

Grzegorz ZIELIŃSKI

## NOWE PODEJŚCIE DO PROBLEMATYKI AI Z WYKORZYSTANIEM MODUŁÓW CODI - WYNIKI PRAC ZESPOŁU EEC „ATLANTIS”

**Streszczenie.** Dokument opisuje prace zespołu EEC „Atlantis” nad problemami sztucznej inteligencji opartej na modułach CoDi, prezentujące wybrane zagadnienia przybliżające specyfikę podejścia, jak również nowatorską metodykę ich ewoluowania wykorzystującą algorytmy genetyczne.

### A NEW APPROACH TO AI PROBLEM USING CODI MODULES – EEC „ATLANTIS” TEAM WORK RESULTS

**Summary.** The document describes the work of EEC “Atlantis” at artificial intelligence problems based on CoDi modules. It also presents selected questions that approximate their specific approach to AI and also shows the innovative methodology of their evolution using genetic algorithms.

## 1. Wprowadzenie

Od lat w zagadnieniach, takich jak sterowanie, rozpoznawanie obrazów, wspomaganie decyzji itp. wykorzystuje się elementy sztucznej inteligencji. Jednak mimo postępującego zaawansowania tych prac daleko jeszcze do otrzymania „prawdziwego” sztucznego intelektu. W przypadku najbardziej popularnej metody budowy AI – w oparciu o sieci neuronowe – największym problemem jest gwałtowny wzrost komplikacji tak procesu projektowania, jak i uczenia sieci wraz ze wzrostem ilości neuronów składowych.

Nowe podejście do problemu zaproponowane przez zespół naukowców z Instytutu ATR pod kierownictwem dra Hugo de Garisa [1] stanowi połączenie trzech dziedzin: sieci neuronowych, algorytmów genetycznych i automatów komórkowych. Zdefiniowano swego rodzaju „elementarną sieć neuronową”, którą następnie zamknięto w pudełku automatu komórkowe-

go. W ten sposób otrzymano kostkę 24x24x24 komórek – moduł CoDi (Collect and Distribute). Kostka ta jest „zawinięta” – każda komórka ma więc 6 sąsiadów, brak komórek skrajnych. Opis kostki nie zawiera jednak schematu sieci, lecz jej stan początkowy – rozmieszczenie neuronów, oraz instrukcje wzrostu zapisywane w pozostałych, pustych komórkach. Taka pierwotna kostka przechodzi fazę wzrostu, w której z neuronów wyrastają drzewa dendrytów i aksonów, rozprzestrzeniając się zgodnie z napotkanymi w kolejnych komórkach instrukcjami wzrostu. Po zakończeniu procesu w kostce znajduje się gotowa sieć neuronowa, zawierająca maksymalnie 1152 neurony. Jest to niewiele, jednak wystarczy do stworzenia modułu wykonującego proste czynności. Moduły takie można następnie łączyć w wielomodułowe struktury, otrzymując sieci o dowolnym stopniu komplikacji. Zapis struktury sieci nie wprost ma jeszcze dwie zalety: pozwala pominąć proces projektowania sieci – poszczególne moduły otrzymuje się na drodze „ewolucji”, za pomocą algorytmu genetycznego. Pozwoli to w przyszłości ewoluować całe struktury. Drugą zaletą sposobu opisu sieci jest duża elastyczność – mutacja pojedynczych komórek nie grozi odcięciem całych obszarów sieci. Jednocześnie specyfika zastosowanego opisu zapewnia jednoznaczność – dla jednego stanu początkowego istnieje jedna i tylko jedna sieć.

Metoda ta czyni realne operowanie na strukturach rzędu setek milionów neuronów. W projekcie budowy wysoce zaawansowanej sztucznej inteligencji biorą udział zespoły z dziewięciu krajów. Jednym z nich jest polski zespół stworzony w 1998 r. przez dra Andrzeja Bullera z Politechniki Gdańskiej – GABRI (Gdańsk Artificial Brain Research Initiative) [3]. W połowie roku 1999 GABRI skupiało już entuzjastów z całej Polski. Nastąpił podział na Kluby Inżynierii Ewolucyjnej. Jednym z nich jest EEC „Atlantis” (EEC – Evolutionary Engineering Club).

EEC „Atlantis”, którego autor tego artykułu jest koordynatorem, podobnie jak pozostałe Kluby, jest jednostką rozproszoną. Poszczególni członkowie Klubu komunikują się z koordynatorem za pośrednictwem sieci Internet. Początkowo współpraca z Klubem przypomina inny rozproszony projekt – SETI@Home: członkowie otrzymują pocztą elektroniczną symulator wraz z instrukcją obsługi i przykładowym eksperymentem. Stopniowo wgłębiają się w Projekt wykonując coraz bardziej samodzielne prace. Rozdział zadań i analiza wyników należą do koordynatora Klubu. Koordynatorzy poszczególnych EEC otrzymują bardziej ogólnie sformułowane problemy od Głównego Koordynatora GABRI, ten zaś komunikuje się bezpośrednio z drem Bullerem z Instytutu Starlab. Struktura organizacyjna GABRI jest nieprzypadkowa – odpowiada ona założeniom wspomnianej wyżej strategii top-bottom-top: główny projektant dzieli zadanie na kilka – kilkanaście w miarę niezależnych części i przekazuje opis

wymaganych cech każdej z części odpowiedniemu projektantowi poziomu drugiego, ten rozdziela pracę między projektantów kolejnego poziomu – proces trwa do czasu aż projektanci n-tego poziomu otrzymają zadania możliwe do wykonania za pomocą pojedynczych modułów CoDi. Wyhodowane przez siebie moduły przekazują na poziom n-1, gdzie tworzone są połączenia międzymodułowe, po czym gotowe fragmenty przechodzą na poziom n-2 itd. do czasu aż główny projektant otrzyma gotowe komponenty pozwalające na złożenie gotowej struktury. Jednak opisany proces jest w stadium projektu. Aktualnie prowadzone przez EEC „Atlantis” prace dotyczą badań nad własnościami modułów CoDi, podejmowane są też próby wyhodowania modułów pełniących konkretne funkcje (Flip-Flop, moduł opóźniający itp.). Ponadto, oprócz prac zleconych przez kierownictwo projektu, EEC „Atlantis”, podjął badania nad możliwością przyspieszenia procesu otrzymywania modułu – jest to krytyczne z uwagi na duże moce obliczeniowe niezbędne przy użyciu dotychczasowych metod.

Aktualne prace dotyczą specyficznych modułów o zarówno wejściach, jak i wyjściach cyfrowych (w przyszłości planowane są prace nad modułami analogowymi i mieszanymi), jak również nad metodami przyspieszenia procesu ewolucji. Niektóre z wyników tych prac prezentowane są poniżej.

## 2. Moduł przerzutnika Flip-Flop

Przez dłuższy czas sądzono, iż moduły CoDi nie będą wykazywały zdolności uczenia się. Przez naukę rozumie się tu proces, w wyniku którego działanie modułu jest uwarunkowane wcześniejszymi zdarzeniami. Przełomem było wyhodowanie przez współpracujące zespoły EEC „Silesia” i EEC „Atlantis” modułu Flip-Flop [2]. Działa on podobnie do znanego z elektroniki przerzutnika R-S: pobudzony jedyneką na wejściu wpisującym generuje na wyjściu jedynekę aż do otrzymania jedynekę na wejściu zerującym. Różnica polega na tym, że zarówno zapisująca, jak i kasująca jedynekę musi być nie krótsza niż cztery bity (jest to jedna z cech modułów CoDi – eksperymenty wykazują, iż najmniejsza porcja sygnału rozpoznawana przez moduł to właśnie 4 bity). Przykładowe odpowiedzi modułu Flip-Flop przedstawione są poniżej:



Przytoczone przykłady pokazują dwie interesujące właściwości tego modułu:

1. mimo użycia jedynie 10 zestawów treningowych moduł bardzo dobrze odpowiada na zestawy **nie użyte** w procesie hodowli (średnie przystosowanie dla zestawów testowych wynosi 95 %),
2. do zestawu treningowego 02 zakradł się błąd (oczekiwana odpowiedź jest o 1 bit za krótką), jednak moduł mimo to osiągnął bardzo dobre parametry; moduł **skorygował błąd** swoich nauczycieli; jeśli cecha ta ujawni się w większej liczbie przypadków, można będzie mówić o prawdziwym przełomie – struktury sztucznej inteligencji z autokorekcją już na etapie tworzenia stworzą zupełnie nową klasę w tej dziedzinie.

Przed wszystkim jednak moduł Flip-Flop otwiera drogę do projektowania modułów pamięci.

### 3. Moduł opóźniający

Eksperyment ten jest jeszcze w toku, jednak już dotychczasowe prace dostarczyły materiału dla kilku bardzo interesujących obserwacji.

Celem eksperymentu jest wyhodowanie modułu opóźniającego sygnał. Moduł ma zapamiętywać informację podawaną na wejście pierwsze (informacyjne) tak długo, jak długo wejście drugie (opóźniające) znajduje się w stanie wysokim (jest na nie podawana jedynka). Po zgaszeniu wejścia opóźniającego moduł powinien zwrócić na wyjściu zapamiętaną informację.

Prace nad modułem posuwają się powoli w związku z tym, że większość cech modelu CoDi nie została jeszcze zbadana i opisana.

Odpowiedzi otrzymanego modułu nie odwzorowywały z wystarczającą dokładnością opóźnianego sygnału. Analiza przyczyn pojawienia się takiego błędu oraz porównanie wyników z wynikami wcześniejszych eksperymentów pozwoliło wysunąć tezę, iż moduły CoDi są niewrażliwe na sygnały o długości poniżej czterech bitów (odpowiednik jednej „szpilki” analogowej).

Dalsze prace ujawniły inną cechę modelu – moduły CoDi wydają się być bardzo wrażliwe na symetrię i powtarzalność sygnałów. Niewykluczone, iż ma to związek z cechą zauważoną i potwierdzoną eksperymentalnie przez siostrzany zespół EEC „Silesia”: redundancja wejść

znacznie poprawia szybkość doskonalenia się modułu, jak i osiąganą dokładność. W tym przypadku okazało się, iż sygnały opóźniane w zestawach treningowych były zbyt symetryczne, co w połączeniu ze stałą wartością opóźnienia (32 cykle) dało niepożądany lecz ciekawy, wart głębszego zbadania efekt: moduł „odciął” wejście opóźniające, a pierwsza cztero-bitowa jedyńka na wejściu informacyjnym powoduje rozpoczęcie (z opóźnieniem 32 cykli) generacji na wyjściu sygnału prostokątnego. Symetria i powtarzalność sygnału w zestawach treningowych przyczyniły się również do bardzo szybkiego otrzymania modułu, jak i do wysokiej oceny wystawionej mu przez symulator (odpowiedzi na zestawy treningowe miały 100 % dokładności).

Można przypuszczać, iż kolejne podejście (zestawy treningowe pozbawione wcześniejszych wad: pojedynczych zer / jedynek, nadmiernej symetrii sygnału, stałej wartości opóźnienia) pozwoli otrzymać omawiany moduł. Niestety, uwzględnienie wspomnianych poprawek wymagać będzie znacznego zwiększenia zestawów treningowych (ok. 30 w miejsce dotychczasowych 10). Trwają prace nad nowym symulatorem zdolnym obsłużyć taką liczbę zestawów.

#### 4. Metodyka akceleracji procesu ewolucji modułu

EEC „Atlantis” prowadzi również własne badania w tej dziedzinie. Jednym z osiągnięć wydaje się być metodyka pozwalająca na szacowanie poziomu, jaki dana populacja jest w stanie osiągnąć wskutek ewolucji, na podstawie wyników uzyskanych w pierwszej fazie procesu hodowli [4].

Jednym z największych problemów, z jakimi borykają się zespoły pracujące nad CoDi, jest bardzo duża moc obliczeniowa pochłaniana przez proces ewolucji modułu. Na wykorzystywanych przez kluby EEC komputerach osobistych klasy PC ewolucja pojedynczego modułu trwa od dwu tygodni do dwu miesięcy, przy czym zazwyczaj do osiągnięcia pozytywnych rezultatów niezbędne jest prowadzenie kilkunastu takich hodowli równoległe. W tej sytuacji bardzo istotne jest stworzenie technologii pozwalającej na akcelerację procesu ewolucji.

Analizując wyniki kolejnych eksperymentów nadsyłane przez członków zespołu, zauważono w wielu przypadkach współzależność między wysoką końcową wartością przystosowania modułu a szybkim jej wzrostem w początkowych generacjach. W przypadku gdyby

współzależność ta okazała się prawidłowością, można by zredukować liczbę prowadzonych w późniejszych generacjach hodowli przez wybór najlepszych. Konieczne stało się zarówno potwierdzenie prawidłowości wspomnianej współzależności, jak i dokładne określenie kryteriów selekcji.

Omawiany wcześniej proces otrzymywania sieci neuronowej o pożądanym działaniu można sprowadzić do zagadnienia optymalizacji polegającego na minimalizacji wskaźnika jakości reprezentującego odległość potencjalnego rozwiązania od optimum przy zastosowaniu algorytmu genetycznego do tworzenia lepszych rozwiązań (przeгляд zupełny należy odrzucić ze względu na znaczne rozmiary przestrzeni rozwiązań). Konstrukcja zastosowanego oprogramowania, które dostarcza informacji o dobroci rozwiązania (a nie o odległości od optimum), powoduje nieznaczną modyfikację zagadnienia, polegającą na tym, iż rozpatrujemy przypadek maksymalizacji wskaźnika jakości.

Mimo zastosowania algorytmu genetycznego proces poszukiwania optimum wymaga znacznej mocy obliczeniowej przy stosunkowo przeciętnej efektywności. Przyjmijmy wielkość populacji w każdym z omawianych przypadków wynoszącą 100 osobników, z których każdy jest potencjalnym rozwiązaniem. Zastosowany algorytm genetyczny operuje na populacji o stałej wielkości, przy czym w każdej generacji populacja szeregowana jest ze względu na wartość przystosowania, po czym 20 osobników o najlepszym przystosowaniu przechodzi do następnej generacji bez zmian, reszta zaś ulega wymianie na osobniki utworzone z górnej połowy populacji na drodze krzyżowania i mutacji. Populacja początkowa tworzona jest losowo. Przykładowy standardowy proces to 10 populacji prowadzonych przez ok. 1000 generacji każda. Redundancja ta wynika ze stwierdzonego empirycznie faktu, iż losowość pierwszej generacji powoduje start z materiałem genetycznym, który niekoniecznie zawiera geny potrzebne do wytworzenia optymalnego modułu – stąd intuicyjna strategia prowadzenia kilku hodowli jednocześnie.

Metoda o roboczej nazwie QuickBall zakłada redukcję czasu hodowli przy jednoczesnym zwiększeniu liczby populacji, a więc i zwiększeniu rozpatrywanego w każdej generacji obszaru przestrzeni rozwiązań. Przykładowy proces z użyciem metody QuickBall wygląda następująco: tworzonych jest 48 niezależnych populacji ewoluowanych przez 10 generacji każda. Następnie do dalszej hodowli wybieranych jest 10 najlepszych. Jeśli po kolejnych 40 generacjach nie uzyskano jeszcze wystarczająco dobrych rozwiązań, to 2 najlepsze populacje ewoluowane są przez dalszych 60 generacji każda. W ten sposób wymagana moc obliczeniowa zredukowana jest dziesięciokrotnie (1000 generacji w miejsce dotychczasowych 10000) przy jednoczesnym wzroście efektywności. Prawie 5-krotnie większy obszar rozwiązań rozpatry-

wanych w każdej generacji zwiększa szansę znalezienia rozwiązania bliskiego optimum. W dotychczasowych doświadczeniach już po 50 generacjach otrzymywano nawet kilka rozwiązań o stopniu dopasowania rzędu 90%, podczas gdy przy użyciu standardowej metody często nie otrzymywano ani jednego.

Celem przedstawienia szczegółów metody QuickBall należy najpierw wprowadzić kilka nowych pojęć. Dotychczasowo każdą hodowlę identyfikowano za pomocą parametru MaxVal, opisującego stopień przystosowania najlepszego w danej generacji osobnika. Jak już wspomniano, algorytm zakłada przejście 20% najlepszych osobników do kolejnej generacji w stanie nienaruszonym (z pominięciem krzyżowania i mutacji). Funkcja MaxVal(t) jest więc funkcją niemalejącą. Charakter tej funkcji zbliżony jest do wykładniczego, przy czym punkt, powyżej którego wykres wchodzi w obszar plateau, przypada na ok. 100-110 generacji, tak więc obszar najszybszego wzrostu można ograniczyć do przedziału [0;50] generacji. Jednak można zaobserwować zarówno populacje, których wykres MaxVal(t) zaczyna się dosyć wysoko, lecz jest stosunkowo płaski, jak i takie, których wykres startuje stosunkowo nisko, lecz rośnie bardzo stromo, osiągając po np. 50 generacjach znacznie lepsze rezultaty. Celem wyjaśnienia tego zjawiska wprowadzono drugi parametr opisujący populację – dynamikę wzrostu. Parametr ten, oznaczany dalej S1, jest współczynnikiem nachylenia prostej przybliżającej pierwszych 10 generacji wykresu MaxVal(t):

$$S1 = \frac{\text{MaxVal}(10) - \text{MaxVal}(1)}{10}$$

W dotychczasowych eksperymentach stwierdzono, iż wybierając po 10 generacjach z 10 populacji o największych wartościach S1 trafia się 4 do 6 z 10 o największych wartościach MaxVal(50).

Parametr S1 jest więc próbą opisanie potencjału populacji, podobnie jak MaxVal opisuje poziom. Jednak o ile poziom jest prostą funkcją przyjmującą w każdej generacji t wartość przystosowania najlepszego osobnika, o tyle potencjał populacji jest wielkością złożoną – na wartość potencjału wpływ mają zarówno ustawienia symulatora (modyfikujące działanie algorytmu oraz strukturę rozwiązań), jak i kształt obszaru przestrzeni rozwiązań objętego daną populacją.

Jako przykładowy materiał do analizy wzięto wyniki dwu identycznych eksperymentów: at002 i at005. W obu przypadkach 48 populacji hodowano przez 50 generacji, przy stałych ustawieniach symulatora, przy czym dla jednego z parametrów przyjęto 2 nastawy – po 24



populacje z eksperymentu na nastawę. Na etapie analizy pogrupowano populacje o identycznych nastawach, tworząc dwa wirtualne eksperymenty – at00(2-5)a i at00(2-5)b.

Na wykresach rozkładów  $\text{MaxVal}(t)$  eksperymentów at002 i at005 wyraźnie widać dwa ciągle w czasie lokalne maksima zbliżone do funkcji eksponencjalnej. Porównanie z wykresami dla eksperymentów wirtualnych, poparte analizą porównawczą wykresów rozkładów S1, wykazało, iż dwa maksima odpowiadają dwu nastawom symulatora. Populacje zgrupowane w at00(2-5)a charakteryzują się zdecydowanie wyższą dynamiką S1, a co za tym idzie osiągają wyższe wartości  $\text{MaxVal}(t)$ , ze szczególnym uwzględnieniem  $\text{MaxVal}(50)$ . Stosowne wykresy są na dużym obszarze praktycznie rozłączne, co potwierdza tezę, iż analizując po 10 generacjach dynamikę S1 można wybrać optymalne ustawienia symulatora i stosowne populacje rozwijać dalej – w tym przypadku wyższy potencjał populacji zgrupowanych w at00(2-5)a uwidocznił się poprzez obsadzenie przez te populacje górnej połowy wspólnego wykresu tak rozkładu S1, jak i rozkładu  $\text{MaxVal}(50)$ . Również istotny jest fakt, iż obserwacje poczynione dla eksperymentów wirtualnych, a więc o jednakowych nastawach symulatora, prawdziwe są również dla oryginalnych eksperymentów, gdzie mamy do czynienia z wymieszaniem procesów o różnych nastawach.

Kolejny krok stanowi analiza wkładu w potencjał populacji drugiego czynnika – kształtu obszaru przestrzeni rozwiązań obejmowanego przez daną populację. Tu analiza jest o tyle trudniejsza, że następuje wymiana informacji w obrębie populacji. Efekt widać na wykresach śladów czasowych poszczególnych populacji – o ile można dość łatwo rozdzielić je na dwa pasma odpowiadające eksperymentom at00(2-5)a i at00(2-5)b, o tyle w ramach pojedynczego pasma powtórzenie operacji rozdzielania jest niemożliwe. Z pomocą przychodzą tablice EPT (EPT - Error Prediction Table) umożliwiające oszacowanie liczby populacji, których proces ewolucji należy kontynuować na etapie 10-50 generacji, aby otrzymać zadaną liczbę modułów o wysokim współczynniku  $\text{MaxVal}(50)$ . Wprawdzie tablice te stworzone zostały na podstawie danych z konkretnych eksperymentów, jednak wahania dokładności dla innych eksperymentów nie powinny przekroczyć 10-20 %, co jest zupełnie wystarczające. Można też w przypadku eksperymentu na szerszą skalę stworzyć dokładną tablicę dla reprezentatywnej próby i na jej podstawie analizować całą grupę populacji.

Podsumowując, metoda QuickBall składa się z dwu etapów: analizy widmowej dynamiki S1 oraz przewidywania błędów z wykorzystaniem tablic EPT. Przykładowy eksperyment może więc wyglądać tak:

1. utworzyć grupy populacji zróżnicowane pod względem ustawień symulatora,
2. po 10 generacjach obliczyć dla każdej populacji współczynnik dynamiki S1,

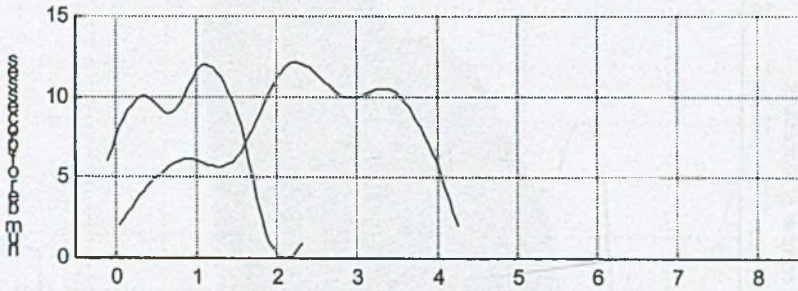
3. utworzyć rozkłady współczynnika  $S_1$  dla poszczególnych grup i wybrać najkorzystniejszy (najkorzystniejsze),
4. z wybranej grupy (grup) wyslekcjonować jednostki za pomocą tablic EPT,
5. wybrane populacje poddać dalszej ewolucji do osiągnięcia pułapu 50 generacji,
6. w przypadku niesatysfakcjonującej wartości bezwzględnej  $MaxVal(50)$  wybrać kilka najlepszych populacji jako kryterium, stosując wartość  $MaxVal(50)$ ,
7. wybrane jednostki poddać dalszej ewolucji do osiągnięcia pułapu ok. 100-120 generacji.

Technologia ta, opracowana przez autora na podstawie danych z losowo wybranego eksperymentu EEC „Atlantis”, pozwala na skrócenie czasu trwania ewolucji kilkukrotnie, w przypadku standardowej strategii hodowli stosowanej w GABRI redukcja ta jest nawet dziesięciokrotna. Co więcej, przy okazji udowadniania tej teorii stworzone zostały narzędzia pozwalające na określenie liczby równoległych hodowli niezbędnych do otrzymania określonej ilości modułów o wysokim stopniu przystosowania, jak również szacowania ilości takich modułów otrzymanych po wyewoluowaniu określonej liczby populacji przez określony czas.

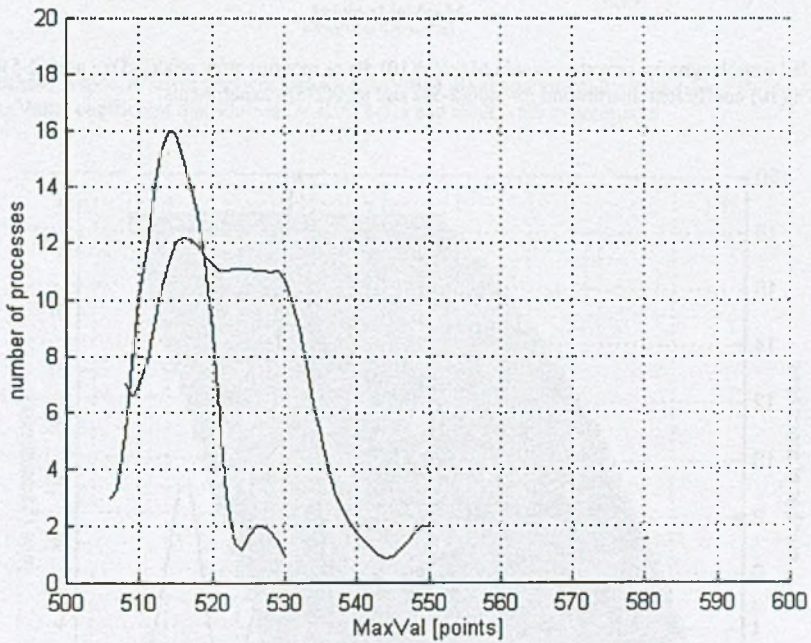
## 5. Perspektywy na przyszłość

Planowane są zarówno prace nad nowymi zagadnieniami, jak i nad kontynuacją dotychczasowych badań. Po pierwsze, metoda QuickBall może być jeszcze znacznie ulepszona, zwłaszcza technika tablic EPT. Po drugie, rozpoczęto już prace nad stworzeniem nowej klasy symulatora, z implementacją najnowszych osiągnięć zespołów EEC. Po trzecie, nowy symulator pozwoli przypuszczalnie zakończyć eksperyment opóźniacza, co mogłoby otworzyć drogę do znacznie bardziej skomplikowanych zagadnień: rozpoznawanie obrazów, układy pamięci...

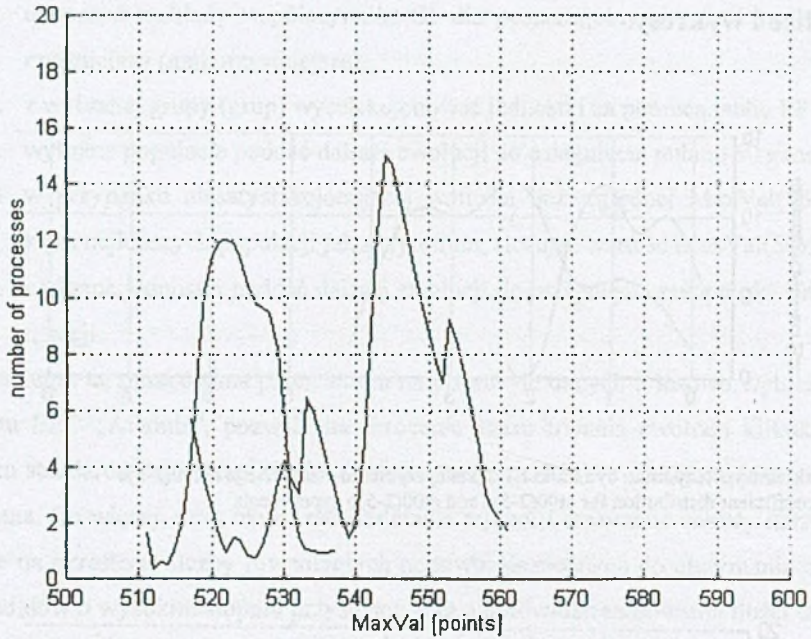
## 6. Tablice i wykresy



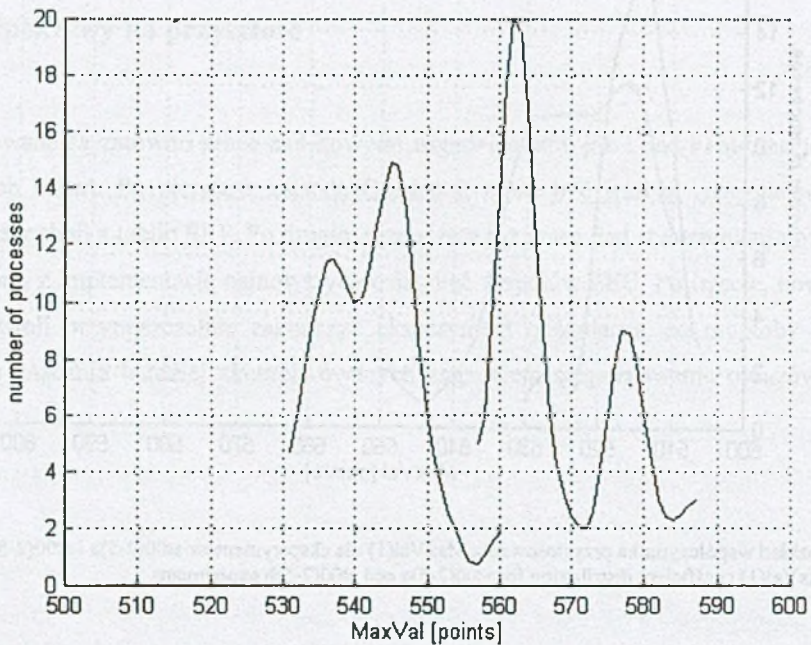
Rys.1. Rozkład współczynnika dynamiki S1 dla eksperymentów at00(2-5)a i at00(2-5)b  
 Fig.1. S1 coefficient distribution for at00(2-5)a and at00(2-5)b experiments



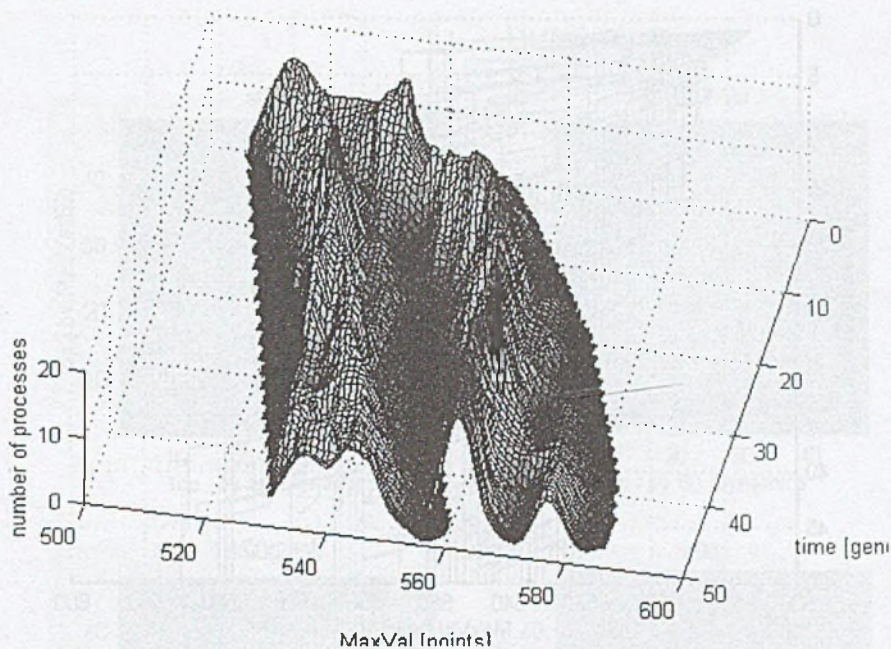
Rys.2. Rozkład współczynnika przystosowania MaxVal(1) dla eksperymentów at00(2-5)a i at00(2-5)b  
 Fig.2. MaxVal(1) coefficient distribution for at00(2-5)a and at00(2-5)b experiments



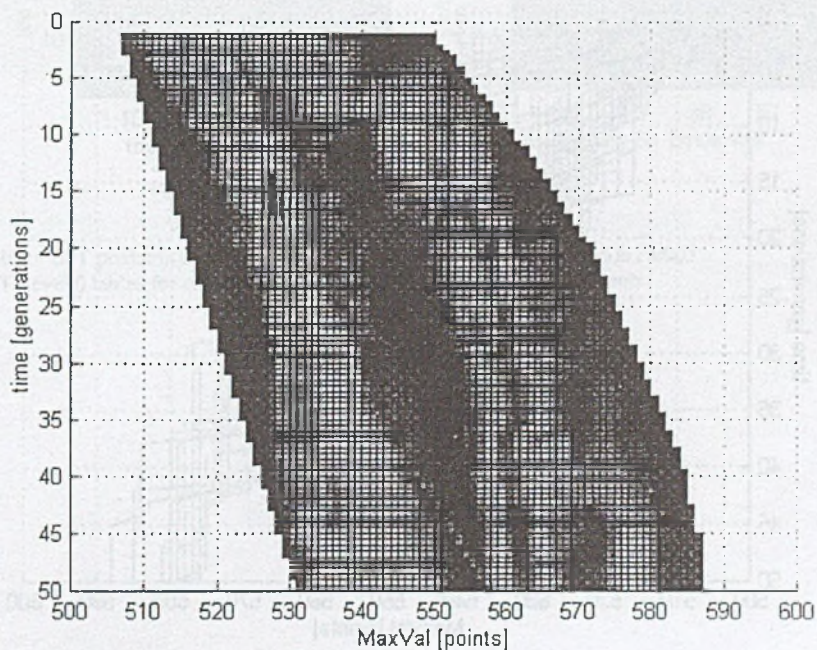
Rys.3. Rozkład współczynnika przystosowania MaxVal(10) dla eksperymentów at00(2-5)a i at00(2-5)b  
 Fig.3. MaxVal(10) coefficient distribution for at00(2-5)a and at00(2-5)b experiments



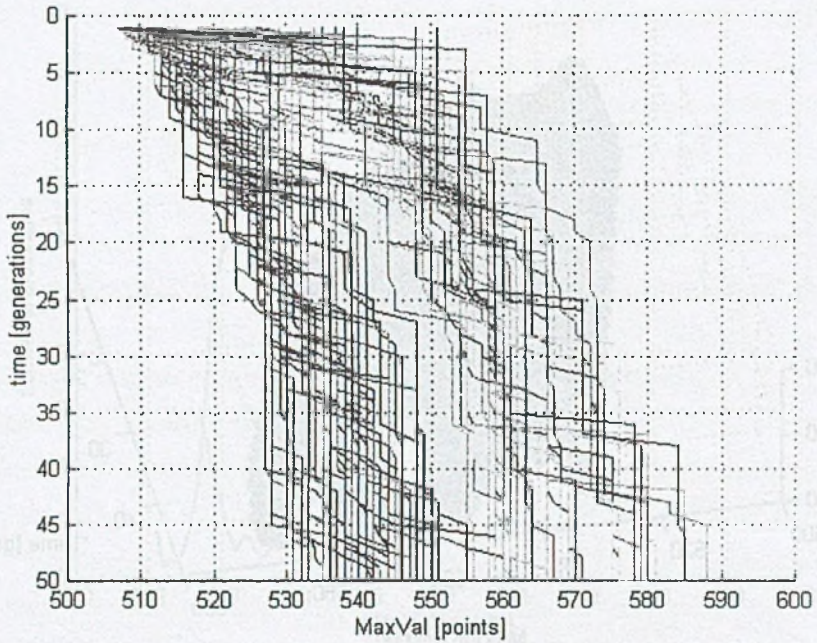
Rys.4. Rozkład współczynnika przystosowania MaxVal(50) dla eksperymentów at00(2-5)a i at00(2-5)b  
 Fig.4. MaxVal(50) coefficient distribution for at00(2-5)a and at00(2-5)b experiments



Rys.5. Rozkład współczynnika przystosowania MaxVal(t) dla eksperymentów at00(2-5)a i at00(2-5)b  
 Fig.5. MaxVal(t) coefficient distribution for at00(2-5)a and at00(2-5)b experiments

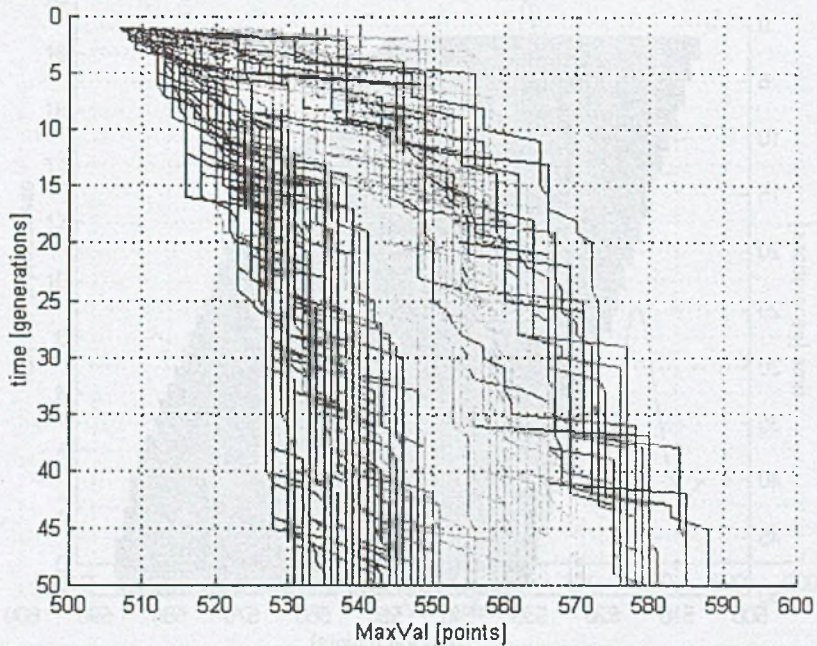


Rys.6. Rozkład współczynnika przystosowania MaxVal(t) dla eksperymentów at00(2-5)a i at00(2-5)b  
 Fig.6. MaxVal(t) coefficient distribution for at00(2-5)a and at00(2-5)b experiments



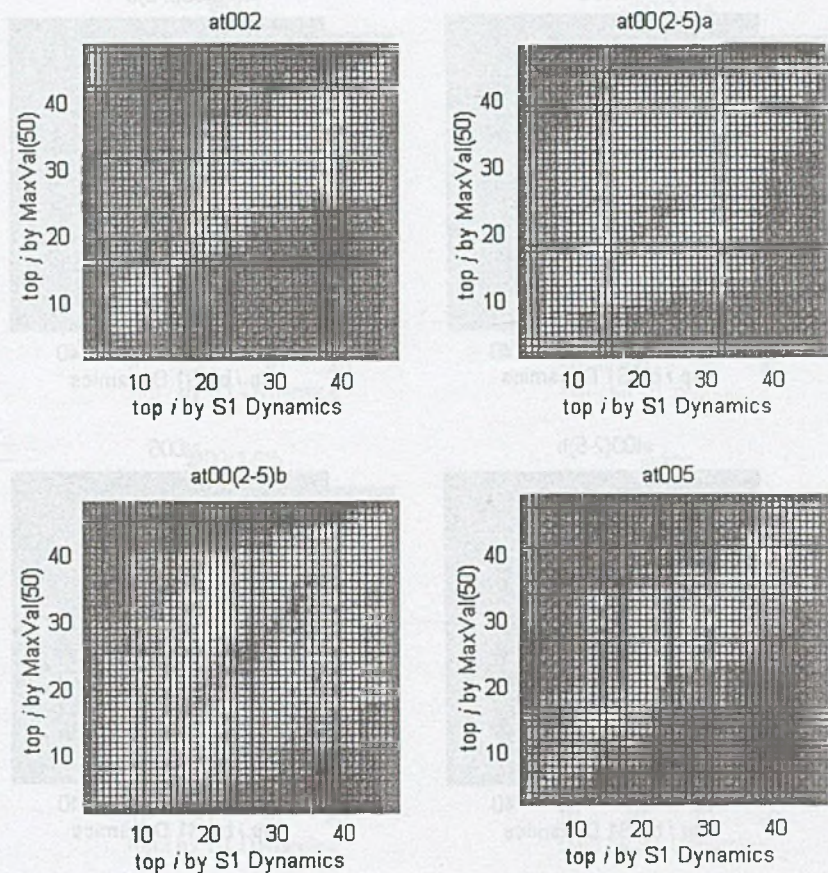
Rys.7. Ślady czasowe przystosowania  $\text{MaxVal}(t)$  dla poszczególnych procesów eksperymentów at00(2-5)a i at00(2-5)b

Fig.7.  $\text{MaxVal}(t)$  coefficient time trails for particular processes of at00(2-5)a and at00(2-5)b experiments

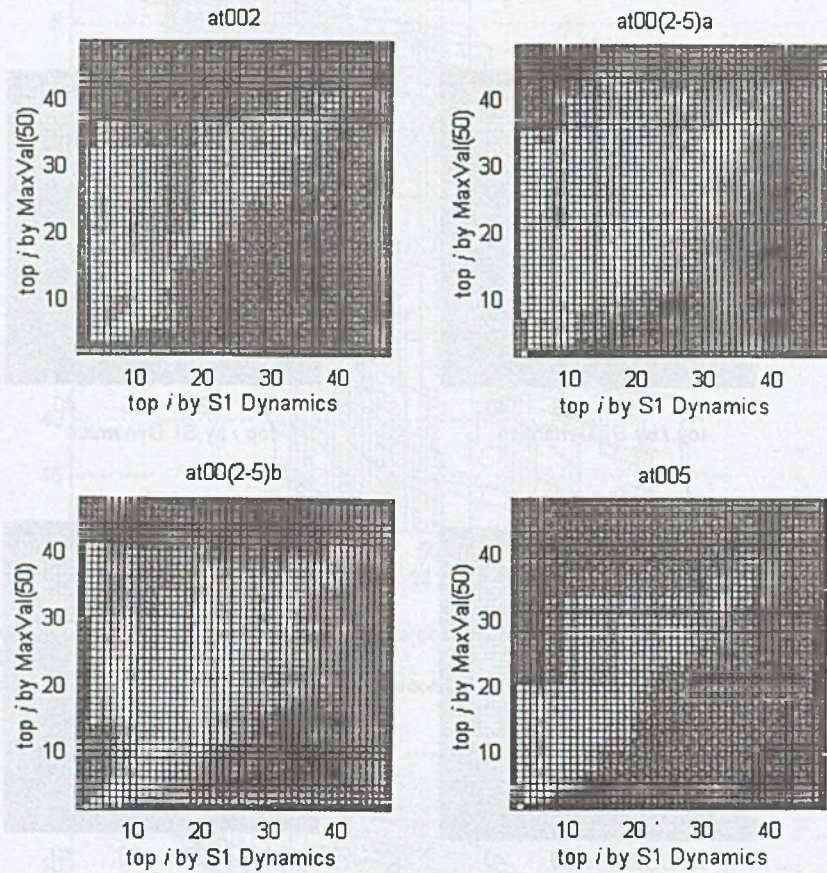


Rys.8. Ślady czasowe przystosowania  $\text{MaxVal}(t)$  dla poszczególnych procesów eksperymentów at00(2-5)a i at00(2-5)b

Fig.8.  $\text{MaxVal}(t)$  coefficient time trails for particular processes of at00(2-5)a and at00(2-5)b experiments



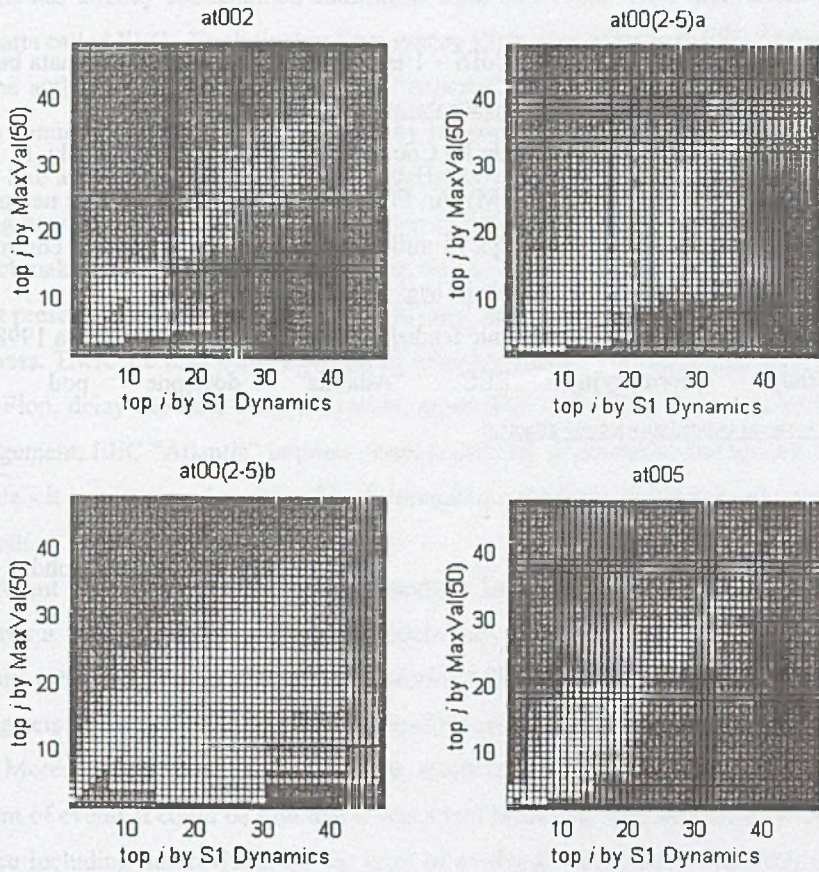
Rys.9. Tablice EPT poziomu 0 dla eksperymentów at002, at00(2-5)a, at00(2-5)b i at005  
Fig.9. EPT Level 0 tables for at002, at00(2-5)a, at00(2-5)b and at005 experiments



Rys.10. Tablice EPT poziomu 1 dla współczynnika dynamiki S1 dla eksperymentów at002, at00(2-5)a, at00(2-5)b i at005

Fig.10. EPT Level 1 tables for S1 Dynamics coefficient for at002, at00(2-5)a, at00(2-5)b and at005 experiments





Rys.11. Tablice EPT poziomu 1 dla współczynnika przystosowania MaxVal(50) dla eksperymentów at002, at00(2-5)a, at00(2-5)b i at005

Fig.11. EPT Level 1 tables for MaxVal(50) coefficient for at002, at00(2-5)a, at00(2-5)b and at005 experiments

## LITERATURA

1. Gers F., de Garis H., Korkin M.: CoDi – 1 Bit: A simplified cellular automata based neuron model. Available on website <http://foobar.starlab.net/~degaris/>.
2. de Garis H., Korkin M., Guttikonda P., Cooley D., Buller A., Chodakowski T., Zieliński G.: The CAM-Brain Machine (CBM). An FPGA based tool which evolves neural net circuit models in seconds for building a 75 million neuron artificial brain that controls a life-sized kitten robot. Available on website <http://foobar.starlab.net/~degaris/>.
3. Buller A.: Sztuczny mózg – to już nie fantazje. Prószyński i S-ka. Warszawa 1998.
4. Materiały informacyjne EEC “Atlantis” dostępne pod adresem <http://www.zts.ia.polsl.gliwice.pl/~atlantis/>.

Recenzent: Doc. dr inż. Bohdan Wołczak

Wpłynęło do Redakcji 21 czerwca 2000 r.

**Abstract**

The elements of artificial intelligence have been used in questions such as controlling, picture identification and making decisions assistance for ages. Although, in spite of advanced progress of these works, it is still far from getting “the real” artificial intellect. In case of the most popular method of AI building – that bases on neural networks – the biggest problem is the violent increase of complications network designing and teaching phases with increasing number of neurons.

The new approach to the problem, proposed by ATR Institute’s group of scientists in charge of dr. Hugo de Garis, is to combine the three ranges: the neural networks, genetic algorithms and cellular automata.

This method makes real operating on structures containing hundred billions of neurons. In the project of building this high-leveled artificial intelligence take part groups from 9 countries. One of them is polish group created in 1998 by Andrzej Buller from Technical Univer-

sity of Gdansk – GABRI (Gdansk Artificial Brain Research Initiative). In the middle of 1999, GABRI has already concentrated enthusiasts from all Poland. Then there came the division into parts called EEC - Evolutionary Engineering Club. One of them is EEC “Atlantis”.

The author of the article is the EEC “Atlantis” coordinator. Members of this dispersed group communicate with the coordinator by Internet. Firstly, the collaboration with the club looks like another disperse project: SETI@Home: members receive simulator program containing service manual and exemplary experiment by e-mail. Gradually, they hollow into the Project making more and more independent works.

At present, the works made by EEC “Atlantis” are related to inquires about CoDi modules' rightness. There are also made attempts of modules' rearing that can make specific functions (Flip-Flop, delay module, etc.). Moreover, apart from tasks, that were ordered by the main management, EEC “Atlantis” inquires about possibility of accelerate the process of obtaining module - it is necessary because of high computational powers indispensable in using former methods.

Present works are related to specific modules having both digital output and input. As an example a Flip-Flop (electronic R-S) module may be used. Despite using only 10 training sets, the module replies to sets unused in rearing process very well (the average adaptation for testing sets amounts 95%). First of all, Flip-Flop module prospects to creating memory modules. Moreover, the module corrected its teacher's error. If that feature revealed in more amount of events it could be said that it was a real breakthrough. The structures of artificial intelligence including autorevising on the level of evolving will create completely new class in this range.

The QuickBall technology, elaborated on the grounds of evidence of EEC “Atlantis” by chance chosen experiment, allows to shorten few times the time of evolution. In case of standard strategy of growth used in GABRI, the reduction is even more than 10 times. More, in occasion of proving this theory some tools were created. These tools allow us to estimate the number of parallel evolved populations necessary to obtain the concrete amount of modules of a high level of accommodation. They also help us to estimate the amount of those modules obtained after evolution of the concrete number of populations in concrete time.

The plans for future are wide. Firstly, the QuickBall method can be much more improved, especially the EPT tables technology. Secondly, the works at creating the new kind of simulator with the implementation the newest achievements of EEC groups began. Thirdly, the new simulator allows to finish the delay module experiment and shows the way to much more complicated problems: recognizing pictures, the memory units...