

Wrocław, 12. stycznia 2006

Wprowadzenie do Systemów Ekspertowych i Sztucznej Inteligencji

Politechnika Wroclawska
Wydział Informatyki i Zarządzania
IV rok studiów

Inteligentne metody w szachach (seminarium)

Autor dokumentu: **STAWARZ Pawel**
Indeks: **125939**
Grupa: **czwartek np. / 11.15**
Termin seminarium: **17. listopada 2005**

Prowadząca: **prof. Halina Kwaśnicka**

Spis treści

ABSTRAKT	3
HISTORIA AUTOMATÓW SZACHOWYCH	4
PIERWSZA MASZYNA SZACHOWA	4
PIERWSZY ELEKTRONICZNY MECHANIZM SZACHOWY	5
„INTELIENCJA” MASZYNY	6
<i>Ogólny test Turinga</i>	<i>6</i>
<i>Szachowy test Turinga</i>	<i>6</i>
„PAPIEROWA” MASZYNA TURINGA	7
STRATEGIE CLAUDA SHANNONA	7
<i>Drzewo gry</i>	<i>7</i>
<i>Alpha-Beta</i>	<i>8</i>
CIEKAWOSTKI LAT 60. I 70.	8
MASZYNA „BELLE”	9
MASZYNA „HiTECH” ORAZ „DEEP THOUGHT”	9
CIEKAWOSTKI PRZEŁOMU XX I XXI WIEKU	10
DEEP BLUE	11
„INTELIENCJA” KOMPUTERÓW SZACHOWYCH.....	11
SZTUCZNA INTELIGENCJA W PRAKTYCE	13
ALGORYTM GENETYCZNY	13
<i>Ogólna charakterystyka</i>	<i>13</i>
<i>Schemat działania.....</i>	<i>14</i>
WYKORZYSTANIE ALGORYTMU GENETYCZNEGO	15
<i>Definicja problemu</i>	<i>15</i>
<i>Definicja problemu w kontekście algorytmu genetycznego.....</i>	<i>17</i>
<i>Przeprowadzenie eksperymentów</i>	<i>19</i>
<i>Krótkie podsumowanie przeprowadzonych eksperymentów.....</i>	<i>21</i>
LITERATURA	22

Abstrakt

Szachy już od początku swego istnienia, były uważane za wyznacznik ludzkiej inteligencji. Była to królewska gra przeznaczona tylko i wyłącznie dla osób „inteligentnych” o otwartym umyśle i elastycznym myśleniu. Z biegiem czasu, szachy stały się jednak na tyle popularne, że przyjemność płynącą z samej gry mógł zasmakować każdy – niezależnie od wieku czy hierarchii społecznej. Nie zmieniło to jednak faktu, iż opanowanie wszystkich złożonych reguł taktycznych oraz strategicznych tej gry na poziomie arcymistrzowskim, wymagało nie lada intelektu potrafiącego ogarnąć niemalże „nieskończoność”.

Naturalnym więc, w dobie automatyzacji wielu czynności, stało się przeświadczenie o tym, że automat dobrze grający w szachy wykazałby się intelektem porównywalnym z ludzkim.

Pierwsza część niniejszej pracy dotyczy historii automatyzacji gry w szachy, której celem było tchnienie „życia” i „inteligencji” martwym maszynom. W tej części uwaga zostanie skupiona na ogólnym zarysowaniu tendencji stosowanych rozwiązań na przestrzeni kilkuset ostatnich lat, idących w parze z rozwojem techniki. Otworzy ona zasadnicze pytania: „czy maszyna w ogóle potrafi myśleć?” (test Turinga) i „czy rzeczywiście w dzisiejszych czasach maszyny przewyższyły możliwości intelektualne ludzi?”.

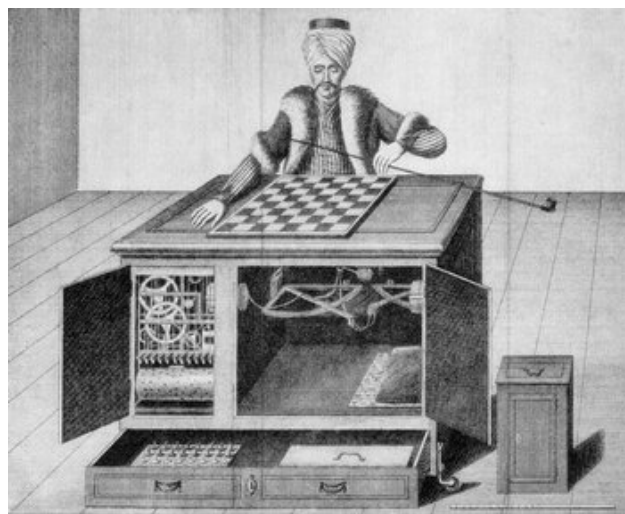
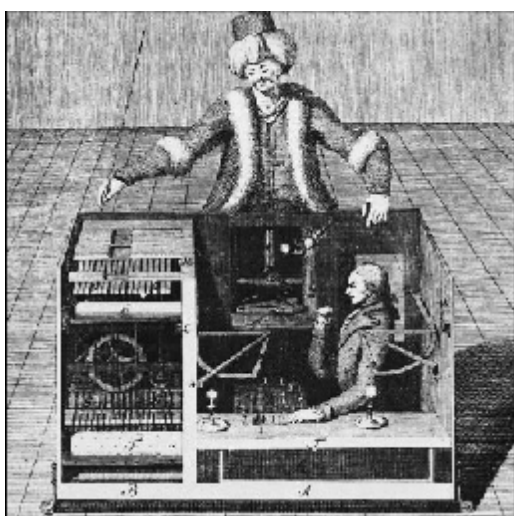
Druga część pracy będzie podsumowaniem przedstawionej historii, w którym znajdzie się m.in. subiektywna ocena autora dotycząca rozwoju programów szachowych. Przedstawienie tematu z tej perspektywy powinno pobudzić do własnych przemyśleń.

W ostatniej części opracowania uwaga zostanie skupiona na nieco głębszej analizie przykładowego zastosowania sztucznej inteligencji w programowaniu szachów. Analizowanym przykładem będzie praca magisterska Tomasza Michniewskiego (absolwenta Wydziału Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego) napisana w 1995 roku pod kierunkiem dr Jerzego Cytowskiego. Przykład zobrazuje praktyczne wykorzystanie algorytmu genetycznego do generowania optymalnej strategii gry. Warto jednak zwrócić uwagę na to, że przykład ten jest typowo dydaktyczny, tzn. nie znajduje on prawdziwego zastosowania w generowaniu współcześnie silnie grających programów szachowych. Natomiast nie oznacza to wcale, że przykład ten jest całkowicie bezsensu, a wręcz przeciwnie przedstawia jedno z klasycznych podejść tworzenia gier w ogóle.

Historia automatów szachowych

Pierwsza maszyna szachowa

Pierwsze próby stworzenia „automatycznego” szachisty sięgają już XVIII wieku, kiedy to w 1769 roku węgierski Baron Wolfgang von Kempelen zbudował dla rozrywki austriackiej królowej Marii Teresy maszynę świetnie grającą w szachy. Stała się ona przy tym najsłynniejszym automatem w historii. Było to mechaniczne urządzenie skonstruowane w kształcie Turka siedzącego przy szafie, na której stała szachownica. Tajemnica niewiarygodnie silnej gry tej maszyny kryła się w środku... w którym mógł się ukryć prawdziwy szachista. Szachista siedzący wewnątrz tego urządzenia miał do dyspozycji własne szachy, na których mógł analizować przebieg partii, delikatne światelko umożliwiające korzystanie z szachownicy, oraz różnego rodzaju korbki i uchwyty, dzięki którym mógł panować nad ruchami wykonywanymi przez gumowego Turka. Dzięki dobrze przemyślanej konstrukcji, szachista siedzący w środku widział nad swoją głową aktualny stan szachownicy leżącej na szafie. Oczywiście jak widać pierwsze próby stworzenia maszyny grającej w szachy były najzwyczajszym oszustwem.



Jako ciekawostkę warto dodać, że obecnie Heinz Nixdorf Museum w Paderbornie (Niemcy) wyprodukowało identycznie działającą replikę dawnego Turka.

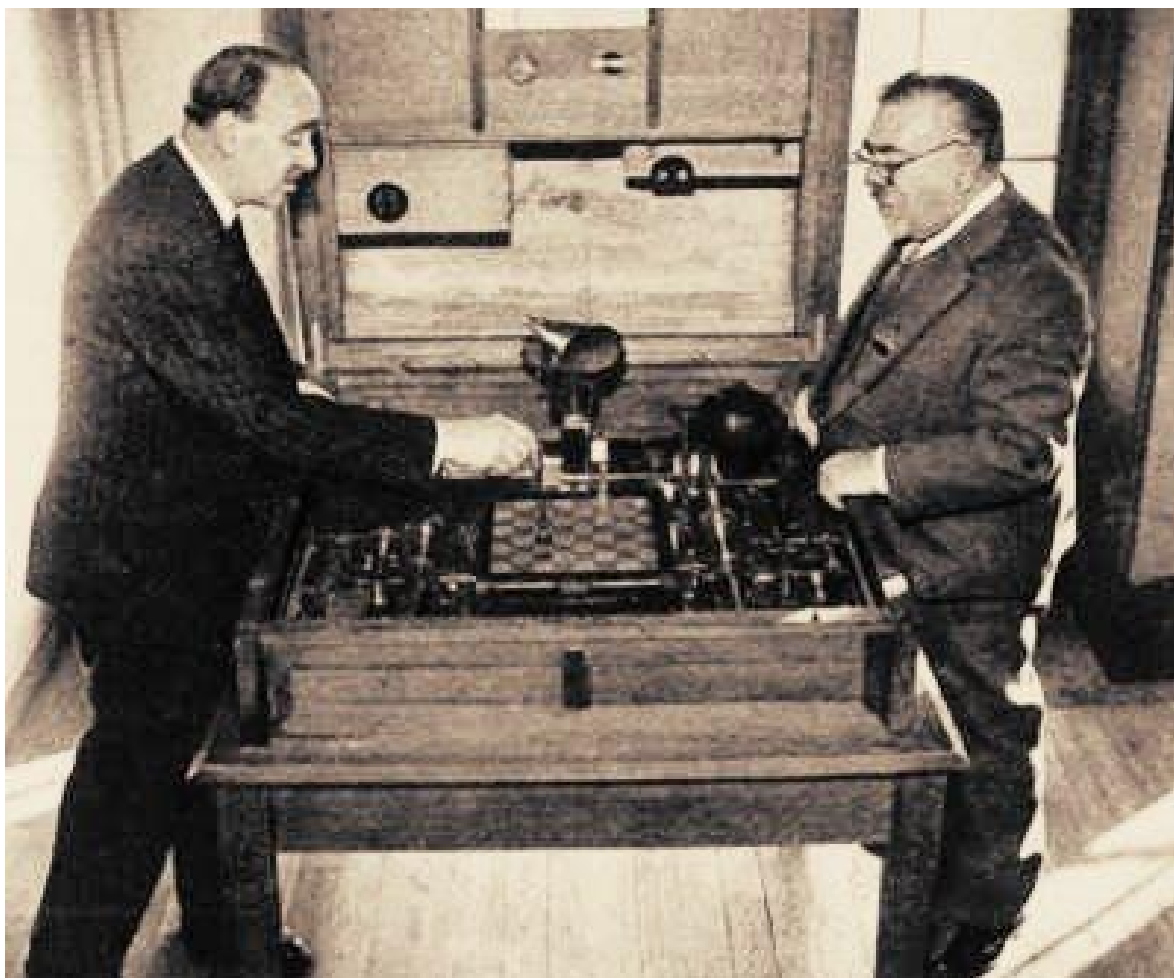


Pierwszy elektroniczny mechanizm szachowy

Pierwsze próby stworzenia elektronicznego szachisty pochodzą już z końca XIX wieku. Hiszpański inżynier, matematyk oraz wynalazca Leonardo Torres y Quevedo (1852-1936), uważany za jednego z twórców podstaw automatyki, już od 1890 roku konstruował pierwszy elektroniczny mechanizm wielkości szafy, nazwany przez niego 'El Ajedrecista' (z hiszpańskiego 'szachista'), którego celem było rozgrywanie końcówek szachowych (król+wieża przeciwko królowi). W 1912 roku Torres zakończył budowę pierwszej wersji tej maszyny. Zgodnie z założeniem, niezależnie od pozycji początkowej, maszyna potrafiła zamatować „gołego” króla przy wykorzystaniu króla i wieży. Chociaż sam manewr matowania nie był właściwie optymalny, tzn. liczba ruchów potrzebna na osiągnięcie celu niekoniecznie była minimalna, ale sam fakt, że maszyna potrafiła całkowicie samodzielnie (automatycznie) dobrnąć do mata budziło już wielkie emocje. W dzisiejszych czasach, problem, który rozwiązywała maszyna Torres'a, wydaje się być trywialny, jednak na początku XX wieku to było wielkie wyzwanie. Zapoczątkowanie przez Torres'a elektronicznej realizacji gry w szachy jak i inne prace (wynalazki), do dzisiaj uważane są za prekursorckie w dziedzinie sztucznej inteligencji i współczesnych obliczeń. Nad kolejnymi udoskonaleniami mechanizmu pracował jeszcze wraz z synem Gonzalesem do 1920 roku. Obie, wciąż działające, wersje maszyny znajdują się w muzeum Politechniki Madryckiej.



Leonardo Torres y Quevedo



G

Gonzales Torres y Quevedo przedstawia niesamowitą maszynę Norbertowi Weinerowi

„Inteligencja” maszyny

Ogólny test Turinga

Wraz z rozwojem techniki (głównie za sprawą rozwoju elektroniki) zaczęto się zastanawiać nad „inteligentnymi” możliwościami maszyn. Zaczęto zadawać sobie pytanie „czy maszyna jest inteligenta?”. Nawet w dzisiejszych czasach odpowiedź na to pytanie nie jest łatwa. Jednak już w połowie XX wieku, świetny matematyk, pionier informatyki, Alan Turing w odpowiedzi na pytanie „czy maszyna potrafi myśleć?” zaproponował swój sławny test: jeśli podczas rozmowy z kilkoma „osobami” w języku naturalnym, człowiek nie będzie w stanie wiarygodnie określić, która „osoba” jest maszyną, to wówczas powiemy, że maszyna przeszła test. Jednak do tej pory istnieją spory o to, czy test Turinga rzeczywiście we właściwy sposób definiuje inteligencję maszyn, ponieważ:

- § maszyna, która przejdzie test Turinga może być w stanie symulować ludzkie zachowanie konwersacyjne poprzez zastosowanie sprytnie wymyślonych reguł - czy to już jest „inteligencja”?
- § maszyna może być „inteligentna” bez ludzkiej umiejętności „gawędzenia”;
- § wielu ludzi mających przejść ten sam test co maszyna, mogłoby sobie z nim nie poradzić, tzn. nie uwiarygodnili by swojej tożsamości;



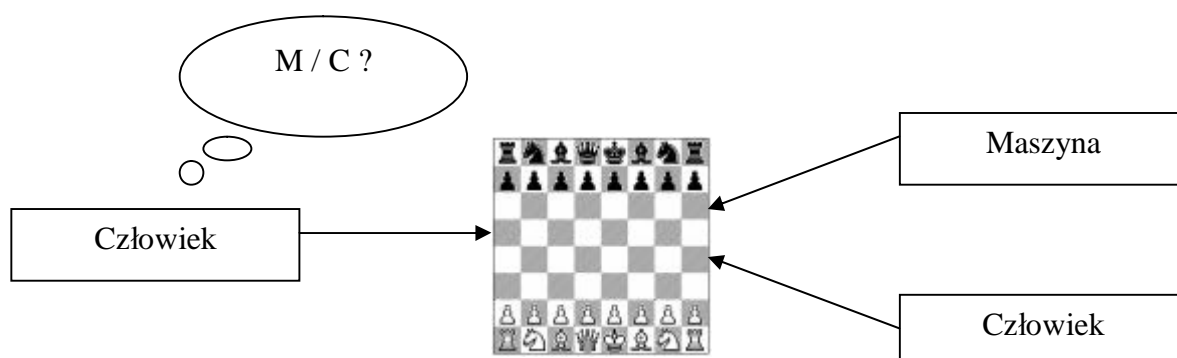
Alan Turing

Przy okazji zdefiniowania ogólnego testu, Turing przewidział, że w końcu maszyny będą w stanie przejść cały test. W szczególności wyliczył, że do roku 2000, maszyny z pamięcią ok. 120MB będą w stanie oszukać około 30% ludzi testujących ich w pięciominutowym teście. Turing zapowiedział również, że od pewnego momentu, fakt „myślącej maszyny” już nie będzie nikogo dziwił, a z kolei maszynowe uczenie będzie jedną z najistotniejszych części budowanych maszyn.

Natomiast abstrahując od tego czy test jest właściwie zdefiniowany czy nie, sam fakt uczynienia maszyny na tyle „mądrej”, aby mogła przeprowadzić „ludzką” rozmowę do dzisiaj jest wielkim wyzwaniem. Warto przy okazji ogólnego testu Turinga wspomnieć, że począwszy od 1990 roku rozgrywane są co roku zawody o nagrodę Loebnera, które wyłaniają bota najlepiej naśladowującego mowę człowieka. Celem zawodów jest niezaprzeczalne przejście testu Turinga. Osoba, której bot jako pierwszy przejdzie ogólny test Turinga wygra 100 tysięcy dolarów, a nagroda Loebnera zostanie rozwiązana. Póki co nikt jeszcze nie przeszedł całego testu Turinga, a osoby wygrywające konkurs (tzn. osoby, których boty najlepiej radzą sobie z testem) otrzymują nagrodę w wysokości 2-ch tysięcy dolarów.

Szachowy test Turinga

Aby nieco uprościć zadanie testowanej maszyny, stosuje się często ograniczone testy Turinga. Jednym z takich testów jest szachowy test Turinga, który opisuje następującą sytuację: osoba grająca w szachy z pewnym przeciwnikiem, na podstawie wykonywanych ruchów musi ustalić czy gra z człowiekiem czy też z maszyną.



Zmodyfikowana wersja tego testu zakłada, że osoba nie gra bezpośrednio w szachy, tylko otrzymuje zapis pewnej partii, z której musi określić kim byli przeciwnicy: czy grała dwójka ludzi, czy człowiek z maszyną, czy może dwójka maszyn. Celem tej modyfikacji było wzmocnienie uproszczonego testu przez dodatkową trudność, ponieważ grając w szachy bezpośrednio z przeciwnikiem, którego „tożsamość” należy ustalić, można specjalnie wykonywać pewne bezsensowne ruchy, czy też specjalnie się podkładać, aby wiarygodnie odpowiedzieć na pytanie. W tak zmodyfikowanym szachowym teście Turinga, podobnie jak w oryginalnym ogólnym teście Turinga, im „mądrzejsza” będzie maszyna, tym trudniej będzie ustalić „tożsamość” przeciwników. Pod koniec XX wieku w nieformalnie przeprowadzonym eksperymencie, bazującym na zmodyfikowanym szachowym teście Turinga, ówczesny szachowy Mistrz Świata Garry Kasparov, na podstawie przedstawionych zapisów różnych partii nie był w stanie wiarygodnie określać „tożsamości” szachistów. Zatem zmodyfikowany szachowy test Turinga został już w pewnym stopniu zaliczony. Warto jednak zastanowić się na czym tak naprawdę opiera się ta „sztuczna inteligencja” programów szachowych potrafiących współcześnie grać na bardzo wysokim poziomie.

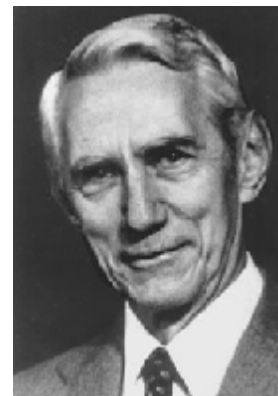
„Papierowa” maszyna Turinga

Właściwie zanim zostały wynalezione pierwsze programowalne komputery, Alan Turing zdążył już zaproponować swój pierwszy program szachowy, który został zrealizowany za pomocą maszyny stanów. Chociaż, pomimo wielkiego talentu matematycznego, Turing wcale nie był dobrym szachistą, to jednak ośmielił się napisać pewien ciąg instrukcji umożliwiający maszynie grę w szachy. W celu przetestowania swojego „papierowego” wynalazku w 1952 roku w Manchesterze rozegrał partię szachową ze swoim kolegą (Alick Glennie). Na każdy ruch Turing potrzebował średnio pół godziny, aby odpowiednio wykonać rozpisany przez siebie ciąg instrukcji. Niestety w ostatecznym rozrachunku maszyna Turinga przegrała z Alickiem, ale najważniejszym było postawienie kolejnego kroku w stronę zautomatyzowania gry w szachy.

Strategie Clauda Shannona

Drzewo gry

Niemalże w tym samym czasie co Turing, kolejny geniusz matematyki, Claude Shannon zastanawiał się nad możliwością uczenia maszyn gry w szachy. W swojej pracy „Programming the computer for playing chess” rozpatruje szachy jako grę skończoną, dla której można wygenerować drzewo, którego węzłami są możliwe do uzyskania pozycje figur na szachownicy po wykonaniu odpowiednich posunięć, w związku z czym liściom tego drzewa, a więc pozycjom końcowym, można przyporządkować ostateczny wynik gry. Co prawda istnieją takie sytuacje na szachownicy, że w skończonej liczbie ruchów nie da się doprowadzić do końca gry (tzn. sytuacji, w której nie można wykonać już żadnego ruchu), ale dodatkowo zakłada się, że jeżeli na przykład w ciągu 50 ruchów od ostatniego bicia, żadna ze stron nie doprowadzi do wygranej, to przyjmuje się wynik nierozstrzygnięty, czyli remis. Oczywiście cały problem ogarnięcia szachów tkwi w niemożności określenia całego drzewa gry (na szczęście, bo straciłyby one swój niepowtarzalny urok). Jeżeli na przykład przyjmiemy, że każda ze stron wykonująca ruch ma do dyspozycji średnio 30 możliwości, to dla przeanalizowania jednego ruchu mamy już $30 \times 30 = 900$ możliwości, a przy średniej liczbie 50 ruchów do rozstrzygnięcia gry liczba ta wzrasta do oktylionu seksdecyliardów (ok. 10^{150}), co jest już liczbą niewyobrażalną. Dlatego też przeszukiwanie drzewa gry ograniczane jest do ustalonej jego głębokości i dla tak ustalonego drzewa Shannon rozróżnił dwie strategie:



Claude Shannon

- § strategia A – oceniająca wszystkie węzły drzewa;
- § strategia B – oceniająca wszystkie węzły obciętego z pewnych gałęzi drzewa;

Obie strategie, w nieco zmodyfikowanej formie, znajdują swoje odbicia we współczesnych programach szachowych, gdzie strategię A utożsamia się z „brutalnym przeszukiwaniem” oraz strategię B z „przeszukiwaniem selektywnym”.

Pomimo znacznego ograniczenia ocenianych możliwości, tzn. podcięcia drzewa do pewnej głębokości, nadal przejście węzłów tego drzewa przy wykorzystaniu dowolnej strategii wiązało się z potężnym procesem obliczeniowym, którego ówczesne maszyny nie były w stanie podjąć w rozsądnym czasie. Wczesne maszyny szachowe potrafiły oceniać około 500 pozycji na sekundę, co daje około 90 tysięcy pozycji na 3 minuty (np. w turniejach, w których obowiązywała zasada trzech minut na ruch). Okazuje się, że przy takiej mocy obliczeniowej maszyna potrafiła przeanalizować co najwyżej półtora ruchu. Oczywiście analiza półtora ruchu

naprzód nie umożliwiała maszynie przeprowadzenia gry na wysokim poziomie. Zatem możliwość zawiadnięcia szachami nadal stała pod wielkim znakiem zapytania.

Alpha-Beta

Na szczęście już w 1958 roku trzech naukowców (Newell, Shaw, Simon) z uniwersytetu w Pittsburghu odkryło bardzo ciekawą zależność, która pozwalała odcinać duże gałęzie przeszukiwanego drzewa bez wpływu na wynik końcowy. Zaproponowany algorytm nazwali cięciami Alpha-Beta. Było to czysto matematyczne podejście i działało bez wykorzystywania dodatkowej wiedzy szachowej. Algorytm opierał się w przybliżeniu na następującym działaniu: po przeanalizowaniu pierwszego ruchu i jego ocenieniu maszyna przechodzi do analizy drugiego i jeśli do ustalonego momentu nie znajdzie lepszej oceny (względem ruchu pierwszego), to rezygnuje z dalszego przeszukiwania tego ruchu.

Zastosowanie tego algorytmu w praktyce pozwoliło maszynom zwiększyć głębokość przeszukiwań drzewa gry nawet do 3.5 ruchu w rozsądnym czasie, osiągając przy tym już dość silny poziom gry. Jednak wykładniczy wzrost możliwości pojawiających się wraz z głębszym przeszukiwaniem drzewa, po raz kolejny umożliwił osiągnięcie tylko pewnej rozsądnej granicy, po przekroczeniu której maszyny nie były w stanie sobie poradzić.

Jako ciekawostkę warto dodać, że w tym samym roku program szachowy NSS wygrał w szachy z osobą, która pierwszy raz w życiu w ciągu godziny uczyła się grać w szachy. Oczywiście nie jest to wielkie osiągnięcie, ale jest na tyle ciekawe, że na pewno na stałe wpisało się w karty historii. Jak widać wciąż trwają badania i przeprowadzane są różnego rodzaju eksperymenty.

Ciekawostki lat 60. i 70.

W 1962 roku Alan Kotok, przy współpracy z John`em McCarthy`m napisał program szachowy, który jako pierwszy potrafił grać w szachy regularnie na bardzo dobrym poziomie. Program był uruchamiany na komputerze IBM 7090, a potrafił przeszukiwać 1100 pozycji na sekundę.

W 1965 roku Instytut Fizyki Teoretycznej i Eksperymentalnej w Moskwie pod przewodnictwem Georgi`ego Adelson`a-Velskiy`ego stworzył program szachowy, który na przełomie lat 1966/1967 rozegrał 9-miesięczny mecz korespondencyjny z programem szachowym Kotok`a oraz McCarthy`ego, który wygrał wynikiem 3:1.

W 1966 roku MAC HACK VI (DEC PDP-6), napisany przez Richard`a Greenblatt`a, stał się pierwszym programem szachowym, który wziął udział w turnieju z prawdziwymi zawodnikami. Były to amatorskie mistrzostwa Massachussets, w którym zdobył 0.5 punktu z 5 partii. Jednakże już wiosną 1967 roku w Mistrzostwach Stanu Massachussets, MAC HACK VI stał się pierwszym programem szachowym, który pokonał zawodnika z rankingiem 1510 USCF. W 1967 roku MAC HACK VI wziął udział łącznie w czterech turniejach, wygrywając 3 partie, remisując – 3 i przegrywając – 12. MAC HACK VI był również pierwszym programem szachowym, w którym zaprogramowano szachową księgę debiutów.

W 1968 roku mistrz międzynarodowy David Levy założył się o 3000 dolarów, że żaden komputer szachowy nie pokona go w przeciągu 10 lat. Zakład został zawarty z badaczem sztucznej inteligencji John`em McCarthy`m w Machine Intelligence Workshop na Uniwersytecie Edynburskim. Jak się okazało David Levy wygrał swój zakład.

W 1970 roku w Nowym Jorku zorganizowano pierwsze Północnoamerykańskie Mistrzostwa ACM (*Association for Computing Machinery*) Komputerów Szachowych, w których zwyciężył program CHESS 3.0 (CDC 6400) napisany przez trójkę naukowców Slate, Atkin oraz Gorlen z Uniwersytetu Northwestern. W turnieju wystąpiły również programy DALY CP, J Brit, COKO III, SCHACH oraz Marsland CP.

W latach 1971-1973 kolejne wersje programu szachowego CHESS (3.5, 3.6, 4.0) zwyciężały kolejne Północnoamerykańskie Mistrzostwa ACM, odbywające się kolejno w Chicago, Bostonie oraz w Atlancie.

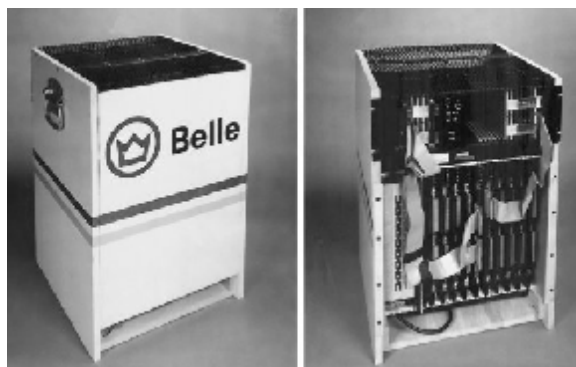
W 1974 roku w Sztokholmie przeprowadzono pierwsze Mistrzostwa Świata Komputerów Szachowych, w których zwyciężył program KAISSA stworzony przez Donskoy`a oraz Arlazarov`a w Moskwie w Instytucie Nauki Sterowania. Drugie miejsce przypadło programowi CHESS 4.0.

Do końca lat osiemdziesiątych prym najlepszych programów szachowych wiedzie wciąż udoskonalany program szachowy CHESS. W 1977 w Toronto program CHESS 4.6 wygrywa drugie Mistrzostwa Świata Komputerów Szachowych. Następnie w 1978 roku, program CHESS 4.7 po raz pierwszy remisuje z szachowym mistrzem międzynarodowym. Od tego momentu optymiści przewidują, że w ciągu 10 lat komputer zostanie mistrzem świata.

W 1980 roku Edward Fredkin organizuje Nagrodę Fredkin`a w wysokości 100 tysięcy dolarów dla pierwszego komputera szachowego, który pokona panującego mistrza świata.

Maszyna „Belle”

Wraz z rozwojem podejścia algorytmicznego w szachach i niemożnością przekroczenia pewnej bariery szybkości w komputerach programowalnych, wciąż zastanawiano się nad rozwiązaniami typowo sprzętowymi. Starano się skonstruować maszynę grającą tylko i wyłącznie w szachy. Na przełomie lat 70. i 80. w laboratoriach Belle`a naukowiec Ken Thompson postanowił zbudować taką właśnie maszynę, która kosztowała około 20 tysięcy dolarów. Maszyna została nazwana „Belle” i potrafiła przeszukiwać 180 tysięcy pozycji na sekundę, co w porównaniu z ówczesnym najszybszym superkomputerem, który potrafił wykonać tę samą operację z szybkością 5 tysięcy pozycji na sekundę, było ponad 30 razy szybszym rozwiązaniem. Rozwiązanie sprzętowe pozwoliło osiągnąć głębokość 4-5 ruchów, dzięki czemu osiągnięto już poziom niemalże mistrzowski. W roku 1980 maszyna „Belle” wygrywa mistrzostwa świata komputerów szachowych i do roku 1983 wiezie prym najlepszego komputera na świecie, do momentu aż pojawia się o wiele lepsze rozwiązanie Cray X-MP kosztujące kilka tysięcy razy więcej dolarów niż „Belle” i które w 1984 wygrywa czwarte Mistrzostwa Świata Komputerów Szachowych.



Maszyna „HiTech” oraz „Deep Thought”

W połowie lat osiemdziesiątych profesor Hans Berliner, będący korespondencyjnym szachowym mistrzem świata, wraz z jego byłym studentem Carl`em Ebeling`iem, tworzą sprzętowe rozwiązanie generatora ruchów szachowych (HiTech). Generator ten opierał się na 64 równoległe działających chipach. W 1985 HiTech, uzyskując ranking 2530 stał się pierwszym komputerem szachowym, który w ogóle przekroczył ranking szachowy 2400. W 1986 roku w piątym Mistrzostwach Świata Komputerów Szachowych, HiTech przegrywając jedyną partię z Cray`em zdobywa za nim dopiero drugie miejsce.



Hans Berliner

Zaraz po wynalazku Berliner`a, jego studenci Feng-hsiung Hsu, Murray Campbell i inni pracowali nad własną maszyną ChipTest, którą w ostatecznej wersji nazwali Deep Thought. Maszyna kosztowała jedynie 5 tysięcy dolarów i była w stanie przeszukiwać 500 tysięcy pozycji na sekundę. Hsu oraz Campbell zaraz po rozstaniu ze swoim nauczycielem (Berliner`em) dołączyli do zespołu IBM, przyczyniając się później do stworzenia potężnej maszyny szachowej Deep Blue.



Feng-Hsiung Hsu

W 1988 roku w Otwartych Szachowych Mistrzostwach Stanów Zjednoczonych, Deep Thought dzieli pierwsze miejsce wraz z arcymistrzem Tony Miles`em, uzyskując jednocześnie turniejowy ranking 2745. W tym samym roku HiTech Berliner`a wygrywa Szachowe Mistrzostwa Stanu Pennsylvania pokonując mistrza międzynarodowego Ed`a Formanek`a oraz wygrywa zorganizowany mecz z arcymistrzem Arnold`em Denker`em i staje się pierwszym komputerem szachowym, który osiągnął ranking arcymistrzowski.



Murray Campbell

Z kolei w 1989 roku w Edmonton, Deep Thought wygrywa szóste Mistrzostwa Świata Komputerów Szachowych, a następnie wygrywa zorganizowany z arcymistrzem Robert`em Byrne`m mecz. Deep Thought będąc wciąż pod opieką swoich twórców potrafi już analizować 2 miliony pozycji na sekundę. W rozegranym w marcu 1989 roku meczu, mistrz świata Garry Kasparov pokonuje jednak maszynę Deep Thought w dwóch grach. Natomiast w zorganizowanym meczu z mistrzem międzynarodowym David`em Levy`m Deep Thought pokonuje go, wygrywając 4 partie.

Ciekawostki przełomu XX i XXI wieku

W 1990 roku były mistrz świata Anatoly Karpov przegrywa w przeprowadzonej w Monachium symultanie z programem MEPHISTO. MEPHISTO pokonuje również arcymistrzów Robert'a Hubener'a oraz David'a Bronstein'a. Wygrywa również Mistrzostwa Niemiec w Szachach Szybkich i zdobywa tytuł mistrza międzynarodowego zdobywając 7 punktów z 11 partii w silnym Openie w Dortmundzie.

W 1992 roku w Kolonii w Niemczech, mistrz świata Garry Kasparov rozgrywa mecz jedenastu pięciominutowych partii z programem szachowym Fritz 2, który kończy się zwycięstwem Kasparov'a 6.5:4.5. Po raz pierwszy program szachowy pokonuje mistrza świata w szachach szybkich.

W 1994 roku w turnieju szachów szybkich w Monachium, program szachowy Fritz 3 pokonuje świetnych arcymistrzów szachowych: Kasparov'a, Anand'a, Short'a, Gelfand'a oraz Kramnik'a. Z kolei po raz pierwszy w historii arcymistrz Robert Huebner w tym samym turnieju odmawia gry z Fritz'em i przegrywa walkowerem. Po raz kolejny rozgrywany jest mecz, tym razem sześciu pięciominutowych partii, pomiędzy Garry'm Kasparov'em a Fritz'em, i po raz kolejny zwycięża mistrz świata wynikiem 5:1.

W 1994 roku w turnieju Intel Speed Chess Grand Prix w Londynie, Kasparov w dwudziestopięciominutowej partii zostaje wyeliminowany z dalszego współzawodnictwa przez program Chess Genius 2.95.

W 1995 roku w Nowym Jorku organizowany jest pojedynek Harvard Cup Human Versus Computer. Arcymistrzowie wygrywają z komputerami szachowymi wynikiem 23.5:12.5. Najlepszą maszyną okazał się Virtual Chess (I-Motion Interactive) zdobywając 3.5 punktu z 6 partii.

W lutym 1996 roku w Filadelfii, IBM ze swoim najnowszym rozwiązaniem Deep Blue wyzywa Garry Kasparov'a na pojedynek w meczu rozgrywanym na poważnych zasadach turniejowych. Pierwszą partię zwycięża Deep Blue, stając się tym samym pierwszym komputerem w historii, który pokonuje panującego mistrza świata. Cały pojedynek kończy się jednak zwycięstwem Kasparov'a 4:2. Dla porównania możliwości obliczeniowych komputera i człowieka podaje się, że Deep Blue był w stanie przeanalizować 50 miliardów pozycji w ciągu każdych trzech minut, natomiast Kasparov w tym samym czasie był w stanie przeanalizować jedynie 10 pozycji.

W kwietniu 1996 roku w Hadze, zorganizowano AEGON Computer Chess Tournament (Mankind vs Machine), w którym wystąpiło 50 mistrzów, mistrzów międzynarodowych oraz arcymistrzów, przeciwko 50 programom szachowym (w większości korzystających z maszyn HP Pentium 166MHz z 16MB RAM). Najlepszym komputerem okazał się QUEST, który zdobył 4.5 punktu z 6 partii uzyskując jednocześnie z gry ranking 2652. Maszyny zwyciężyły nad ludzkością wynikiem 162.5:137.5.

W maju 1997 roku w Nowym Jorku dochodzi do rewanżowego spotkania pomiędzy mistrzem świata Garry'm Kasparov'em a nową wersją Deep Blue firmy IBM, w którym po raz pierwszy w historii wygrywa maszyna. Z tego też względu ustanowiona w 1980 roku nagroda Fredkina (w wysokości 100 tysięcy dolarów) trafiła w końcu do twórców komputera Deep Blue: Feng Hsu, Murray Campbell, Joseph Hoane. Nowa wersja Deep Blue potrafiła analizować już 200 milionów ruchów na sekundę.

W październiku 2002 roku w Bahrain, obecny mistrz świata Vladimir Kramnik remisuje mecz z komputerem Deep Fritz wynikiem 4:4. Chociaż Kramnik wygrywał już 3:1, to ostatecznie nie wytrzymuje presji i trwoni zdobytą przez siebie przewagę.

Na początku 2003 roku w Nowym Jorku Garry Kasparov, rozgrywa mecz z kolejną wersją komputera Deep Junior 7. Mecz kończy się remisem 3:3. Oprogramowywanie Deep Junior'a zajęło 10 lat programistom z Tel Avivu – Amir Ban oraz Shay Bushinsky. Deep Junior potrafił przeanalizować 3 miliony ruchów na sekundę, wgłębiając się w analizę 15 ruchów naprzód.

W listopadzie 2003 roku w Nowym Jorku, Kasparov rozgrywa mecz z X3dFritz w warunkach trójwymiarowej wirtualnej rzeczywistości. Po raz pierwszy w historii stworzono takie warunki gry. Kasparov korzystając ze specjalnych okularów przedstawiających trójwymiarowy obraz szachownicy, rozgrywał każdą partię. Kasparov każdy swój ruch wypowiadał na głos, a komputer z wbudowanym systemem rozpoznawania mowy wychwytywał te informacje i wykonywał stosowny ruch na wirtualnej szachownicy. Ostatecznie Kasparov remisuje wynikiem 2:2.

Zestawienie najlepszych komputerów szachowych (wraz z przybliżoną siłą gry) na 2003 rok wygląda następująco:

- § Shredder 7.04 (2810)
- § Shredder 7.0 (2770)
- § Fritz 8.0 (2762)

- § Deep Fritz 7.0 (2761)
- § Fritz 7.0 (2742)
- § Shredder 6.0 (2724)
- § Chess Tiger 15.0 (2720)

Dla porównania warto pokazać ranking (siłę gry) pięciu najlepszych szachistów na świecie na rok 2006:

- § Kasparov Garry (2812)
- § Topalov Veselin (2801)
- § Anand Viswanathan (2792)
- § Svidler Peter (2765)
- § Aronian Levon (2752)

Deep Blue

W poprzednim podrozdziale wspomniano już krótko o historii związanej z siłą gry komputera Deep Blue stworzonego przez firmę IBM. Z punktu widzenia historycznego jest to maszyna, która zakończyła pewną erę. Wreszcie doprowadzono do rozwiązania, które zwyciężyło z prawdziwym szachowym Mistrzem Świata Garrym Kasparowem. Dla przypomnienia pierwszy wygrany przez Garry Kasparov'a pojedynek został rozegrany w 1996 roku w Filadelfii, natomiast drugi przegrany – w 1997 roku w Nowym Jorku.



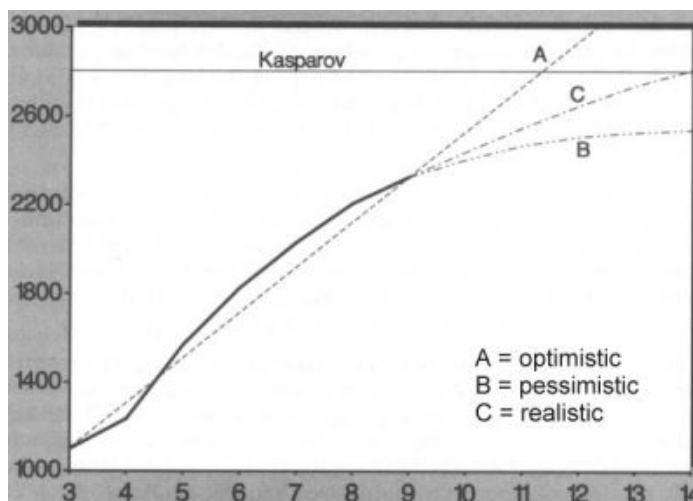
Garry Kasparov

Deep Blue jest komputerem szachowym skonstruowanym tylko i wyłącznie z myślą o grze w szachy. Składa się on z ponad 200 specjalnych chipów przeznaczonych do wykonywania szybkich obliczeń, z których każdy umożliwia przeanalizowanie od dwóch do trzech milionów pozycji na sekundę. Wykorzystanie ponad 200 chipów pozwoliło osiągnąć ekstremalną liczbę 200 milionów analizowanych ruchów na sekundę.

„Inteligencja” komputerów szachowych

Na przestrzeni ostatnich kilkudziesięciu lat można zaobserwować, iż główną tendencją gwarantującą wysoki poziom gry komputerów szachowych było uzyskiwanie jak największej szybkości przeszukiwania drzewa gry w ustalonej na szachownicy pozycji. Za każdym razem osiągnano to głównie dzięki rozwiązaniom sprzętowym idącym w parze z rozwojem elektroniki. Oczywiście niemniejszym zainteresowaniem cieszyły się rozwiązania programowe poszukujące coraz to lepszych możliwości algorytmicznych gry w połączeniu z programowaniem niskiego poziomu, np. przy pomocy języka assemblera gwarantującego szybkie wykonywanie ustalonych instrukcji.

Ciekawe wyniki badań przedstawił w latach 80-tych Ken Thompson – twórca komputera szachowego „Belle”. W przeprowadzanych eksperymentach badał zależność głębokości przeszukiwania drzewa gry z siłą gry komputera szachowego w ten sposób grającego. Wyniki jego badań wykazały, że średnio co pół ruchu głębszego przeszukiwania drzewa, siła gry komputera szachowego wzrasta o 200 punktów rankingowych. Przy głębokości przeszukiwania 4.5 ruchów, siłę gry komputera szachowego określił na 2328 punktów rankingowych. Po wykonanych eksperymentach określił hipotetyczny dalszy wzrost siły gry:



Warto w związku z tym zadać sobie pytanie czy maszyna świetnie grająca w szachy faktycznie jest inteligenta? Zestawiając w obecnych czasach możliwości obliczeniowe komputera i człowieka oczywiste jest, że komputer w miażdżącym stylu przewyższa umiejętności zwykłego człowieka. W tym też tkwi cała moc komputera grającego w szachy, który w każdej pozycji szachowej jest w stanie przeanalizować nawet kilkanaście ruchów naprzód, gdzie dla porównania najlepsi szachiści może i są w stanie przeanalizować kilkanaście ruchów, ale tylko w kilku interesujących ich wariantach, często forsownie przebiegających. Człowiek nie posiada fizycznie takich możliwości, aby w każdej sytuacji móc przejrzeć każdy możliwy wariant do głębokości kilkunastu ruchów. W tej sytuacji jedyną bronią człowieka staje się intuicja podparta teorią, której brakuje komputerowi. To ona wciąż pozwala człowiekowi mierzyć się z komputerem jak równy z równym i może to właśnie ona jest ważnym składnikiem ludzkiej inteligencji. Gdyby ograniczyć możliwości obliczeniowe komputera do możliwości obliczeniowych najlepszego człowieka na ziemi, jestem przekonany, że komputer nie miałby w tej chwili żadnych szans. Gdyby komputer przy ograniczonych możliwościach analizowania gry potrafił zwyciężyć mecz z panującym mistrzem świata, to byłbym bardziej skłonny stwierdzić, że komputer jest „inteligentny”.

Kolejną znaczną przewagą komputera nad człowiekiem jest jego zdolność do gromadzenia ustalonej wiedzy teoretycznej. Pamięć masowa komputerów wciąż rośnie i w związku z tym nie ma problemów z jej oszczędnym wykorzystywaniem. Ważnym czynnikiem wykorzystującym możliwości pamiętania informacji jest oczywiście szybkość dostępu do przechowywanych informacji jak również szybkość wyszukania właściwej informacji w ogromnym zbiorze wszystkich informacji. Pod tym względem człowiek również ustępuje możliwościom komputerów, ponieważ ogarnięcie całej wiedzy teoretycznej związanej z pewnym zagadnieniem może być bardzo trudne, a nawet nie możliwe, ze względu na ograniczenia w szybkości przyswajania tej wiedzy jak również ze względu na czynnik zapominania pewnych szczegółów. Jeżeli chodzi o pamięć masową komputerów, to w tej chwili jest ona wykorzystywana na przykład do przechowywania bazy wiedzy gry początkowej – wszystkich otwarć gry, które ewoluowały na przestrzeni wielu lat i które doczekały się również swojego miejsca w potężnych encyklopediach debiutów szachowych. Baza takiej wiedzy dostarcza komputerowi informacji o najlepszych ruchach w początkowej fazie gry, statystyki wykonywanych ruchów pod względem wygranych, remisów, przegranych, a także siły graczy wykonujących dany ruch. Przy wykorzystaniu tej wiedzy pierwsze obliczenia komputerowe związane z samą grą pojawiać się mogą nawet po 20. ruchu.

Kolejnym ciekawym eksperymentem może pochwalić się Ken Thompson, który w latach 80-tych rozpoczął generowanie i przechowywanie wszystkich możliwych pozycji końcowych zawierających cztery oraz pięć figur szachowych. Warto zwrócić uwagę, iż na przykład typowa pięciofigurowa końcówka zawierająca króla z dwoma gońcami przeciwko królowi ze skoczkiem zawiera 121 milionów pozycji. Ken Thompson dla ogromnej liczby wszystkich możliwych cztero- oraz pięcio-figurowych końcówek wypracował jedynie słuszny sposób gry. Łatwo sobie w ten sposób wyobrazić, że komputer szachowy dysponujący taką bazą wiedzy, dochodząc w grze do cztero- bądź pięcio-figurowej końcówki, rozegra ją perfekcyjnie nie dając żadnych szans człowiekowi. Końcówki wygrane zawsze wygra, końcówki remisowe co najmniej zremisuje, a w końcówkach przegranych zawsze będzie się bronić jedynym optymalnym sposobem mogąc nawet zremisować bądź wygrać z człowiekiem, który tak wielkiej wiedzy na temat wszystkich tego typu końcówek nie posiada. Ken Thompson nie poprzestał jednak na tych badaniach i wciąż pracował nad końcówkami sześć- a nawet siedmio-figurowymi. Wyników tych badań nie znam, ale warto zauważyć, że ogarnięcie wszystkich końcówek sześćfigurowych, a tym bardziej siedmiofigurowych wymaga przejrzania biliarów pozycji.

Jednakże nie samymi szachami człowiek żyje, co oznacza, że inteligencja nie sprowadza się jedynie do umiejętności bardzo dobrej gry w szachy. Inteligencja człowieka wciąż pozwala poznawać rzeczy nie poznane, odkrywać rzeczy nie odkryte i tworzyć rzeczy niestworzone. Natomiast ograniczenie możliwości komputera do wykonywania ściśle określonego zadania nie można nazwać inteligencją. Cokolwiek by to nie było, to zawsze problem będzie sprowadzał się do pewnego ustalonego zakresu, poza który komputer nie będzie zdolny wyjść. Jeżeli udałoby się osiągnąć sytuację, w której maszyna byłaby zdolna osiągać rzeczy nieosiągalne, ale prawdziwe, na podstawie wstępnych założeń, to uznałbym, że świat maszyn jest już inteligentny. Póki co sztuczną inteligencję widzę jako kolejną dziedzinę informatyki/elektroniki opisującą tak naprawdę kolejne techniki algorytmiczne, które często dopuszczając losowość działania potrafią doprowadzać do pewnego optymalnego rozwiązania w nieprzeniknionej przestrzeni rozwiązań. Przykładem takiego algorytmu wykorzystującego losowość, wykorzystywanego w sztucznej inteligencji, jest algorytm genetyczny, który zostanie opisany w kolejnej części pracy. Nasuwa mi się tutaj od razu pytanie czy algorytm oparty na szczęściu można nazwać inteligentnym? Szybkość i trafność znalezionej optymalnej rozwiązania sprowadza się w końcu do tego czy algorytm będzie miał szczęście czy też nie. Ktoś mógłby powiedzieć: „no tak, ale przecież największe odkrycia ludzkości są często zasługą szczęścia czy też zbiegu pewnych okoliczności”. Nie zmienia to jednak faktu, że szczęście nie jest wyznacznikiem inteligencji.

Szczyście co najwyżej pomaga rozwinąć skrzydła wyobraźni i wykorzystać inteligencję do kolejnych odkryć. Szczyście nie podparte wiedzą ani zdolnością interpretacji nie wystarczy do osiągnięcia sukcesu. Często założenia przyjmowane za stuprocentowy pewnik na przestrzeni wielu wieków stają nagle pod wielkim znakiem zapytania. To co kiedyś było niepodważalne teraz staje się kwestią sporną i trzeba nie lada intelektu a nawet geniuszu, aby odkryć prawdę.

Sztuczna inteligencja w praktyce

W niniejszej części pracy zostaną omówione badania przeprowadzone w 1995 roku w pracy dyplomowej Tomasza Michniewskiego z Wydziału Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, pt. „Samouczenie programów szachowych”, pod kierunkiem dr Jerzego Cytowskiego. Samouczenie programów szachowych zostało zbadane przy wykorzystaniu algorytmu genetycznego, którego celem było wyznaczenie optymalnych współczynników zaproponowanej funkcji oceniającej. Zadaniem funkcji oceniającej jest ocena ustalonej pozycji figur na szachownicy w postaci wartości wyliczonej na podstawie tej funkcji. Funkcja oceniająca jest funkcją liniową, której argumentami są wartości zaproponowanych cech gry z punktu widzenia oceny określonej pozycji. Z każdą cechą związana jest ustalona wartość współczynnika. Jak już wcześniej wspomniałem celem algorytmu genetycznego będzie dobranie takich wartości tych współczynników, aby funkcja oceniająca najlepiej spełniała swoje zadanie.

Algorytm genetyczny

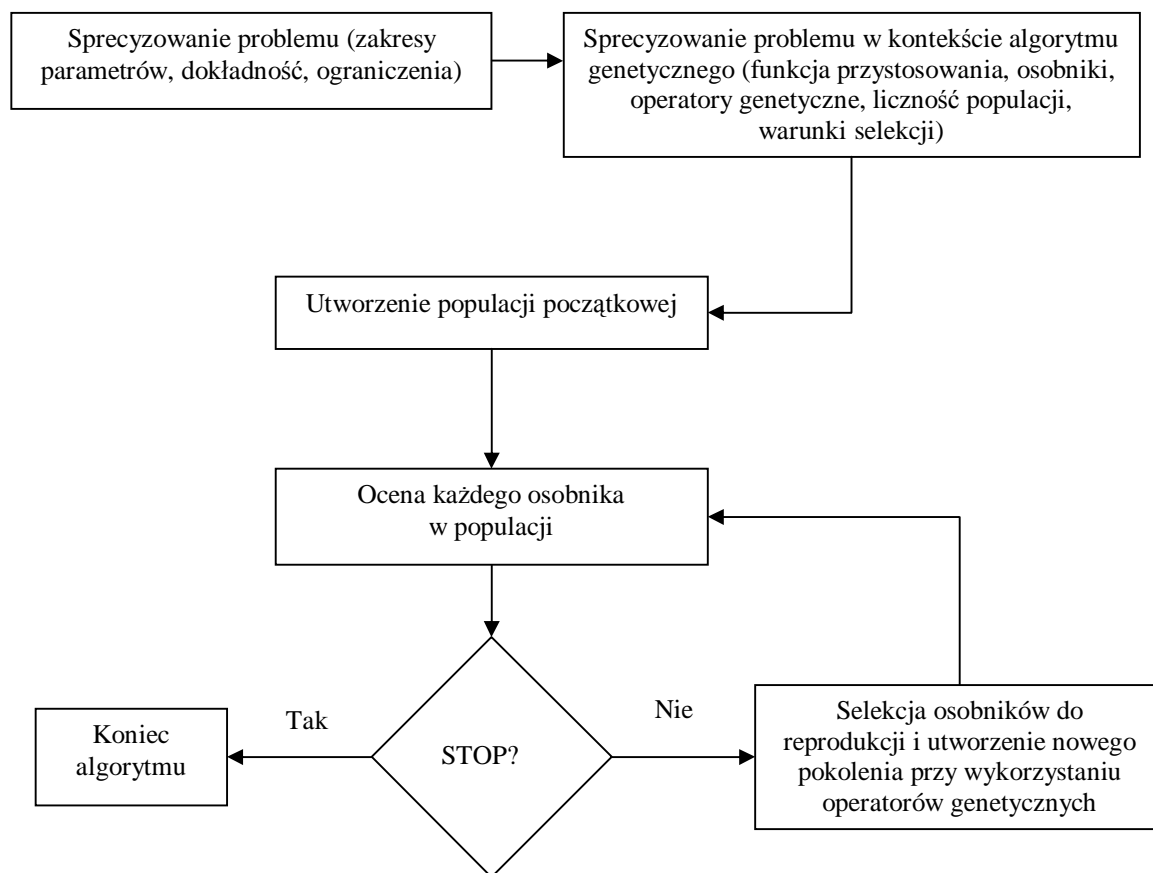
Zanim jednak przejdziemy do opisu samych badań warto na chwilę zatrzymać się przy pojęciu algorytmu genetycznego i chociaż krótko je omówić.

Ogólna charakterystyka

Algorytm genetyczny jest probabilistyczną metodą poszukiwania najlepszych rozwiązań. Nie jest to jednak zwykłe błędzenie losowe, lecz w pewien sposób ukierunkowane poszukiwanie coraz lepszych rozwiązań. Idea algorytmu genetycznego powstała w wyniku „podglądania przyrody”. Poprzez analogię do natury (doboru naturalnego) wymyślono heurystykę bazującą na ewolucji populacji. Populacja składa się z określonej liczby osobników. Natomiast każdy osobnik reprezentuje pewien punkt w przestrzeni wszystkich możliwych rozwiązań określonego zadania. Każdy osobnik posiada również swój własny kod genetyczny, który w klasycznym podejściu stosowania algorytmów genetycznych reprezentowany jest jako ciąg zer i jedynek utożsamianych z ciągiem genów. W każdej populacji można wyróżnić osobniki mniej lub bardziej przystosowane do panujących warunków. W przypadku stosowania algorytmu genetycznego, w każdej populacji można wyróżnić wartość przystosowania danego osobnika względem rozwiązywanego zadania. Osobniki najlepiej przystosowane, które potencjalnie posiadają najwartościowszy materiał genetyczny, stają się podstawą utworzenia kolejnej populacji poprzez ich reprodukcję. Czynnikiem ten sprawia, iż działanie algorytmu genetycznego pomimo swej losowości związanej z doбором osobników do reprodukcji, a także procesu samej reprodukcji jest jak najbardziej ukierunkowany poprzez promowanie najlepiej przystosowanych osobników, które potencjalnie będą mogły wytworzyć grupę jeszcze lepiej przystosowanych osobników. W algorytmach genetycznych wyróżnia się wiele technik selekcji osobników, z których każda opiera się na promowaniu osobników lepiej przystosowanych. Osobniki dobrane zgodnie z odpowiednią techniką selekcji poddawane są następnie reprodukcji przy wykorzystaniu tzw. operatorów genetycznych, tj. krzyżowanie polegające na wymianie materiału genetycznego pomiędzy dwoma osobnikami czy losowa mutacja określonego genu. Liczba operatorów genetycznych nie jest ograniczona – ważne jest tylko to, aby w wyniku ich zastosowania powstawały „poprawne” osobniki. Pod pojęciem „poprawnego” osobnika rozumiemy tutaj poprawny schemat materiału genetycznego oraz opcjonalnie poprawną postać osobnika spełniającą dodatkowe ograniczenia nałożone na rozwiązywane zadanie. Powyższa opcjonalność wynika z tego, iż w przypadku dopuszczenia powstawania osobników nie spełniających ograniczeń zadania, można wprowadzić odpowiednie mechanizmy, które będą powodować, iż wartość przystosowania takiego osobnika będzie bardzo niska. Początkowe pokolenie składa się zazwyczaj z losowych osobników, jednak dopuszcza się możliwość ręcznego ustalenia ich materiałów genetycznych. Kolejne pokolenia powstają w wyniku omówionej wcześniej selekcji oraz reprodukcji. Algorytm genetyczny kończy swoje działanie w przypadku, gdy zostanie znalezione satysfakcjonujące rozwiązanie bądź liczba pokoleń/populacji osiągnie maksymalną dopuszczalną wartość. Przy stosowaniu algorytmu genetycznego należy jednak pamiętać, iż znalezione przez niego rozwiązania niekoniecznie są najlepsze w skali całego problemu, lecz są jedynie najlepsze w skali rozwiązań do tej pory odkrytych. Algorytm genetyczny zapewnia odnalezienie ekstremum lokalnego rozwiązania, ale nie zapewnia odnalezienia ekstremum globalnego. Stosowanie tego algorytmu jest wskazane w problemach o dużej złożoności, w której przestrzeń rozwiązań

jest ogromna, a znalezienie najlepszego rozwiązania w rozsądnym czasie nie jest zapewniane przez żaden inny deterministyczny algorytm.

Schemat działania



Powyższy schemat przedstawia najistotniejsze etapy wykorzystania algorytmu genetycznego w praktyce. Analiza tego schematu krok po kroku pozwoli podsumować wiedzę zdobytą przy okazji ogólnej charakterystyki algorytmu genetycznego.

Pierwszym i najważniejszym krokiem jest wyczerpujące sprecyzowanie problemu do rozwiązania. Należy szczegółowo określić co ma być rozwiązane/zoptymalizowane, jakie mierzalne parametry są istotne z punktu widzenia problemu, jakie wielkości są poszukiwane, z jakiego zbioru wartości one pochodzą i z jaką dokładnością mają być reprezentowane w potencjalnych rozwiązaniach. Po dokładnym zdefiniowaniu zadania do zoptymalizowania można przystąpić do następnego kroku.

Drugim niemniej ważnym krokiem jest sprecyzowanie powyższego problemu w kontekście algorytmu genetycznego. Należy określić w jaki sposób będzie reprezentowany genotyp osobnika, będącego potencjalnym rozwiązaniem zadania, oraz jakie wartości będą mogły przyjmować geny osobnika. Następnie należy określić funkcję przystosowania każdego osobnika w populacji w taki sposób, aby wyższa jej wartość odzwierciedlała „optymalność” rozwiązania. Dla tak zdefiniowanych osobników oraz ich przystosowań należy oszacować wielkość populacji. Jak się okazuje oszacowanie liczby osobników w populacji wcale nie jest łatwe i w znaczny sposób wpływa na wyniki działania algorytmu genetycznego. Przy określaniu tej liczby warto m.in. zwrócić uwagę na długość procesu obliczeniowego związanego z obliczaniem przystosowania osobnika. Przystosowanie osobnika może sprowadzać się do obliczenia wartości zwykłej funkcji, ale może to być też wartość powstała w wyniku wykonywania wielu doświadczeń. Należy jednak pamiętać, że im mniejszą liczbę osobników liczy populacja, tym mniejsza jest różnorodność poszukiwanych rozwiązań. Może się okazać, że algorytm genetyczny szybko skończy swoje działanie osiągając najbliższe optimum lokalne, do którego zbiegną wszystkie osobniki. Przy większej liczbie osobników, a tym samym większej różnorodności rozwiązań, znalezione optimum będzie bardziej obiecujące. Warto jednak wciąż pamiętać o długości procesu obliczeniowego dotyczącego jednej populacji, gdyż zbieżność do pewnego optymalnego rozwiązania może pojawić się na przykład dopiero w setnej populacji. Ostatecznie należy

sprecyzować jeszcze techniki selekcji osobników populacji do reprodukcji, a także zbiór operatorów genetycznych wraz z prawdopodobieństwami ich stosowania, które wykorzystywane są w procesie reprodukcji.

Dokładne wykonanie dwóch poprzednich kroków pozwala ostatecznie przystąpić do fizycznej realizacji algorytmu. Po udanej implementacji spełniającej wszystkie nałożone założenia oraz ograniczenia pozostaje jedynie określić osobników populacji początkowej. Zazwyczaj postać tych osobników określa się w sposób losowy, czasami jednak wykorzystując dodatkową wiedzę na temat rozwiązywanego problemu można już na wstępie ukierunkować populację początkową poprzez ręczne ustalenie genotypów osobników.

Ostatnim krokiem jest wystartowanie algorytmu genetycznego dla pokolenia początkowego i pozostawienie wszystkiego w mocy ewolucji. Dla każdej populacji algorytm ocenia czy znaleziono już satysfakcjonujące rozwiązanie – jeżeli tak, to kończy działanie, w przeciwnym przypadku, o ile nie została przekroczona maksymalna liczba pokoleń, algorytm dokonuje selekcji zgodnie z wybraną techniką selekcji, a następnie przystępuje do reprodukcji wybranych osobników wykorzystując zaimplementowane operatory genetyczne.

Wykorzystanie algorytmu genetycznego

Po krótkim scharakteryzowaniu działania algorytmu genetycznego możemy już przejść do jego zastosowania w analizowanej pracy dyplomowej.

Definicja problemu

Zadaniem algorytmu genetycznego będzie znalezienie najlepszej postaci funkcji oceniającej, która służyć będzie do oceny pozycji figur na szachownicy z perspektywy określonego koloru. Wykorzystując tak określoną funkcję oceniającą można oceniać stan gry poprzez różnicę wartości funkcji oceniającej z perspektywy koloru białego z wartością funkcji oceniającej z perspektywy koloru czarnego. Autor pracy definiując funkcję oceniającą sprecyzował jej 24 parametry przyjmujące wartości ze zbioru liczb całkowitych, na które zostały nałożone jeszcze dodatkowe ograniczenia. Poniżej omówione są cechy gry wpływające na obliczanie funkcji oceniającej (kursywą wypisane są parametry tej funkcji):

- § wartość materialna figur:
 - § pionek = 100,
 - § *Skoczek* $\in [200;500]$,
 - § *Goniec* $\in [200,500]$,
 - § *Wieża* $\in [400;600]$,
 - § *Hetman* $\in [800,1100]$,
- § faza gry:
 - § debiut – jeżeli 3 lekkie figury stoją na 1 lub 8 linii,
 - § końcówka – jeżeli pozycja nie jest debiutem i gdy wartość figur na szachownicy nie przekracza zmiennej *MaterialWKoncowce* $\in [1000,2000]$ albo na szachownicy znajduje się co najmniej jeden hetman i wartość figur na szachownicy nie przekracza $2 * \textit{MaterialWKoncowce}$,
 - § matowanie – jeżeli pozycja jest końcówką i jedna ze stron ma przewagę większą niż *PrzewagaPrzyMatowaniu* $\in [400,1000]$,
- § ocena pozycyjna pionków:
 - § za każdego izolowanego pionka (takiego, dla którego w sąsiednich kolumnach nie znajdują się inne pionki tego samego koloru) odejmowana jest *KaraZaIzolowanePionki* $\in [0,100]$,
 - § za każdego zdublowanego pionka (takiego, dla którego w tej samej kolumnie znajduje się drugi pionek) odejmowana jest *KaraZaZdwojonePionki* $\in [0,100]$,
 - § jeżeli na polach e2 i d2 dla koloru białego bądź analogicznie na polach e7 i d7 dla koloru czarnego stoją pionki tego koloru to odejmowana jest *KaraZaD2E2* $\in [60,300]$,
 - § jeżeli na co najmniej jednym z pól d2 i d3 stoi biały pion (analogicznie na co najmniej jednym z pól d7 i d6 stoi czarny pion) odejmowana jest *KaraZaD2D3* $\in [0,150]$,
 - § jeżeli na co najmniej jednym z pól e2 i e3 stoi biały pion (analogicznie na co najmniej jednym z pól e7 i e6 stoi czarny pion) odejmowana jest *KaraZaE2E3* $\in [0,150]$,
 - § jeżeli pozycja jest końcówką, to dla każdego pionka doliczana jest $(\textit{PremiaZaZaawansowaniePionka} \in [0,100]) * < \text{numer linii} >$,

§ matowanie:

§ jeżeli pozycja jest matowaniem, to dla strony matującej doliczana jest wartość $10^* < \text{odleglosc matowanego krola od srodka szachownicy} >$, gdzie odległość matowanego króla od środka szachownicy zdefiniowana jest przez poniższą tabelkę:

5	5	4	3	3	4	5	5
5	4	3	2	2	3	4	5
4	3	2	1	1	2	3	4
3	2	1	0	0	1	2	3
3	2	1	0	0	1	2	3
4	3	2	1	1	2	3	4
5	4	3	2	2	3	4	5
5	5	4	3	3	4	5	5

§ za każdą bierkę strony matującej odejmuje się $2^* < \text{odleglosc od czarnego krola} >$, gdzie odległość dwóch pól (x_1, y_1) oraz (x_2, y_2) , to $\max(|x_1 - x_2|, |y_1 - y_2|)$ dla $x_i, y_i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.

§ ocena króla:

§ jeżeli pozycja jest końcówką, to odejmowana jest wartość $(KrolWCentrum \in [0, 100])^* < \text{odleglosc krola od srodka szachownicy} >$, w przeciwnym przypadku dodawana jest ta wartość,

§ jeżeli pozycja nie jest końcówką, to za króla stojącego na bezpiecznych polach a1,b1,g1,h1 dla białych i analogicznie a8,b8,h8,h8 dla czarnych, dodawana jest $PremiaZaBezpiecznegoKrola \in [0, 100]$,

§ rozwój:

§ za każdą lekką figurę na 1 linii (analogicznie na 8 linii dla czarnych) odejmowana jest $KaraZaBrakRozwoju \in [0, 100]$,

§ ocena pozycyjna skoczków:

§ w debiucie za każdego skoczka stojącego na 1 linii (analogicznie na 8 linii dla czarnych) odejmowana jest $KaraZaBrakRozwojuSkoczka \in [0, 100]$,

§ w końcówce za każdego skoczka odejmowana jest $(KaraZaOdlegloscSkoczkaOdKrola \in [0, 100])^* < \text{odleglosc skoczka od krola} >$,

§ jeżeli pozycja nie jest końcówką, to odejmowana jest $(KaraZaOdlegloscSkoczkaOdCentrum \in [0, 100])^* < \text{odleglosc skoczka od srodka szachownicy} >$,

§ ocena pozycyjna gońców oraz wież:

§ za każdego gońca dodawana jest $(PremiaZaIloscRuchowGoncow \in [0, 10])^* < \text{liczba mozliwych ruchow gonca} >$,

§ za każdą wieżę dodawana jest $(PremiaZaIloscRuchowWiezy \in [0, 10])^* < \text{liczba mozliwych ruchow wiezy} >$,

§ za każdą wieżę na 7 linii dla białych (analogicznie na 2 linii dla czarnych) dodawana jest $PremiaDlaWiezyNa7 \in [0, 100]$,

§ ocena pozycyjna hetmanów:

§ w końcówce za każdego hetmana odejmowana jest $(KaraZaOdlegloscHetmanaOdKrola \in [0, 20])^* < \text{odleglosc hetmana od krola} >$,

§ jeżeli pozycja nie jest końcówką, to odejmowana jest $(KaraZaOdlegloscHetmanaOdCentrum \in [0, 20])^* < \text{odleglosc hetmana od srodka szachownicy} >$,

§ jeżeli król przeciwnika nie znajduje się na jednym z bezpiecznych pól, to za hetmana dodawana jest $PremiaDlaHetmanaZaOslabienieKrolaPrzeciwnika \in [0, 100]$.

Autor pracy nie uzasadnił powodów, dla których ustalił takie, a nie inne ograniczenia dziedzinowe parametrów, wspominając jedynie o tym, że zostały one wyznaczone wraz z ekspertem dziedzinowym (szachistą). Argumentacja dotyczyła jedynie intuicyjnych wartości materialnych figur, dla których np. wartość 0 jest bezsensowna, a dla których w rzeczywistości przyjęły się pewne wartości w celu kształtowania wyobraźni początkujących szachistów (pionek-1, skoczek-3, gонец-3, wieża-5, hetman-9/10).

Jak widać zaproponowana funkcja oceniająca jest na tyle złożona – liczba wszystkich możliwych kombinacji wartości parametrów wynosi w przybliżeniu 10^{45} – że ciężko jest określić optymalne wartości jej parametrów. Świetnym rozwiązaniem w poszukiwaniu optymalnej postaci tej funkcji będzie zastosowanie algorytmu genetycznego. Warto przy tym zauważyć, że algorytm genetyczny zostanie wykorzystany tylko i wyłącznie w celu wyznaczenia funkcji oceniającej, a nie będzie on wykorzystywany w końcowej realizacji tego programu. W ostatecznej wersji programu szachowego zostanie po prostu zastosowana najlepsza postać funkcji oceniającej. Jest to jedno z podejść wykorzystania algorytmów genetycznych w grach, tzn. wykorzystanie ich w celu znalezienia najlepszej „strategii gry”.

Definicja problemu w kontekście algorytmu genetycznego

Każdy osobnik populacji reprezentuje zestaw wartości 24 parametrów funkcji oceniającej. Każdy parametr będący genem osobnika reprezentowany jest przez liczbę całkowitą. Pierwszym i największym problemem staje się w tym momencie określenie funkcji przystosowania dla osobników populacji. Jak wiadomo funkcja ta ma w jednoznaczny sposób określić lepszość jednego z dwóch rozwiązań. Niestety okazuje się, że dla rozpatrywanego problemu określenie takiej funkcji nie jest łatwe, nie mówiąc już o wyznaczeniu dla niej wzoru matematycznego. Trudność w jej określeniu bierze się ze względnego pojęcia „lepszy”, którego w analizowanym zagadnieniu nie da się jednoznacznie sprecyzować. Patrząc przez analogię do rzeczywistości można by zadać sobie pytanie, który z dowolnie wybranych dwóch szachistów jest lepszy? Oczywiście, aby to stwierdzić należałoby skonfrontować ich ze sobą. Niestety jeden pojedynek nie może być podstawą oceny wyższości jednego szachisty nad drugim. Wypadałoby rozegrać między nimi mecz złożony z większej liczby pojedynków i dopiero wówczas wysnuć odpowiedni osąd. Okazuje się jednak, że powstaje tutaj kolejny problem – czy możemy jednoznacznie stwierdzić, że jeżeli szachista s_1 jest lepszy od szachisty s_2 , a szachista s_2 jest lepszy od szachisty s_3 , to szachista s_1 jest lepszy od szachisty s_3 ? Oczywiście sprawa nie jest taka prosta. W przypadku, gdy różnica w poziomie gry pomiędzy kolejnymi szachistami jest znacząca, to powyższe stwierdzenie byłoby prawdziwe. Jednak w przypadku, gdy poziom gry jest prawie porównywalny, to na podstawie dwóch wymienionych przesłanek nie można wysnuć takich wniosków, gdyż może się okazać, że to właśnie szachista s_3 jest lepszy od szachisty s_1 . W związku z tym autor pracy doszedł do wniosku, że najlepszym rozwiązaniem będzie przeprowadzanie turniejów dla wszystkich osobników populacji w systemie „każdy z każdym” i na podstawie jego wyników wyłaniania będzie potencjalna lepszość jednych osobników nad drugimi. Oczywiście tak jak w życiu może się okazać, że przeprowadzenie po raz drugi tego samego turnieju dałoby w rezultacie odmienne wyniki.

Ze względu na ograniczony czas przeprowadzania eksperymentów autor pracy przyjął ostatecznie, że:

- § przystosowanie osobników wyznaczane jest przez ich wyniki w przeprowadzanym w ramach populacji turnieju systemem „każdy z każdym”,
- § liczba osobników w populacji jest liczbą parzystą N , dla której należy rozgrywać $\frac{N(N-1)}{2}$ partii,
- § kolejne partie każdego osobnika rozgrywane są kolorami na przemian (tak jak w prawdziwym turnieju),
- § każda partia trwa maksymalnie 70 ruchów,
- § za zwycięstwo przez zamotowanie króla przeciwnika, wygrany otrzymuje (20000-<numer ostatniego ruchu>) punktów, natomiast przegrany otrzymuje przeciwną wartość punktów,
- § w przypadku nie zakończenia partii do 70 ruchu, ocenie podlega wartość materialna figur stojących na szachownicy: osobnik grający kolorem białym otrzymuje liczbę punktów równą różnicy sumy wartości materialnych białych figur z sumą wartości materialnych czarnych figur, natomiast, osobnik grający kolorem czarnym otrzymuje przeciwną liczbę punktów,

Jak widać funkcja przystosowania została tak zdefiniowana, aby promować osobników potrafiących matować swoich przeciwników. Można łatwo sprawdzić, że osobnik, któremu udało się zamatować króla przeciwnika w najgorszym wypadku uzyska 19930 punktów, natomiast osobnik, któremu nie udało się zamatować, ale uzyskał znaczną przewagę w najlepszym skrajnym przypadku posiadając wszystkie figury podczas gdy jego przeciwnik tylko króla, uzyska 6200 punktów (o ile parametry odzwierciedlające wartości materialne figur przyjmować będą maksymalne dopuszczalne wartości).

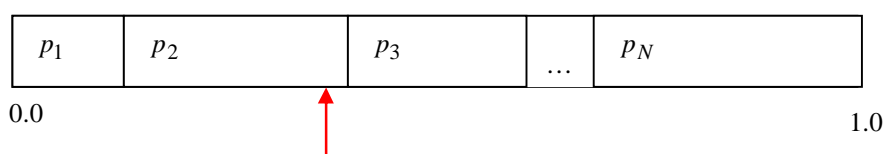
W ten sposób największy, a zarazem najistotniejszy problem został potencjalnie rozwiązany. Pozostało jedynie określić techniki selekcji, operatory genetyczne wykorzystywane podczas reprodukcji oraz liczebność populacji.

W ramach selekcji i reprodukcji zaproponowany został następujący model: spośród N osobników populacji po rozegraniu turnieju wyłaniania jest tzw. elita – dwóch najlepszych osobników – która przechodzi do

następnej populacji. W tym momencie w przedstawianej pracy pojawia się pewna sprzeczność. Autor stwierdza, że pozostała reszta $N-2$ osobników podlega selekcji wg metody stochastycznej z powtórzeniami, tzw. metody ruletki, w ramach której mogą zostać wyselekcjonowane również dwa najsilniejsze osobniki. Stąd też wydaje się, że mimo wszystko autor miał na myśli poddanie wszystkich osobników populacji powyższej metodzie selekcji. Metoda ta polega na stworzeniu „ruletki” na podstawie przystosowań wszystkich osobników podlegających selekcji. Wspomniana „ruletka” jest tak naprawdę przedziałem wartości rzeczywistych $[0,1]$, z którego każdemu osobnikowi przypisana jest jego część (rozłączna względem części innych osobników). Części te wyznaczone są na podstawie prawdopodobieństwa ich wylosowania wyznaczonego jako

$$p_i = \frac{P_i}{\sum_{j=1}^N P_j}$$

gdzie $i \in \{1,2,\dots,N\}$ dotyczy i -tego osobnika populacji, natomiast P_i jest jego przystosowaniem.



Przykład jednego „obrót ruletki”

Powyższy przykład przedstawia wybranie drugiego osobnika poprzez pojedynczy „obrót ruletki” sprowadzający się do wylosowania liczby rzeczywistej z przedziału $[0,1]$. W analizowanym zagadnieniu spośród N osobników populacji należy zatem wybrać w powyższy sposób $N-2$ osobników. Oczywiście może się okazać, że $N-2$ razy zostanie wylosowany tylko i wyłącznie jeden najlepszy osobnik. Jak sama nazwa wskazuje jest to metoda stochastyczna z powtórzeniami, a więc jeśli jakiś osobnik przewyższa w znaczny sposób średnie przystosowanie populacji, to może się zdarzyć, że zostanie on wylosowany dokładnie $N-2$ razy. Wyznaczona w ten sposób grupa $N-2$ osobników podlega reprodukcji zgodnie z zaimplementowanymi operatorami genetycznymi.

W pracy zostały wykorzystane właściwie nieco zmodyfikowane dwa klasyczne operatory genetyczne: krzyżowanie oraz mutacja.

Krzyżowaniu podlegają wszystkie skojarzone pary kolejno wyselekcjonowanych osobników, a więc 1 z 2, 3 z 4, ..., $N-3$ z $N-2$. Dla każdej takiej pary, osobniki krzyżują się między sobą z prawdopodobieństwem $p_c = 0.9$. Krzyżowania polegające na wymianie materiału genetycznego osobników przebiega w następujący sposób:

$$\begin{pmatrix} a_1, a_2, a_3, \dots, a_{i-1} | a_i, \dots, a_{23}, a_{24} \\ b_1, b_2, b_3, \dots, b_{i-1} | b_i, \dots, b_{23}, b_{24} \end{pmatrix} \rightarrow \begin{pmatrix} a_1, a_2, a_3, \dots, a_{i-1}, \frac{a_i + b_i}{2}, \dots, \frac{a_{23} + b_{23}}{2}, \frac{a_{24} + b_{24}}{2} \\ \frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}, \frac{a_3 + b_3}{2}, \dots, \frac{a_{i-1} + b_{i-1}}{2}, b_i, \dots, b_{23}, b_{24} \end{pmatrix}$$

- § losowany jest wspólny dla obu osobników punkt przecięcia znajdujący się pomiędzy genem $i-1$ oraz genem i , gdzie $i \in \{2,3,\dots,24\}$,
- § część pierwszego potomka znajdująca się na prawo od punktu przecięcia powstaje w wyniku uśrednienia każdej wartości genów obu rodziców z tej samej części, natomiast część pierwszego potomka znajdująca się na lewo od punktu przecięcia jest identyczna z tą samą częścią pierwszego rodzica,
- § analogicznie powstają części drugiego potomka z tymże uśrednieniu podlega część lewa a część prawa jest identyczna z tą samą częścią drugiego rodzica.

Powyższy sposób krzyżowania jest nieco zmodyfikowanym szczególnym przypadkiem krzyżowania arytmetycznego, w którym geny potomków wyliczane są następująco:

Potomek I: $g_i = k * a_i + (1-k) * b_i$, gdzie a_i to i - ty gen rodzica I, b_i to i - ty gen rodzica II, $k \in (0,1)$
Potomek II: $g_i = k * b_i + (1-k) * a_i$

Łatwo zauważyć, że szczególny przypadek tego krzyżowania dla $k = \frac{1}{2}$ sprowadza się do wyliczenia średnich arytmetycznych $g_i = \frac{a_i + b_i}{2}$ dla każdego genu obu potomków.

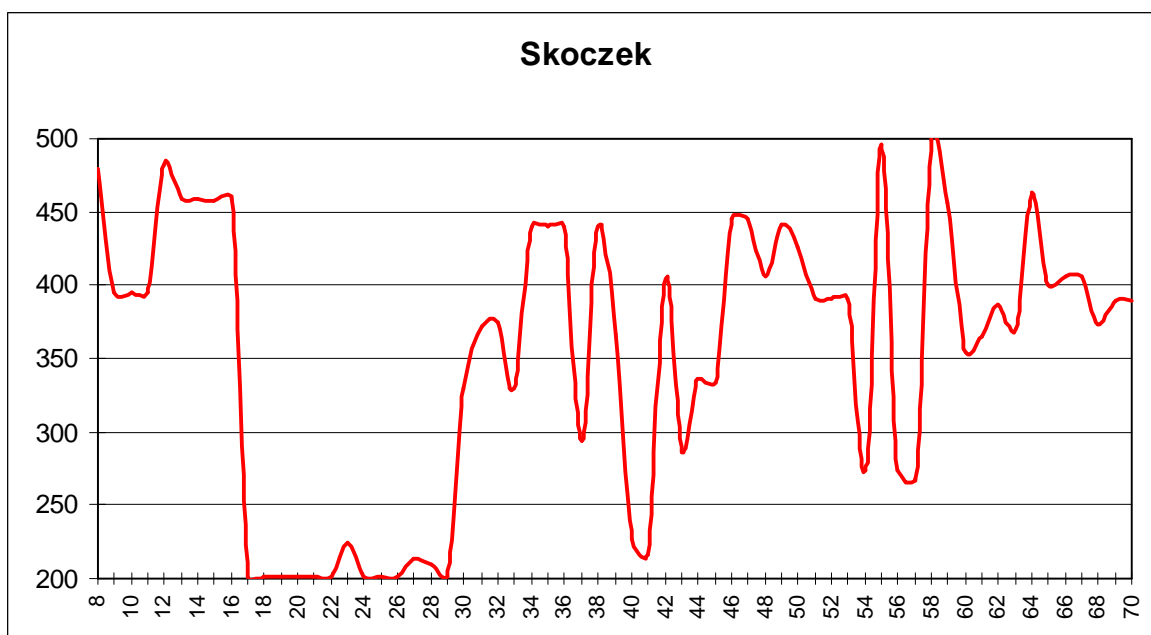
Mutacji podlegają wszystkie wyselekcjonowane osobniki ($N-2$), w taki sposób, że każdy parametr każdego osobnika konwertowany jest na binarny kod Graya, w którym zmiana dowolnego bitu powoduje zmianę początkowej wartości całkowitej parametru dokładnie o 1. Następnie każdy bit powyższej postaci binarnej parametru podlega mutacji, tzn. jest negowany, z prawdopodobieństwem $p_m = 0.01$. Po uwzględnieniu każdego bitu postaci binarnej parametru, jest ona ponownie konwertowana na wartość całkowitą. Wynika stąd, że w skrajnym przypadku mutacja może spowodować zmianę wartości całkowitej parametru o liczbę równą długości łańcucha postaci binarnej parametru. Jeżeli okaże się, że mutacja spowoduje wykroczenie wartości parametru poza dopuszczalny przedział wartości, to wartość tego parametru jest losowo wybraną liczbą całkowitą z tego przedziału.

Ostatnią najważniejszą sprawą jest ustalenie liczebności ewoluującej populacji. Ze względu na bardzo czasochłonny proces obliczania wartości funkcji przystosowania osobnika każdej populacji, autor założył, że wystarczającą liczbą osobników będzie 8. Zakładając, że każda partia dwóch osobników trwa około 5 minut, to rozegranie wszystkich partii w pokoleniu trwać będzie $\frac{8*7}{2} * 5 \text{ minut} = 140 \text{ minut} = 2 \text{ h } 20 \text{ minut}$. Jak widać nawet dla 8 osobników populacji, proces wyznaczenia przystosowania każdego z nich jest naprawdę bardzo kosztowny.

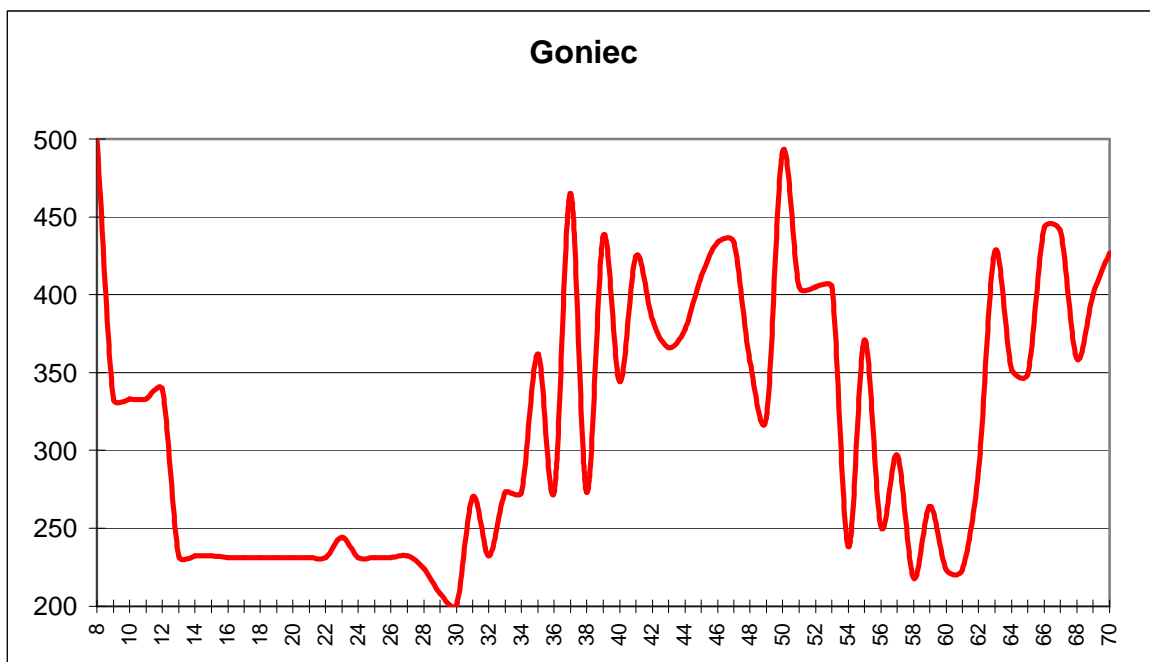
Przeprowadzenie eksperymentów

Po zdefiniowaniu zagadnienia w kontekście algorytmu genetycznego autor pracy mógł wreszcie przystąpić do przeprowadzenia eksperymentów. Okazało się że w przeciągu dwóch miesięcy udało mu się przeprowadzić jedynie 70 turniejów, w ramach których rozegrano około 2 tysiące partii, chociaż wstępnym założeniem było przeprowadzenie nawet kilkuset turniejów.

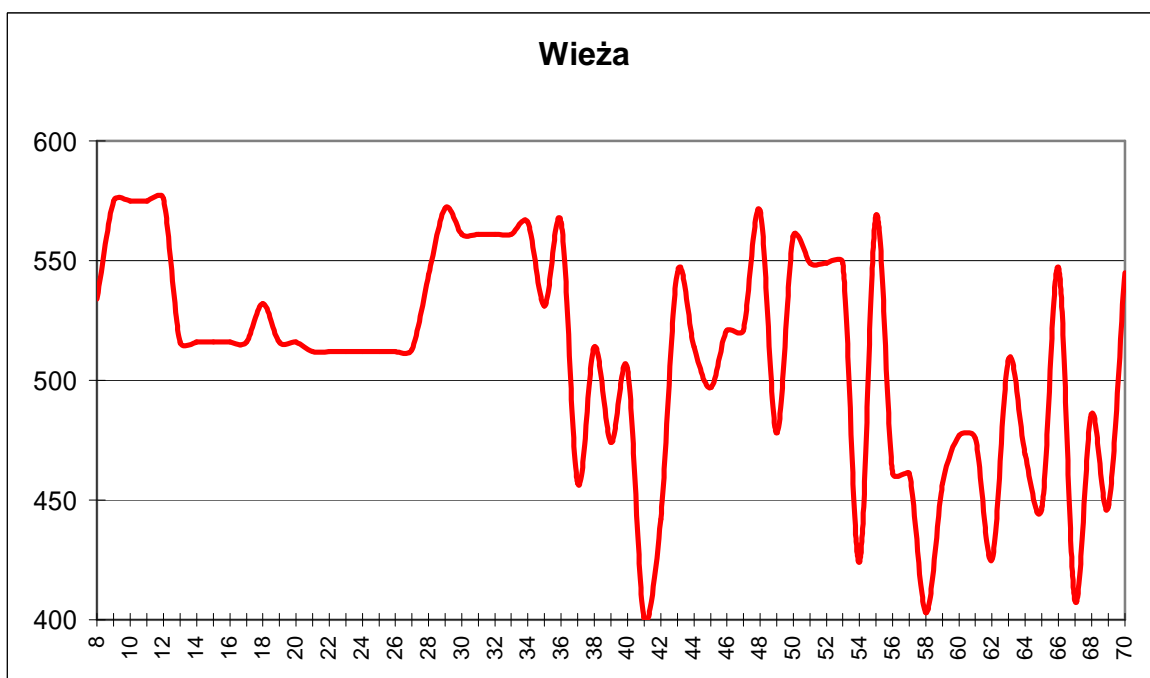
Korzystając z wyników badań umieszczonych w pracy sporządziłem wykresy wartości w zależności od numeru pokolenia dla czterech parametrów, których intuicyjne wartości są znane: *Skoczek*, *Goniec*, *Wieża*, *Hetman*. Należy tutaj jeszcze dodać, że wyniki te dotyczą najlepszych osobników w każdym pokoleniu. Zgodnie z rzeczywistością wykładaną szachistom początkującym wartości tych parametrów powinny być zbliżone do następujących wartości: *Skoczek*=300, *Goniec*=300, *Wieża*=500, *Hetman*=1000. Spójrzmy jakie wyniki powstały w wyniku zastosowania algorytmu genetycznego:



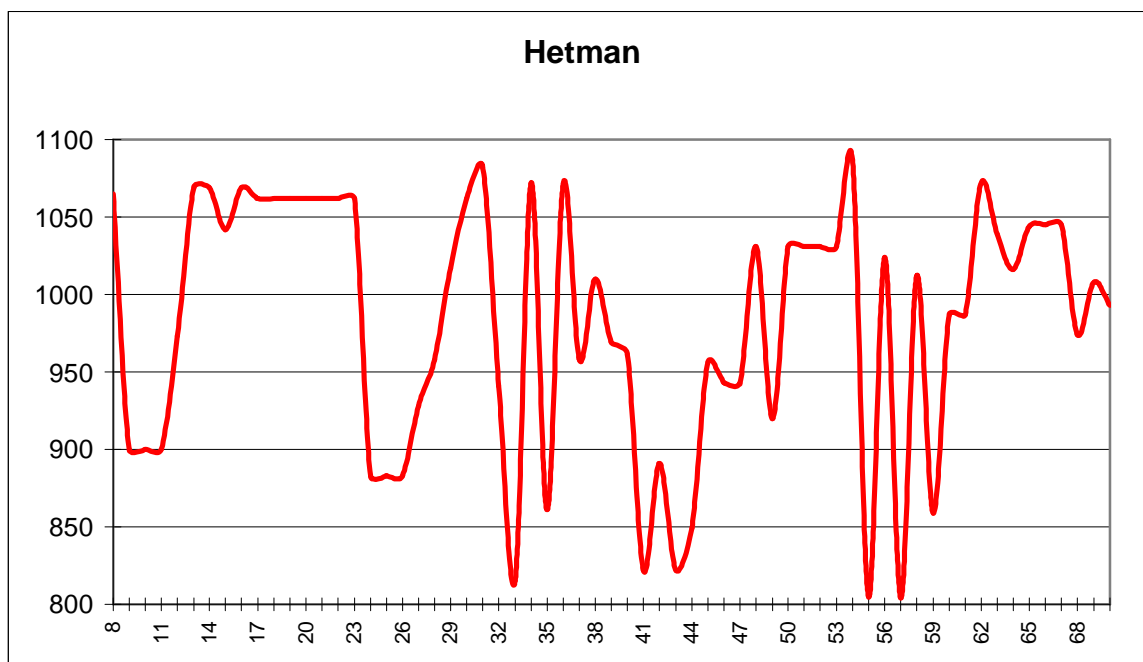
W przypadku parametru *Skoczek* widać, że na przestrzeni wszystkich pokoleń jego wartość oscylowała od najmniejszych do największych wartości. Wydawałoby się, że pod koniec eksperymentu wartość ta nieco się ustabilizowała, ale nie można mieć pewności, że to już jest punkt zbieżności. Warto zauważyć, że począwszy od 17 pokolenia aż do 29 pokolenia wartość ta utrzymywała się w granicach 200, by w końcu wystrzelić w górę i uzyskać bardzo dużą amplitudę. Niestety liczba pokoleń w ramach zastosowanego algorytmu genetycznego nie jest na tyle zadowalająca aby jednoznacznie stwierdzić, czy *Skoczek* rzeczywiście osiągnął swój punkt zbieżności.



W przypadku parametru *Goniec* widać pewną analogię do parametru *Skoczek*. Ponownie okazuje się, że liczba pokoleń algorytmu genetycznego nie była wystarczająca, aby uzyskać ostateczny punkt zbieżności tej wartości. Pod koniec eksperymentu wartość *Gońca* oscylowała w przedziale wartości [350,450]. Można również zauważyć, że podobnie jak w przypadku *Skoczka*, wartość *Gońca* począwszy od pokolenia 13 do pokolenia 28 utrzymywała się w granicach 230, by w końcu wystrzelić w górę i uzyskać bardzo dużą amplitudę.



Najgorzej sprawa wygląda w przypadku *Wieży*. Okazuje się, że pod koniec eksperymentu wartość tego parametru oscylowała w bardzo szerokim przedziale wartości [400;550], czyli praktycznie w $\frac{3}{4}$ części wszystkich dopuszczalnych wartości. Co ciekawe po raz kolejny można zauważyć, że podobnie jak w przypadku *Skoczka* oraz *Gońca*, wartość *Wieży* począwszy od pokolenia 13 do pokolenia 27 utrzymywała się w granicach 520, by w końcu całkowicie rozprzestrzenić się po całym dopuszczalnym przedziale wartości.



Nieco lepiej od *Wieży*, ale nadal nienajlepiej, wygląda sprawa *Hetmana*. Na przestrzeni całego eksperymentu widać wielkie oscylacje od minimalnych wartości z jednego pokolenia do maksymalnych wartości z następnego pokolenia. Koniec eksperymentu daje niewielkie nadzieje, że wartość tego parametru zaczyna zbiegać do wartości 1000, ale podobnie jak w przypadku *Skoczka*, nie można jednoznacznie stwierdzić czy tak w istocie będzie.

Krótkie podsumowanie przeprowadzonych eksperymentów

Niestety nie można być zachwyconym wynikami przeprowadzonych eksperymentów, ponieważ w gruncie rzeczy nie zrealizowały one swojego głównego zadania, tzn. nie wyznaczyły jednoznacznie najlepszej postaci funkcji oceniającej. Tym bardziej potwierdza to złożoność problemu do rozwiązania. Na pewno wielką szkodą jest fakt, że autorowi nie udało się przeprowadzić większej liczby turniejów – może okazałoby się, że już niewiele brakowało, aby wartość jakiegoś parametru się ustabilizowała. Kolejnym niedociągnięciem jest fakt, że jedna populacja liczyła zaledwie 8 osobników, a jest to naprawdę bardzo mała liczba. Jedni mogliby powiedzieć „niestety”, a drudzy „na szczęście” problem programowej realizacji szachów jest na tyle złożony, że prawie niemożliwe staje się znalezienie jedynie słusznego rozwiązania.

Literatura

Ramon Jimenez, *The Rook Endgame Machine of Torres y Quevedo*,
<http://www.chessbase.com/newsdetail.asp?newsid=1799>, 2004

Von Kempelen's Chess Turk recreated, <http://www.chessbase.com/newsdetail.asp?newsid=1574>, 2004

Frederic Friedel, *A short history of computer chess*,
<http://www.chessbase.com/columns/column.asp?pid=102>, 2004

David G. Stork, *The end of an era, the beginning of another? HAL, Deep Blue and Kasparov*,
<http://www.research.ibm.com/deepblue/learn/html/e.shtml>, 1997

Bill Wall, *Computer Chess History*, 2004, <http://www.geocities.com/SiliconValley/Lab/7378/comphis.htm>

Wolna Encyklopedia Wikipedia, <http://pl.wikipedia.org>

Tomasz Michniewski, *Samouczenie programów szachowych – praca magisterska*, Wydział Matematyki,
Informatyki i Mechaniki Uniwersytetu Warszawskiego, 1995