osowski, 1996, s. 32].

rozwiązywaniu problemów rzadko dysponujemy nawet zarysem rozwiązania, z tego też względu problematyczne wydaje się uczenie z nadzorem; ponadto przy uczeniu z nauczycielem można mówić o ogromnym wkładzie twórczym tego ostatniego. Na drugie ograniczenie wskazuje Ficoń:

Niestety, sieć neuronowa nie jest dobrą propozycją w przypadku, gdy rozwiązanie danego problemu wymaga zastosowania wieloetapowego rozumowania [...] Sztuczna sieć neuronowa nie potrafi, tak jak mózg człowieka, podzielić złożonego zadania na mniejsze problemy, które mogą być rozwiązywane odrębnie [...]. W rezultacie należałoby dysponować całym systemem różnych sieci neuronowych rozwiązujących poszczególne zadania cząstkowe w kolejnych etapach. Na obecnym stanie badań każda z tych sieci może działać tylko samodzielnie i jednoetapowo, bez możliwości automatycznego pobierania wyników z poprzedniego etapu. [Ficoń, 2013, s. 157–158]

A tak się jednak składa, że większość trudnych problemów ma strukturę złożoną.

Podsumowanie

Boden wyróżnia trzy podstawowe metody generowania twórczych rozwiązań [Boden, 2004; 2009]: nowatorskie połączenie (*combination*) znanych elementów, zgłębianie (*exploration*) nieodkrytych dotychczas rejonów przestrzeni konceptualnej wyznaczonej przez – w większości niejawne – reguły, a także przekształcenie (*transformation*) tej przestrzeni przez odrzucenie pewnych ograniczających reguł, ich zmiana czy też dodanie nowych. Ze względu na odmiennosc tych metod można wyróżnić trzy rodzaje twórczości (kreatywności): kombinacyjną, eksploracyjną i transformacyjną. Nietrudno zauważyć, że w przypadku komputerów największą trudność przedstawia zaprogramowanie twórczości transformacyjnej.

Stawiam tezę – podobnie zresztą jak Boden – że odpowiednio zaprogramowane komputery, mimo wskazanych wyżej trudności, są w stanie generować twórcze wytwory na przedstawione wyżej wszystkie trzy sposoby. W tym celu warto użyć takich narzędzi, jak analiza morfologiczna Zwicky’ego czy metody pytań kontrolnych Osborne’a, przekształcając je na komputerowe programy przy wykorzystaniu algorytmów ewolucyjnych i sieci neuronowych. Kluczowy wydaje się tutaj udział człowieka w zapro-

gramowaniu maszyny, pomocy w reprezentacji sytuacji problemowej i zastanych reguł, a także w ocenie wytworów pracy komputera. Jak na razie jesteśmy daleko od sytuacji, w której samoprogramujący się „rozwiązywacz problemów wszelakich” w pełni samodzielnie znajduje twórcze rozwiązania dowolnego problemu z dowolnej dziedziny (od sztuki, muzyki i literatury poprzez problemy inżynierskie, ekonomiczne, polityczne i społeczne, aż do problemów naukowych), a następnie wybiera najlepsze z nich.

Dokładnie rzecz biorąc, twierdzę, że procesy twórcze w całej swojej rozciągłości (w odniesieniu do czterech faz Wallasa) nie dają się w pełni zalgorytmizować, co nie znaczy, że wszystkie ich składniki nie poddają się algorytmizacji. Wydaje się, że przynajmniej pewne z nich, jak chociażby niektóre rodzaje wnioskowania, dokonywanie klasyfikacji, skojarzeń, stosowanie metod kombinatorycznych, są możliwe do komputerowej symulacji. Trzeba jednak podkreślić niezbywalną, jak na razie, rolę człowieka w tworzeniu programów, definiowaniu problemu i ostatecznej ocenie wygenerowanych rozwiązań. Przynajmniej z tego względu należy powstrzymać się przed przypisywaniem komputerom, programom i algorytmom cechy bycia twórczym (kreatywnym), rezerwując określenie „twórczy” dla niektórych wytworów tych programów oraz ich autorów. Bez wątplenia warto kontynuować badania nad sztuczną inteligencją i komputerowym symulowaniem procesów twórczych, gdyż nie tylko prowadzą one do udoskonalenia naszych narzędzi informatycznych i wygenerowania interesujących rozwiązań konkretnych problemów z różnych dziedzin (przykłady zastosowań zostały przedstawione w tekście), ale także przyczyniają się do zwiększenia wiedzy na temat ludzkich zdolności umysłowych. Nie musimy obawiać się – przynajmniej w najbliższej przyszłości – że komputery całkowicie zastąpią nas w aktywności twórczej, mogą za to służyć cenną pomocą w twórczym rozwiązywaniu problemów.

Bibliografia

- Balas R., (2003), „Poznawcze i afektywne mechanizmy intuicji”, [w:] R.K. Ohme, M. Jarymowicz [red.], *Automatyzmy w regulacji psychicznej: nowe perspektywy*, Warszawa, Wydawnictwo Instytutu Psychologii PAN i SWPS, s. 79–88.

- Boden M.A., (2004), *The Creative Mind: Myths and Mechanisms*, London, Routledge.
- Boden M.A., (2009), „Computer models of creativity”, *AI Magazine*, s. 23–34.
- Csikszentmihalyi M., (2005), *Przepływ*, przeł. M. Wajda-Kacmajor, Taszów, Biblioteka Moderatora.
- Csikszentmihalyi M., Sawyer K., (1996), “Creative insight: the social dimension of a solitary moment”, [w:] R.J. Sternberg, J.E. Davidson [eds.], *The Nature of Insight*, Cambridge, Mass., The MIT Press, s. 329–363.
- Damasio A.R., (1999), *Bląd Kartezjusza*, przeł. M. Karpiński, Poznań, Rebis.
- Dobrołowicz W., (1993), *Psychika i bariery*, Warszawa, Wydawnictwo Szkolne i Pedagogiczne.
- Dobrołowicz W., (1995), *Myśleć intuicyjnie*, Warszawa, WNT, Fundacja Książka Naukowo-Techniczna.
- Ficoń K., (2013), *Sztuczna inteligencja nie tylko dla humanistów*, Warszawa, BEL Studio.
- Flasiński M., (2011), *Wstęp do sztucznej inteligencji*, Warszawa, Wydawnictwo Naukowe PWN.
- Góralski A., (1990), *Być nowatorem*, Warszawa, Wydawnictwo Naukowe PWN.
- Gwiazda T.D., (2007), *Algorytmy genetyczne. Kompendium*, t. I, Warszawa, Wydawnictwo Naukowe PWN.
- Hertz J., Krogh A., Palmer R.G., (1993), *Wstęp do teorii obliczeń neuronowych*, przeł. S. Jankowski, Warszawa, WNT.
- Kolańczyk A., (1991), *Intuicyjność procesów przetwarzania informacji*, Gdańsk, Wydawnictwo Uniwersytetu Gdańskiego.
- Kosiński R.A., (2014), *Sztuczne sieci neuronowe*, Warszawa, WNT.
- Leciejewski S., (2014), „Ewolucyjna teoria epistemologiczna metodą algorytmów genetycznych”, *Przegląd Filozoficzny – Nowa Seria*, 4(92), s. 263–273.
- Marciszewski W., (1998), *Sztuczna inteligencja*, Kraków, Znak.
- Mayer R.E., (1996), “The search for insight: grappling with gestalt psychology’s unanswered questions”, [w:] R.J. Sternberg, J.E. Davidson [eds.], *The Nature of Insight*, Cambridge, Mass., The MIT Press, s. 3–32.
- Michalewicz Z., (2003), *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Warszawa, WNT.
- Nęcka E., (1994), *TROp... Twórcze rozwiązywanie problemów*, Kraków, „Impuls”.
- Nęcka E., (2003), *Psychologia twórczości*, Sopot, Gdańskie Wydawnictwo Psychologiczne.
- Nęcka E., Orzechowski J., Słabosz A., Szymura B., (2005), *Trening twórczości*, Gdańsk, Gdańskie Wydawnictwo Psychologiczne.
- Osowski S., (1996), *Sieci neuronowe w ujęciu algorytmicznym*, Warszawa, WNT.
- Popper K. R., (2002), *Logika odkrycia naukowego*, przeł. U. Niklas, Warszawa, Aletheia.
- Reynolds R.G., Sverdluk W., (1993), “Solving problems in hierarchically structured systems using cultural algorithms”, [w:] D.B. Fogel, W. Atmar [eds.] *Proceedings of the Second Annual Conference on Evolutionary Programming*, La Jolla, CA, Evolutionary Programming Society, s. 144–153.

- Rutkowski L. (2006), *Metody i techniki sztucznej inteligencji*, Warszawa, Wydawnictwo Naukowe PWN.
- Searle J.R., (1995), *Umysł, mózg i nauka*, przeł. J. Bobryk, Warszawa, Wydawnictwo Naukowe PWN.
- Searle J.R., (2010), *Umysł. Krótkie wprowadzenie*, przeł. J. Karłowski, Poznań, Rebis.
- Szmidt K.J., (2013), *Pedagogika twórczości*, Sopot, Gdańskie Wydawnictwo Psychologiczne.
- Ujwary-Gil A., (2006), „Zastosowanie analizy morfologicznej w praktyce biznesowej”, *Marketing i Rynek*, nr 5, s. 24–30.
- Zimbardo Ph.G., Johnson R.L., McCann V., (2011), *Psychologia. Kluczowe koncepcje*, t. 2, przeł. M. Guzowska-Dąbrowska, J. Radzicki, E. Czerniawska, Warszawa, Wydawnictwo Naukowe PWN.
- Zwicky F., (1969), *Discovery, Invention, Research through the Morphological Analysis*, New York, MacMillan.
- Żurada J., Barski M., Jędruch W. (1996), *Sztuczne sieci neuronowe*, Warszawa, Wydawnictwo Naukowe PWN.

On computer-aided creative problem-solving

ABSTRACT. This article focuses on issues related to algorithmizing creative problem-solving processes and the possibility of carrying out computer simulations of these processes. The author shows that certain components of creative processes can be simulated by computer, and computers can generate creative products, but due to their limitations, which are specified in this article, computers are not able to simulate the full extent of creative processes that are characteristic of human beings.

KEY WORDS: artificial intelligence (AI), creativity, problem-solving, genetic algorithms, neural networks

Andrzej Stępnik, Warszawska Szkoła Reklamy, ul. Dembowskiego 4/82, 02-784 Warszawa, andrzejstepnik@wp.pl