

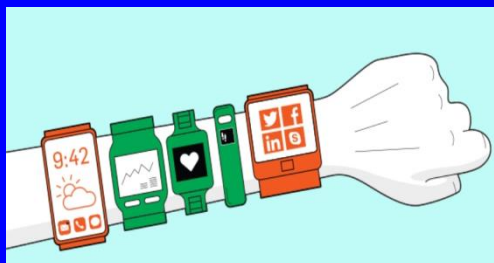


Seminarium Zakład Strukturalnych Metod Przetwarzania Wiedzy

Warszawa, 26 listopada 2015



Inteligentne Systemy Rozproszone



dr hab. inż. Jerzy Balicki, prof. nzw. PW

*Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych*

Plan

- 1. Wybrane tendencje w AI i obliczeniach rozproszonych;**
- 2. Inteligentne metody w projektowaniu systemów rozproszonych;**
- 3. Wybrane zastosowania równoległych metod sztucznej inteligencji.**



Chatterbot Eugene Goostman



- Imituje odpowiedzi 13-letniego ukraińskiego nastolatka z Odessy;
- **Zdał test Turinga w lipcu 2014 roku;**
- 33% odpowiedzi zostało uznanych za odpowiedzi człowieka, a nie maszyny;

Scott: Which is bigger, a shoebox or Mount Everest?

Eugene: *I can't make a choice right now. I should think it out later. And I forgot to ask you where you are from...*

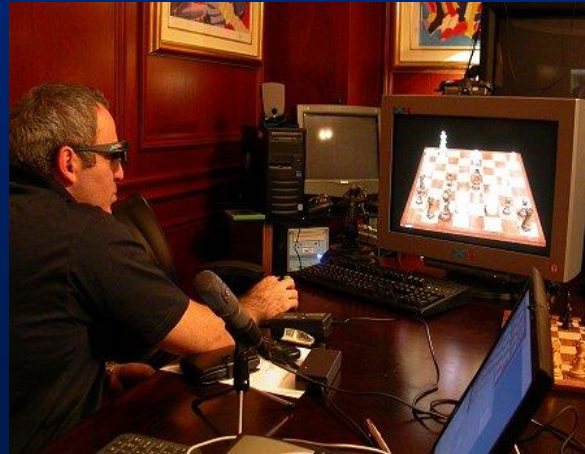
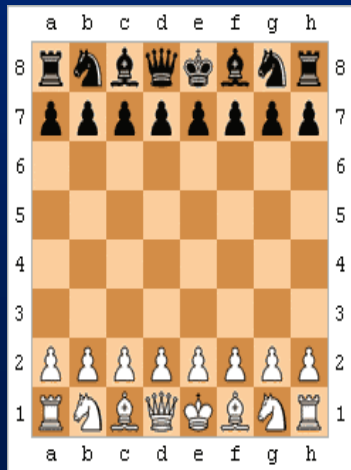
Scott: How many legs does a camel have?

Eugene: *Something between 2 and 4. Maybe, three? :-)))*

By the way, I still don't know your specialty – or, possibly, I've missed it?

Programy do gry w szachy

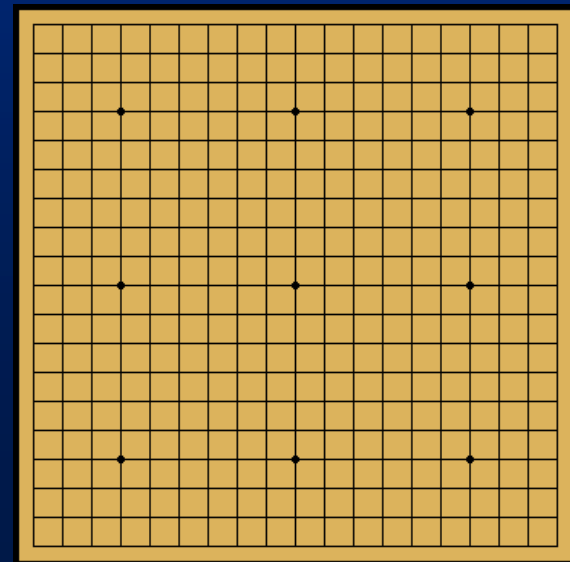
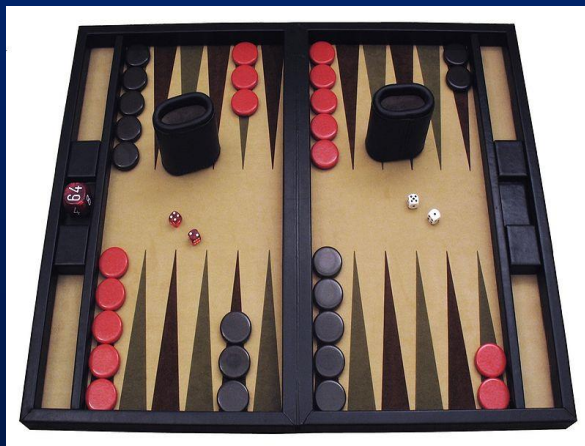
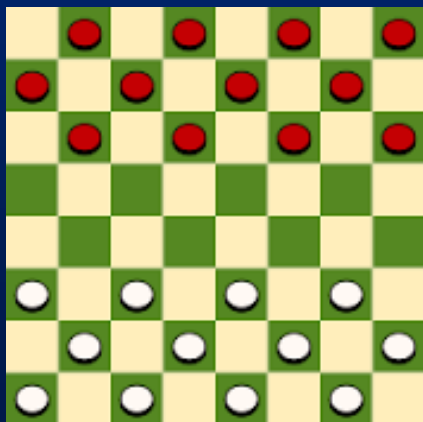
- 1959 - pierwszy program do gry w szachy (Samuel –1959);
- 1997 – **Deep Blue** na superkomputerze (32 procesory IBM RS6000/SP2 + 256 ASIC) wygrał z mistrzem świata G. Kasparowem 3.5:2.5;
- 2002 – program **Deep Fritz** na PC remisuje z V. Kramnikiem;
- 2003 – remis Kasparowa z X3D **Deep Fritz junior** (100 razy wolniejszy od Deep Blue, wirtualna rzeczywistość);
- 2015 – nikt nie wygra z programem **Rybka** w smartfonie;
- <http://rybkachess.com/> - **Rybka** to najsilniejszy silnik gry w szachy;
- komercyjna wersja 4 gra z niedostępną dla człowieka siłą **3000 elo**.



Szacowana liczba sytuacji na szachownicy 10^{123}

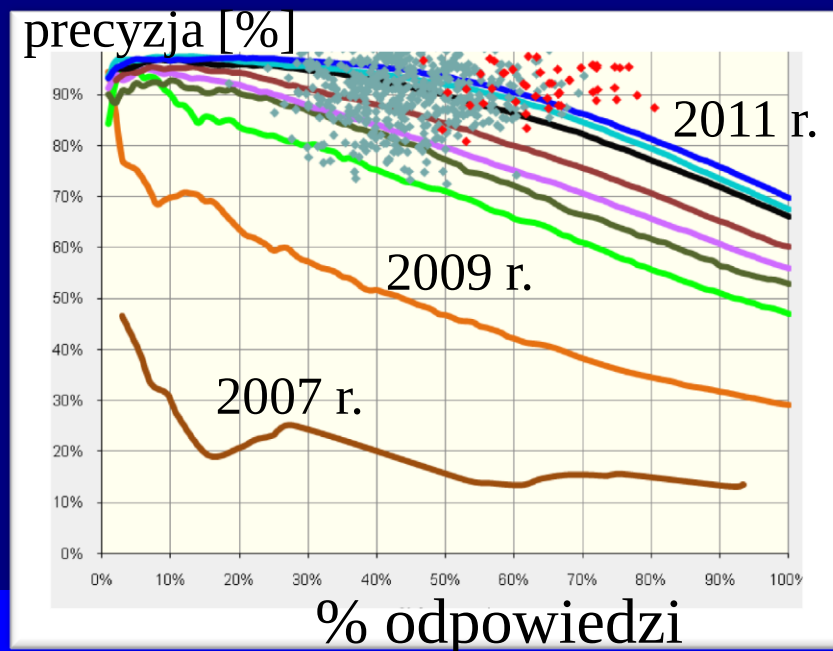
Warcaby, Tryktrak i Go

- 1952 - Samuel, pierwszy program uczący się gry w warcaby;
- 1995 - wygrana programu **Chinook** z mistrzem świata w **warcaby** 1-0 i 31 remisów. Komputera SGI (8 procesorów/512 MB RAM);
- 1995 - program **Logistello** zwycięża mistrza świata w **tryktraka** Takeshi Murakami 6:0;
- 2015 – programy nie mogą pokonać człowieka - mistrza świata w grę Go (tylko dla siatki 9x9, ale nie 13x13 czy 19x19);
- Liczba ruchów w Go to średnio 150-250 (szachy tylko 37);
- Liczba możliwych partii: 10^{761} (szachy 10^{123});
- Tianhe-2 potrzebuje 4 godz. do wyboru strategii na 8 ruchów w przód.



IBM Watson

- 14 stycznia 2011 roku wygrywa quiz *Jeopardy* (w Polsce teleturniej *Va Banque*).



Emily Dickinson		99%	2011
Walt Whitman		60%	
Barnard		10%	

IBM Watson

- Klaster z 90 CPU IBM Power 750;
- W CPU 360 procesorów IBM PowerPC Power7;
- 8 rdzeni w każdym procesorze 3.5 GHz;
- Do 4 równoległych wątków na rdzeń;
- 16 TB RAM ;
- ~ 80 TeraFLOPs ;
- Cena \$3,000,000 .

IBM Watson – potencjalne zastosowania

- **Analiza biznesowa;**
- **Wyszukiwanie w rozległych bazach wiedzy;**
- **Pomoc w podejmowaniu decyzji;**
- **Diagnostyka medyczna;**
- **Edukacja.**

● Sztuczna inteligencja a superkomputery ●

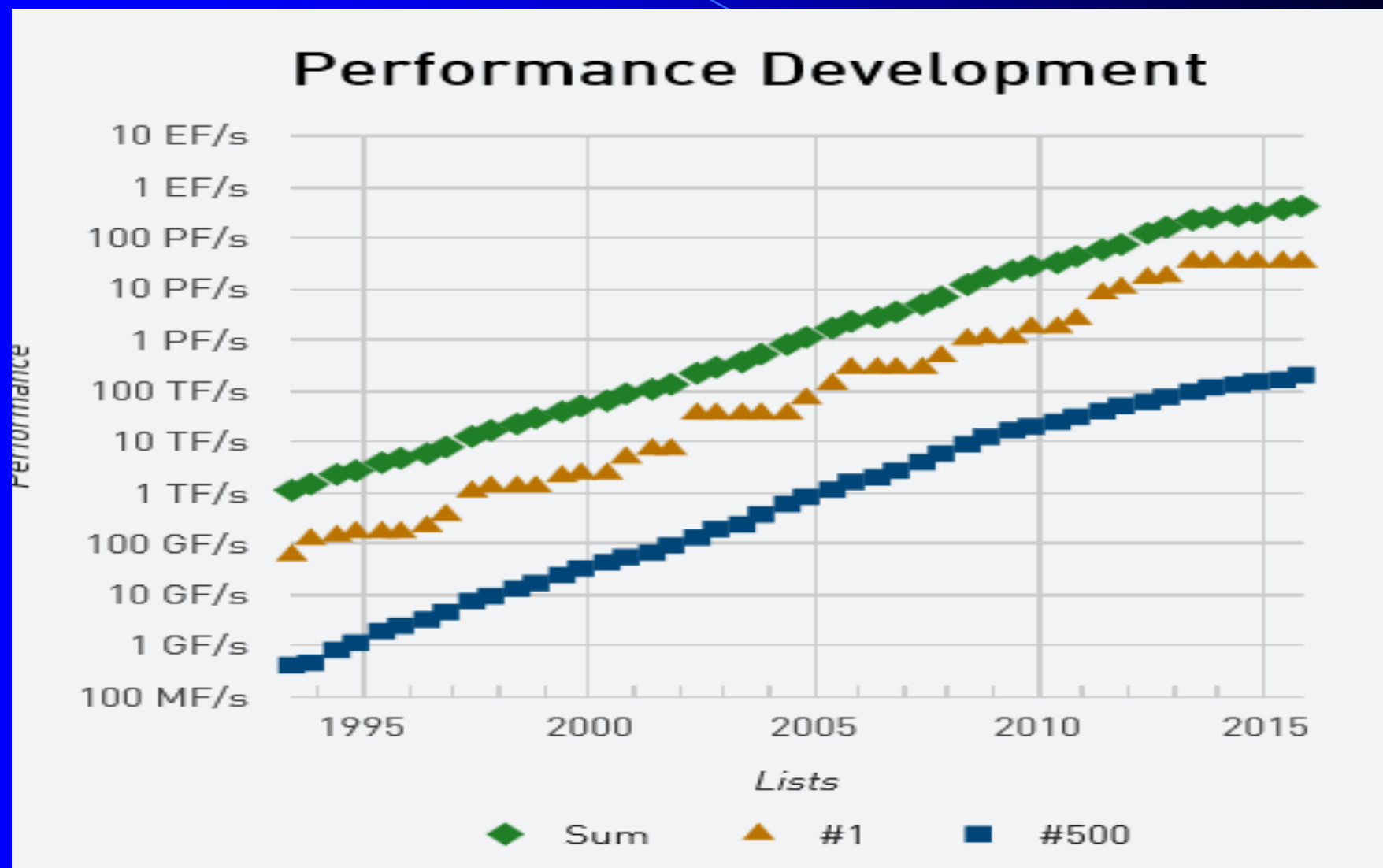
“

By 2018 half of all consumers will regularly interact with services based on cognitive

”

- IDC FutureScape

Moc obliczeniowa superkomputerów podwaja się co dwa lata



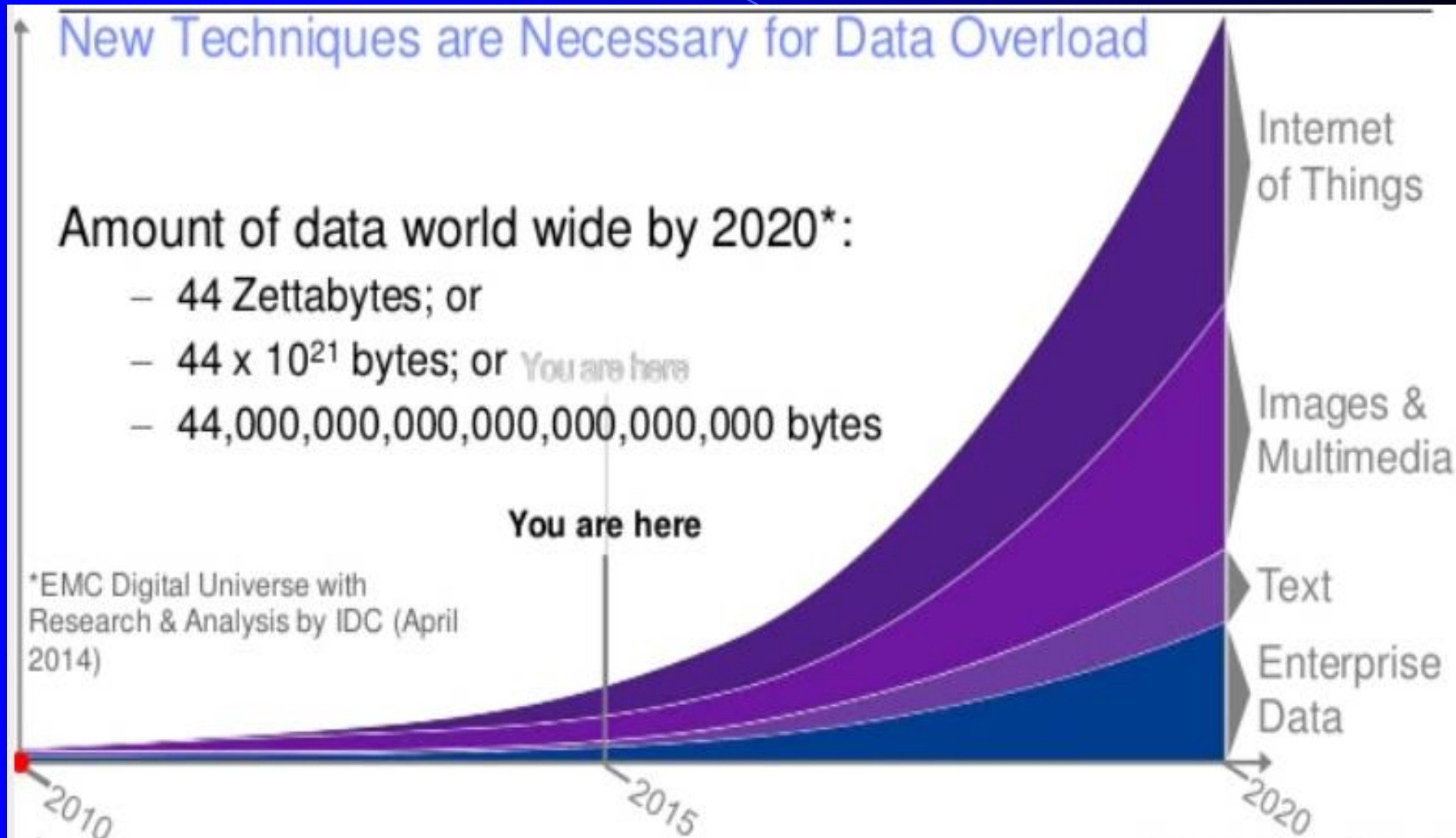
Tianhe-2 (MilkyWay-2)

33,8 PF/s; 17,8 MW; 5 ton



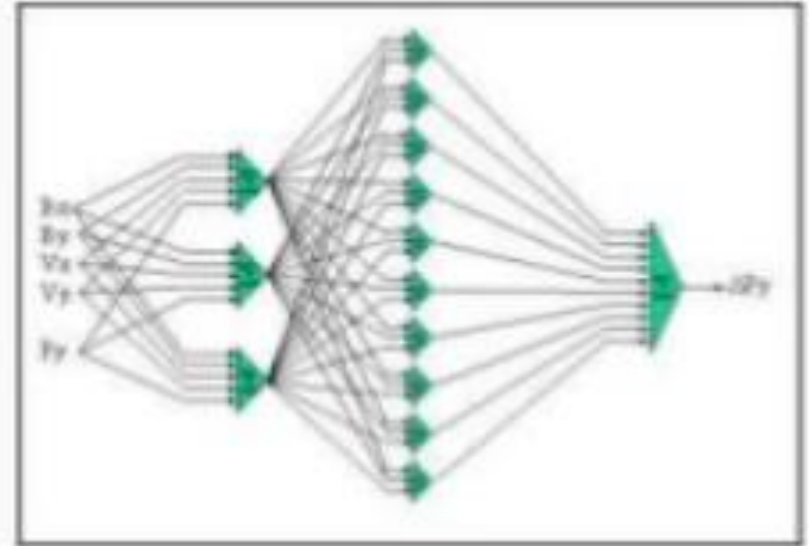
Procesory	32 000 Intel Xeon CPUs, 48 000 Intel Xeon Phis, 4 096 FT CPUs, (3 120 000 rdzeni)
RAM	1 PB
Sieć	TH Express 2 (160 Gb/s)
HDD	12,4 PB (256 węzłów I/O, 65 serwerów HDD)
Szafy	170

Wykładniczy wzrost intensywności przetwarzania danych



Rozwiązanie wg IBM: systemy uczące

```
000100 IDENTIFICATION DIVISION.  
000200 PROGRAM-ID.      HELLOWORLD.  
000300  
000400*  
000500 ENVIRONMENT DIVISION.  
000600 CONFIGURATION SECTION.  
000700 SOURCE-COMPUTER.  RM-COBOL.  
000800 OBJECT-COMPUTER.  RM-COBOL.  
000900  
001000 DATA DIVISION.  
001100 FILE SECTION.  
001200  
100000 PROCEDURE DIVISION.  
100100  
100200 MAIN-LOGIC SECTION.  
100300 BEGIN.  
100400     DISPLAY " " LINE 1 POSITION 1 ERASE EOS.  
100500     DISPLAY "Hello world!" LINE 10 POSITION 10.  
100600     STOP RUN.  
100700 MAIN-LOGIC-EXIT.  
100800     EXIT.
```



1950: PROGRAMMING

- Stored data, instructions
- Languages for computing
- Metrics for computation

2011: COGNITION

- Massive data scale
- Data for training
- Real-world modalities

Serwisy projektu IBM Watson w chmurze

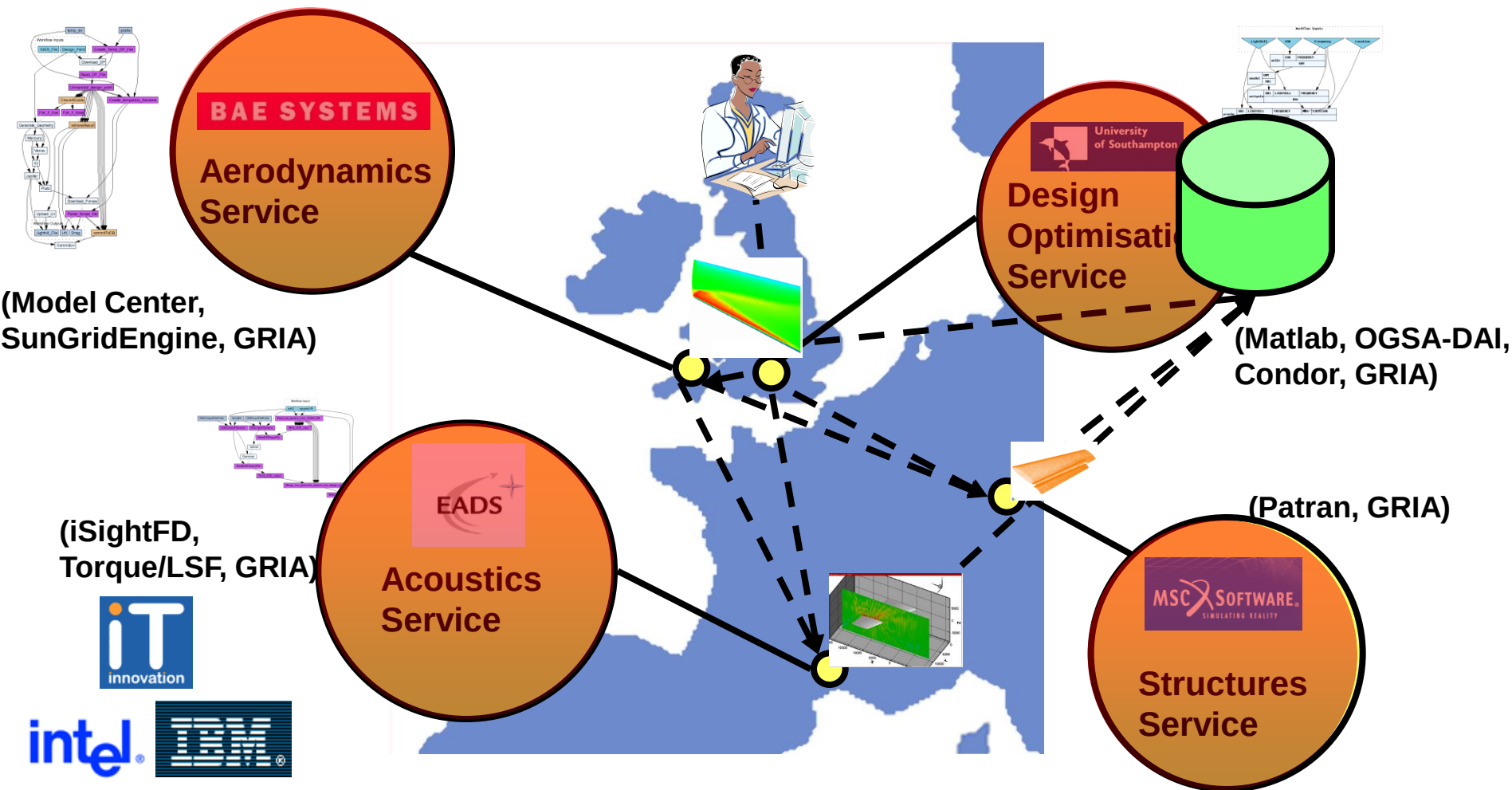
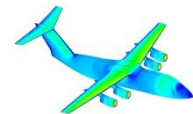
Watson Services are Cloud-Delivered

Text to Speech
Concept Expansion
Personality Insights
Tone Analyzer
Language Identification
Machine Translation
Entity Extraction
Sentiment Analysis
Message Resonance
Question and Answer
Relationship Extraction
Visualization Rendering
Concept Insights
Data News



Tradeoff Analytics
Visual Recognition
Language Detection
Text Extraction
Microformat Parsing
Feed Detection
Keyword Extraction
Linked Data Support
Image Link Extraction
Image Tagging
Face Recognition
Classification
Author Extraction
Taxonomy

Typowy scenariusz biznesowy



Scenariusz operacyjny

Chcę zarządzać dostępem dla członków mojego zespołu do usług moich dostawców



Project Manager

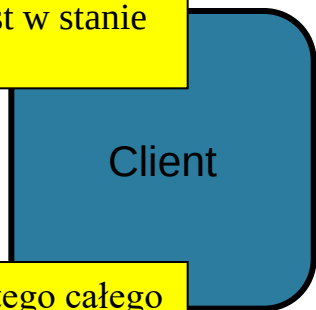
Chcę przeprowadzić kilka eksperymentów naukowych, ale mój komputer stacjonarny nie jest w stanie ich symulować



User

Nienawidzę tego całego zarządzania, gdyż przeszkadza mi w mojej pracy

Client Organisation

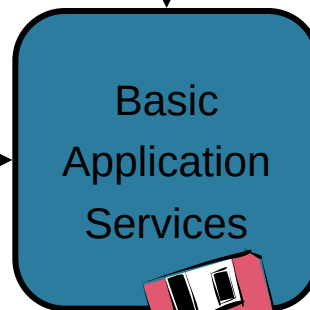
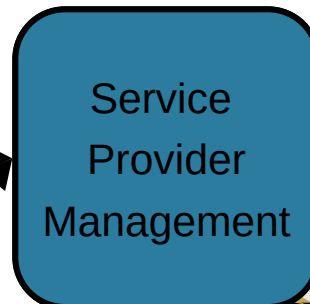


Service Level Agreement

Access Constraints



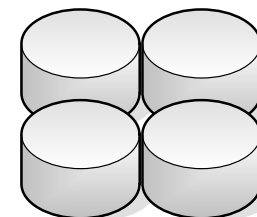
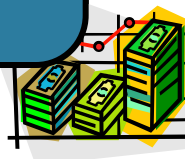
Usage Constraints



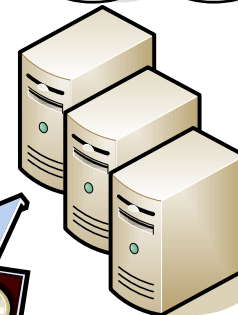
Proponuję świadczenie usług aplikacyjnych oraz określenie zasad i warunków



Administrator

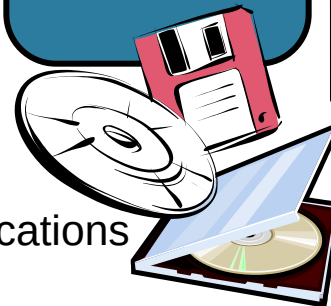


Data Storage



Data Processing

Applications

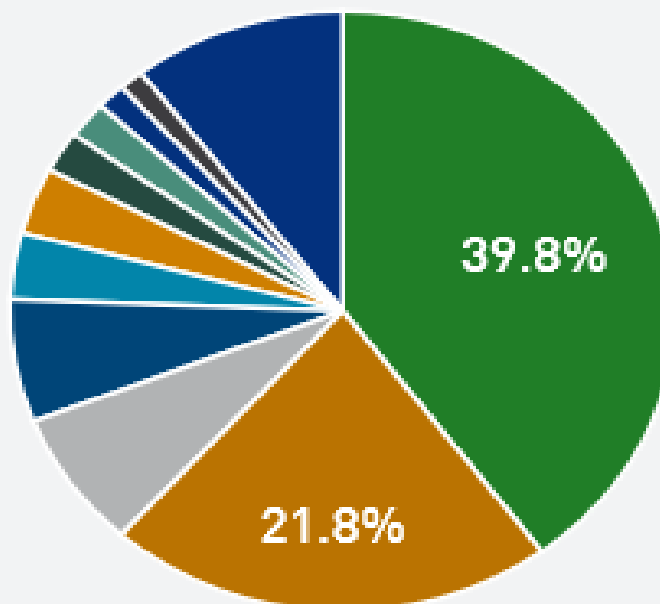


Service Provider

Państwa dysponujące najbardziej wydajnymi superkomputerami

Country System Share

- United States
- China
- Japan
- Germany
- United K...
- France
- India
- Korea, S...
- Russia
- Saudi Ar...
- Others



Wybrane zastosowania superkomputerów

- Przetwarzanie transakcji kart kredytowych;
- **Prognoza pogody;**
- Symulacja zmian klimatycznych;
- **Analiza symulacji wybuchów nuklearnych;**
- Projektowanie statków i samolotów;
- **Dynamika płynów;**
- Sekwencjonowanie ludzkiego genomu;
- **Modelowanie molekularne;**
- Kryptologia;
- **Zaawansowane animacje graficzne;**
- Analiza danych geologicznych;
- **Zarządzanie energią;**
- Poszukiwanie gazu łupkowego;
- **Astronautyka.**

Gospodarka oparta na wiedzy

Gospodarka przemysłowa

Gospodarka oparta na wiedzy

Status Quo

Dynamiczna

Robotnicy vs
Zarząd

Współpraca

Koszt

Zwrot z inwestycji

Lokalna/Krajowa

Globalna sieć

Ten sam rozmiar

Na miarę

Just in case

Just in time

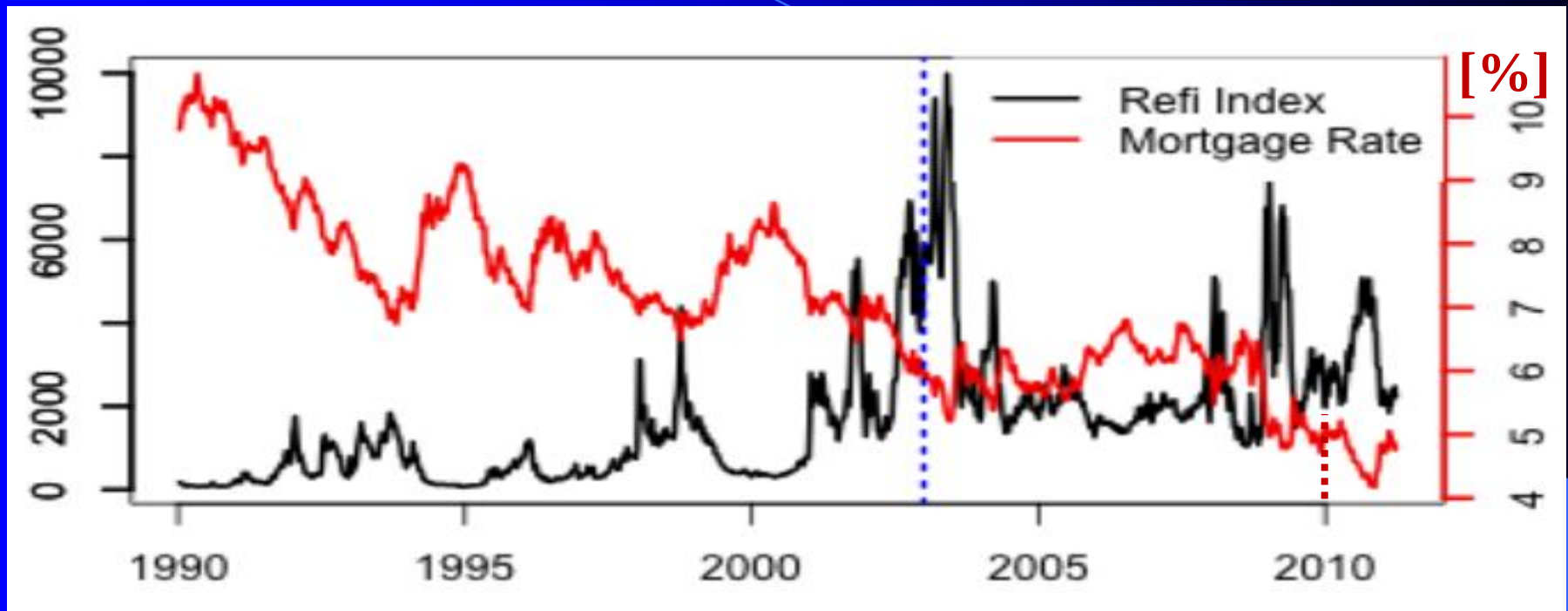
Odizolowana

Wirtualne społeczności

Studia pięcioletnie

Lifelong Learning

Indeks refinansowania kredytów $y(t)$ vs. stopa procentowa w USA



- 2003 r. – boom na nieruchomości, rekordowy spadek oprocentowania $\Rightarrow y^* = \max y(t)$
- 2010 r. – recesja na rynku nieruchomości, niesprzyjające okoliczności po kryzysie bankowym $\Rightarrow$ umiarkowany wzrost $y(t)$

Inteligentna prognoza wyszukiwarki Bing – zwycięzcy Oscarów w lutym 2015

Category	Bing prediction	Oscars winner
Best Actor	Eddie Redmayne (The Theory of Everything)	Eddie Redmayne (The Theory of Everything)
Best Actress	Julianne Moore (Still Alice)	Julianne Moore (Still Alice)
Best Supporting Actor	J.K. Simmons (Whiplash)	J.K. Simmons (Whiplash)
Best Supporting Actress	Patricia Arquette (Boyhood)	Patricia Arquette (Boyhood)
Best Animated Feature Film	How To Train Your Dragon 2	Big Hero 6
Best Documentary Feature	Citizenfour	Citizenfour
Best Foreign Language Film	Ida	Ida
Best Visual Effects	Interstellar	Interstellar
Best Picture	Birdman	Birdman
Best Director	Richard Linklater (Boyhood)	Alejandro González Iñárritu (Birdman)

86% skuteczności.
Wyszukiwarka obserwowała sieć i analizowała opinie krytyków, komentarze internautów oraz oceny przyznawane przez serwisy branżowe.

Superkomputer IBM BLUE BRAIN LAUSANNA, SWITZERLAND



- 10 000 neuronów -> 100 000 000 000;

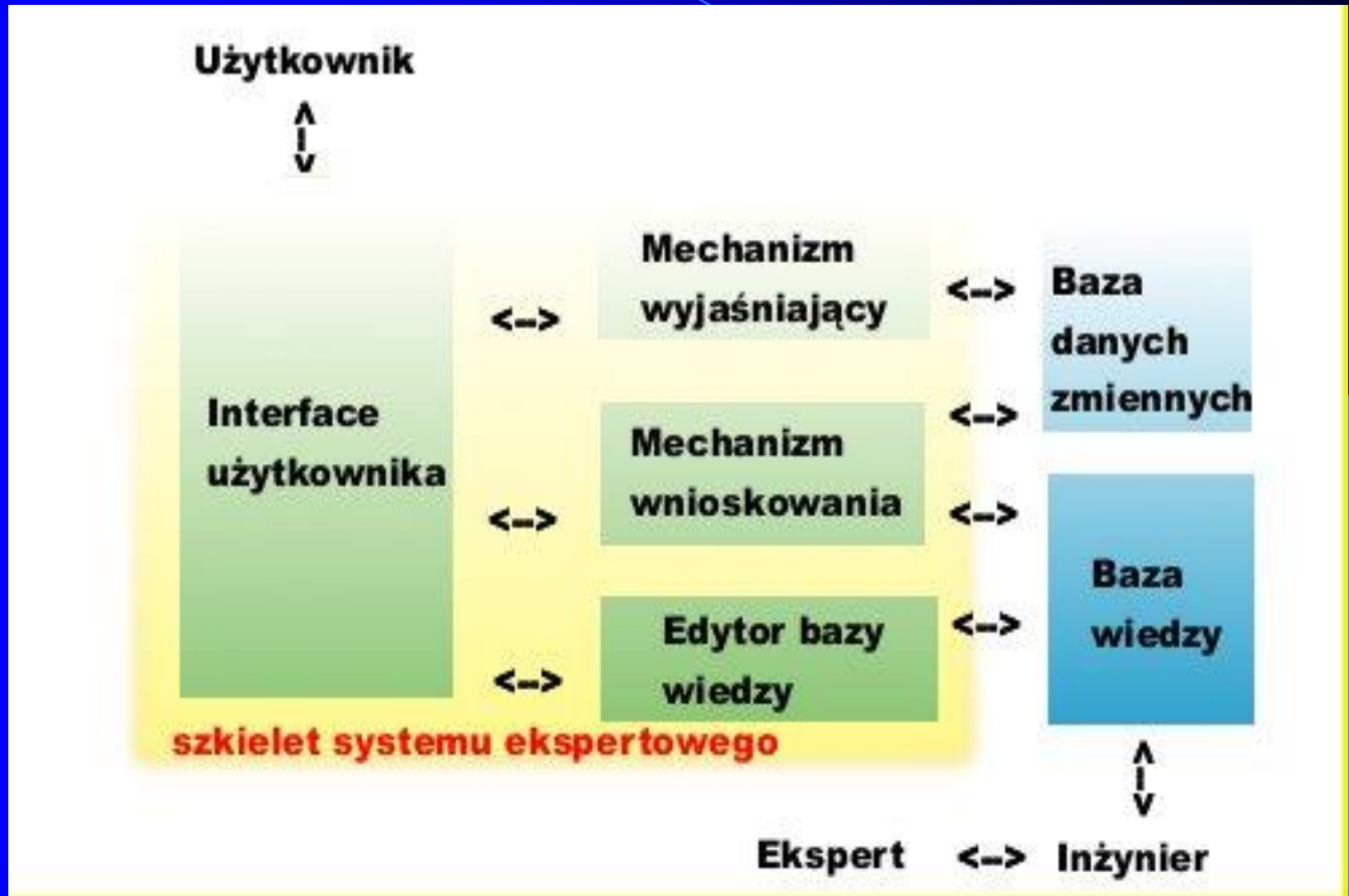
Rodzaje botów

- Pogawędki (automat dyskusyjny) – *ChatterBot*;
- Robienie zakupów internetowych – *ShoppingBot*;
- Promocja nowych produktów;
- Pomoc w nawigacji po Internecie;
- Boty pomocne nie tylko użytkownikom, ale i firmom, którym mogą dostarczać informacji o osobach odwiedzających ich strony;
- Wyszukiwanie informacji (asystent internetowy) – *DataBot*;
- Przeszukiwanie zasobów danych -*SearchBot*.

Zastosowania systemów eksperckich

- *dokonywanie wycen i kalkulacji kosztów naprawy pojazdów przez firmy ubezpieczeniowe;*
- *diagnozowanie chorób;*
- *poszukiwanie złóż minerałów;*
- *identyfikacja struktur molekularnych;*
- *udzielanie porad prawnych;*
- *diagnoza problemu (np. nieprawidłowego działania urządzenia).*

Schemat systemu eksperckiego

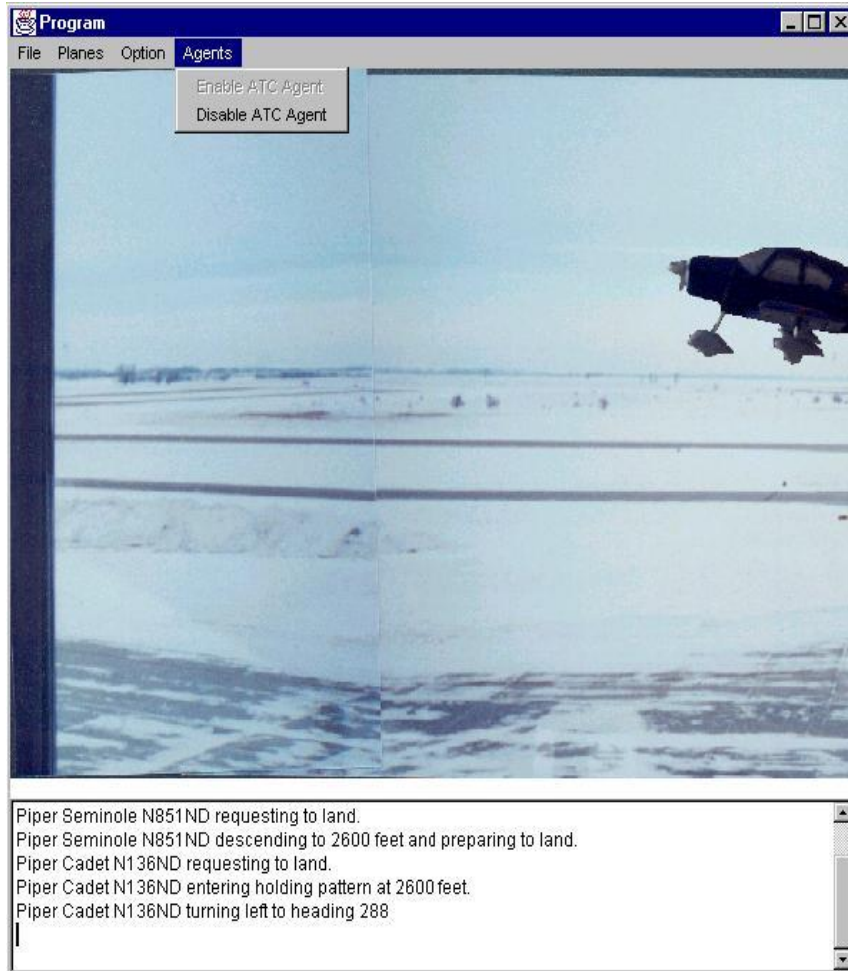


Nieinwazyjne metody odczytu danych z mózgu

- Wykrywanie emocji,
- Wykrywanie gestów,
- Sterowanie myślami, np. *mindset* lub *Emotiv EPOC*.



Rozbieżność między kontrolerem lotu a programistycznym agentem Air Traffic Control



Masdar city - niedaleko Abu Dhabi (ZEA)



Libelium Smart World

Air Pollution

Control of CO₂ emissions of factories, pollution emitted by cars and toxic gases generated in farms.

Forest Fire Detection

Monitoring of combustion gases and preemptive fire conditions to define alert zones.

Wine Quality Enhancing

Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.

Offspring Care

Control of growing conditions of the offspring in animal farms to ensure its survival and health.

Sportsmen Care

Vital signs monitoring in high performance centers and fields.

Structural Health

Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.

Quality of Shipment Conditions

Monitoring of vibrations, strokes, container openings or cold chain maintenance for insurance purposes.

Smartphones Detection

Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.

Perimeter Access Control

Access control to restricted areas and detection of people in non-authorized areas.

Radiation Levels

Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

Electromagnetic Levels

Measurement of the energy radiated by cell stations and WiFi routers.

Traffic Congestion

Monitoring of vehicles and pedestrian affluence to optimize driving and walking routes.

Smart Roads

Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

Smart Lighting

Intelligent and weather adaptive lighting in street lights.

Intelligent Shopping

Getting advices in the point of sale according to customer habits, preferences, presence of allergic components for them or expiring dates.

Noise Urban Maps

Sound monitoring in bar areas and centric zones in real time.

Water Leakages

Detection of liquid presence outside tanks and pressure variations along pipes.

Vehicle Auto-diagnosis

Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.

Item Location

Search of individual items in big surfaces like warehouses or harbours.

Waste Management

Detection of rubbish levels in containers to optimize the trash collection routes.

Smart Parking

Monitoring of parking spaces availability in the city.

Golf Courses

Selective irrigation in dry zones to reduce the water resources required in the green.

Water Quality

Study of water suitability in rivers and the sea for fauna and eligibility for drinkable use.

Klasyfikacja metaheurystyk

- **Przeszukiwanie sąsiedztwa:**
 - Tabu search;
 - Symulowane wyżarzanie.
- **Operacje na populacjach rozwiązań:**
 - Algorytmy ewolucyjne;
 - Programowanie genetyczne;
 - *Algorytmy harmoniczne;*
 - Algorytm kolonii mrówek;
 - Algorytmy roju.

Algorytmy harmoniczne

1. Geem Z.W.: Optimal design of water distribution networks using harmony search. Ph.D. Thesis, Korea University 2001.
2. Kaveh A., Talatahari S.: Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Comput. Struct.*, Vol. 87, 2009, pp. 267–283.
3. Kulluk S., Ozbakir L., Baykasoglu A.: Training neural networks with harmony search algorithms for classification problems. *Eng. Appl. Artif. Intell.*, Vol. 25, No. 1, 2012, pp. 11–19.
4. Landa-Torres I. et al.: A multi-objective grouping harmony search algorithm for the optimal distribution of 24-hour medical emergency units. *Expert Syst. Appl.*, Vol. 40, No. 6, 2012, pp. 2343–2349.
5. Manjarres D. et al.: On the design of a novel two-objective harmony search approach for distance- and connectivity-based localization in wireless sensor networks. *Eng. Appl. Artif. Intell.*, Vol. 26, No. 2, 2013, pp. 669–676.

Algorytm harmoniczny

- HS to metaheurystyka inspirowana procesem doboru najlepszego brzmienia akustycznego podczas improwizacji muzyków jazzowych;
- Muzycy podczas improwizowanych sesji wybierają *ad hoc* odpowiednie linie melodyczne, aby uzyskać najlepsze brzmienie, tak jak funkcja przystosowania w algorytmie harmonicznym decyduje o wyborze optimum globalnego;
- Jeśli dźwięki tworzą dobrą harmonię, muzyk przechowuje w pamięci to zdarzenie, a możliwość uzyskania lepszej harmonii w przyszłości wzrasta;
- Podobna sytuacja występuje w optymalizacji: początkowe rozwiązanie jest generowane losowo z uwzględnieniem ograniczeń. Jeśli wartość funkcji przystosowania jest większa niż przystosowanie najgorszego rozwiązania z pamięci algorytmu, to nowe rozwiązanie zastępuje stare i zwiększa szansę na uzyskanie lepszego wariantu w przyszłości.



Inicjalizacja pamięci harmonicznej

- Początkowa pamięć harmoniczna **HM** (*ang. harmony memory*) zawiera **HMS** (*ang. harmony memory size*) wektorów generowanych losowo, gdzie

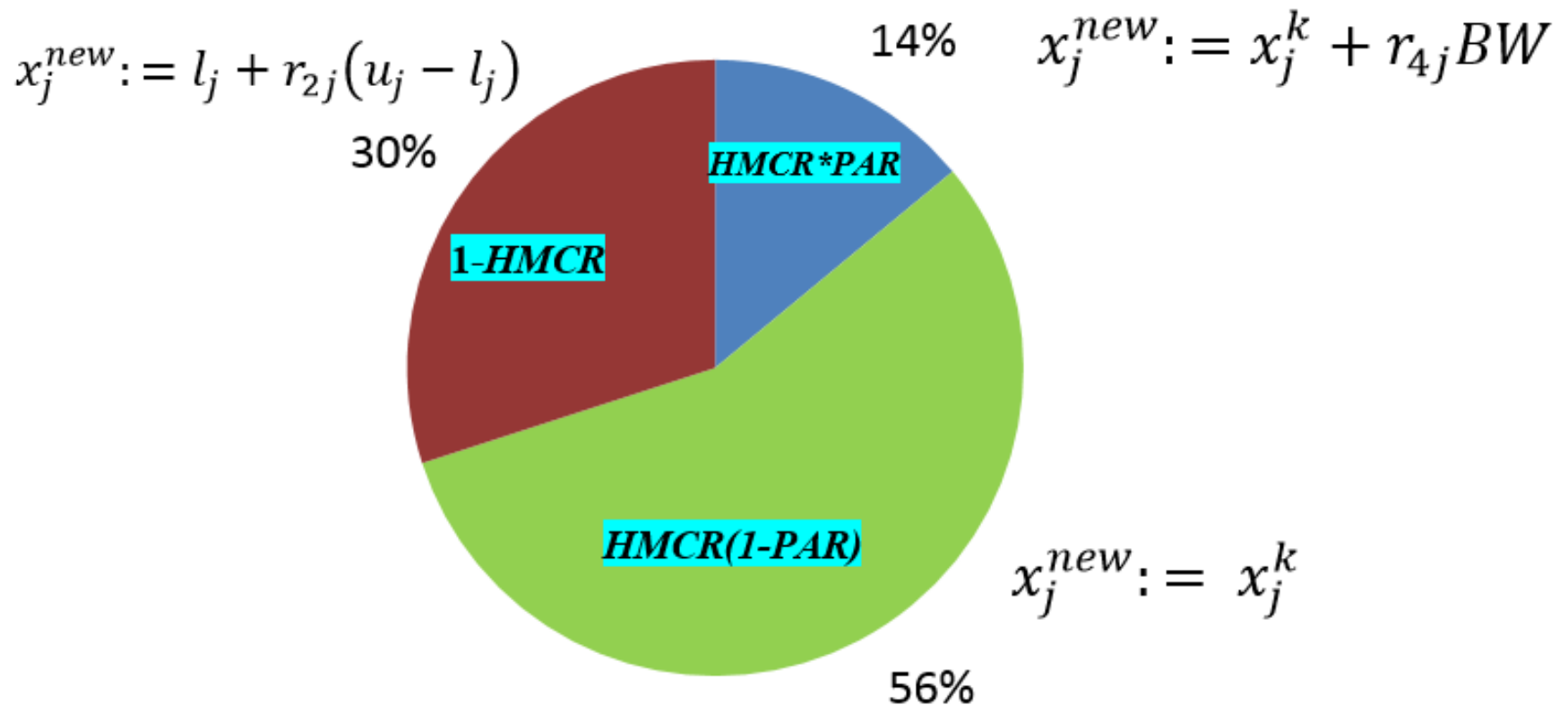
$$x_i = [x_{i1}, \dots, x_{ij}, \dots, x_{in}], i = 1, \dots, \text{HMS} \text{ oraz } j = 1, \dots, n.$$

- Pamięć **HM** można przedstawić za pomocą następującej macierzy dwuwymiarowej o **HMS** wierszach oraz **n** kolumnach:

$$HM = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{HMS1} & x_{HMS2} & x_{HMS3} & \dots & x_{HMSn} \end{bmatrix}$$



Reguła trójwartościowej ruletki



dla $HMCR = 0,7$ oraz dla $PAR = 0,2$

Improwizacja typu HMCR

- **HMCR** (ang. *harmony memory considering rate*) jest parametrem algorytmu harmonicznego i oznacza tempo wyboru wartości zmiennych decyzyjnych z pamięci do nowej improwizacji x^{new} ;
1. Generowana jest losowo liczba r_1 z zakresu $[0,1]$;
 2. Jeśli $r_1 \leq HMCR$, to nowym skonstruowanym wektorem zmiennych decyzyjnych x^{new} jest wektor wyznaczony losowo na podstawie aktualnej pamięci **HM**, jak poniżej:

$$HM = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{HMS1} & x_{HMS2} & x_{HMS3} & \dots & x_{HMSn} \end{bmatrix}$$

$$x^{new} = \begin{bmatrix} x_1^k, & x_2^k, & x_3^k, \dots & x_j^k, \dots & x_n^k \end{bmatrix}$$

$j = 1, 2, \dots, n$; k – losowo wybrany indeks wiersza dla każdej kolumny j

Improwizacja typu PAR

- Po improwizacji typu HMCR zachodzi improwizacja typu PAR;
- Wektor zmiennych decyzyjnych x^{new} wyznaczony za pomocą improwizacji typu **HMCR** jest następnie sprawdzany, czy wymaga dostrojenia za pomocą improwizacji typu **PAR**;
- PAR** (ang. *pitch adjusting rate*) to tempo dostrojenia nowego wektora zmiennych decyzyjnych;



- Liczba losowa r_3 jest generowana z zakresu $[0,1]$:
- Jeśli $r_3 < \mathbf{PAR}$, to:

$$x_j^{new} := x_j^{new} + r_{4j} BW, \quad j = 1, 2, \dots n;$$

gdzie:

BW (ang. *bandwidth factor*) – parametr wyznaczający przedział $[-BW, BW]$,
 r_{4j} - liczba losowa z przedziału $[-1;1]$.

Improwizacja losowa

Jeśli improwizacja typu **HMCR** nie zachodzi, $r_1 > \text{HMCR}$, to nowa zmienna decyzyjna jest generowana losowo w ramach dopuszczalnych wartości z przedziału $[l_j, u_j]$, jak niżej:



$$x_j^{new} = l_j + r_{2j}(u_j - l_j), \quad j = 1, 2, \dots, n;$$

gdzie:

r_{2j} – liczba losowa z zakresu $[0;1]$,

l_j – dolne ograniczenie dla j -tej zmiennej decyzyjnej,

u_j – górne ograniczenie dla j -tej zmiennej decyzyjnej.

Aktualizacja pamięci harmonicznej

Po wygenerowaniu nowej improwizacji bada się, czy wyznaczona improwizacja x^{new} jest lepsza pod względem wartości *fitness* od najgorszej alternatywy x^{worst} w pamięci harmonii.

Jeśli tak, to x^{new} zastępuje najgorsze rozwiązanie z archiwum *HM*. Jeśli nie, to aktualizacja pamięci nie zachodzi.

```
if ( $fitness(x^{new}) > fitness(x^{worst})$ ) then  
    update HM as  $x^{worst} = x^{new}$   
endif
```



Wartości parametrów HMCR i PAR

Średnia liczba improwizacji niezbędna do wyznaczenia konfiguracji suboptymalnej cechującej się $\Delta_{\max}=82,5$ [s] dla wybranych wartości parametrów *HMCR*, *PAR*, przy czym *BW*=1 oraz *HMS*=10

<i>PAR</i>	<i>HMCR</i>				
	0,1	0,3	0,5	0,7	0,9
0,1	3 927 038	3 845 374	3 112 857	1 226 145	618 750
0,3	3 837 746	3 837 647	3 937 465	1 226 145	685 273
0,5	3 847 937	3 736 487	3 745 836	1 534 736	534 377
0,7	3 547 362	3 437 773	3 232 221	1 226 145	592 353
0,9	3 464 675	4 347 443	3 343 847	2 545 534	1 458 282

Rekomendowane wartości parametrów:

HMCR=0,9

PAR=0,5

Algorytm harmoniczny

Algorithm 1 Harmony search algorithm

```
1: Set the parameters  $HMS$ ,  $HMCR$ ,  $PAR$ ,  $BW$  and  $NI$ .
2: Set  $t := 0$ . {Counter initialization}.
3: for ( $i = 1 : i \leq HMS$ ) do
4:   Generate an initial population (harmony memory)  $HMS$ . { $HMS$  is the harmony
   memory size (population size)}.
5:   Evaluate the fitness function of each harmony vector  $f(x_i)$ .
6: end for
7: repeat
8:   Generate a new solution (harmony)  $x_i$  as follows
9:   for ( $i = 1 : i \leq HMS$ ) do
10:    for ( $j = 1 : j \leq n$ ) do
11:      if ( $r_1 < HMCR$ ) then
12:         $x_{ij}^{new} = x_{ij}$ 
13:      if ( $r_2 < PAR$ ) then
14:         $x_{ij}^{new} = x_{ij}^{new} \pm r_3 \cdot BW$ , where  $r_1, r_2, r_3 \in (0, 1)$ 
15:      end if
16:      if  $x_{ij}^{new} < l_j$  then
17:         $x_{ij}^{new} = l_j$ 
18:      end if
19:      if  $x_{ij}^{new} > u_j$  then
20:         $x_{ij}^{new} = u_j$ 
21:      end if
22:    else
23:       $x_{ij}^{new} = l_{ij} + r_4(u_{ij} - l_{ij})$ , where  $r_4 \in rand(0, 1)$ 
24:    end if
25:  end for
26: end for
27: if ( $fitness(x^{new}) > fitness(x^{worst})$ ) then
28:   update HM as  $x^{worst} = x^{new}$ 
29: end if
30: Set  $t = t + 1$ . {Iteration counter increasing}.
31: until ( $t < NI$ ). {Termination criteria satisfied}.
32: Produce the best solution (harmony vector)  $x^{new}$ .
```

Ustawienie wartości parametrów

Inicjalizacja pamięci

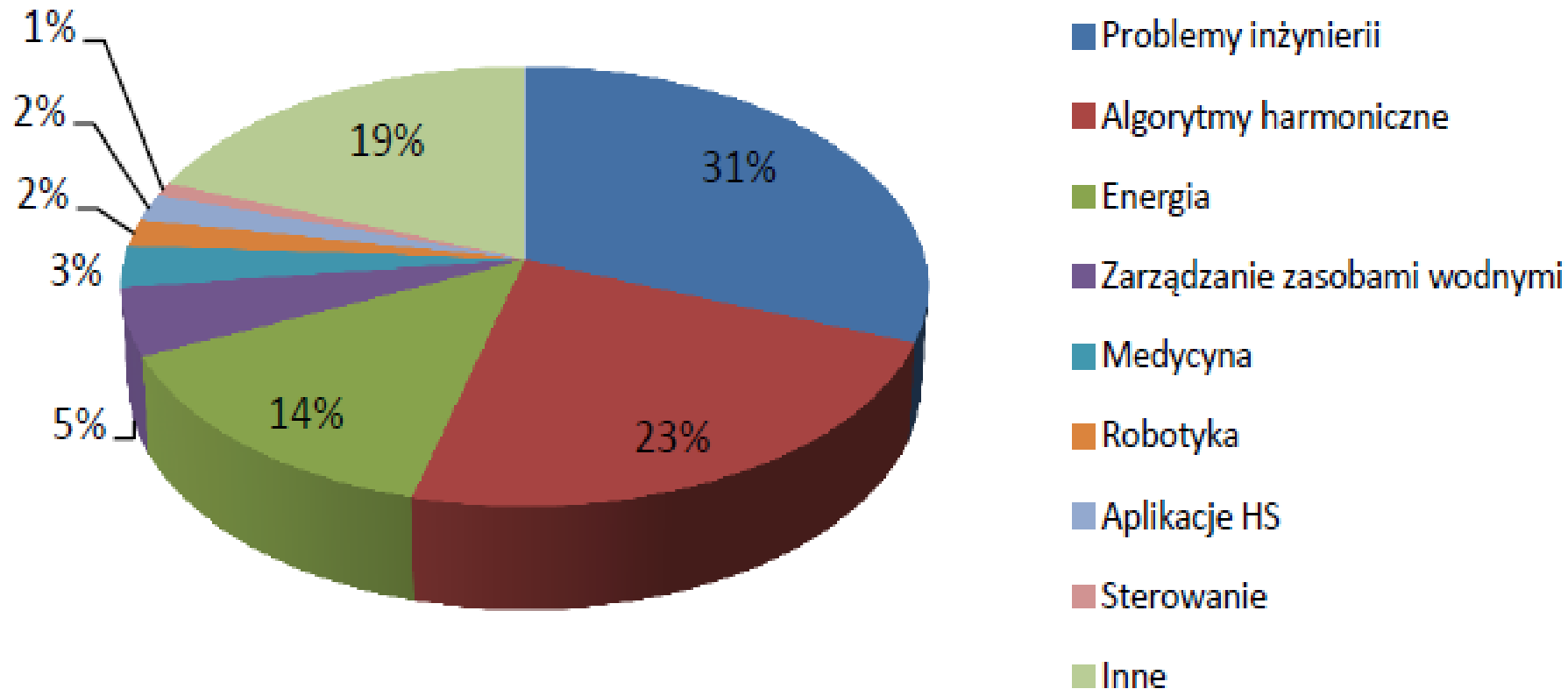
Improwizacja HMCR

Improwizacja PAR

Improwizacja losowa

Aktualizacja pamięci harmonicznej

Zastosowania algorytmu harmonicznego

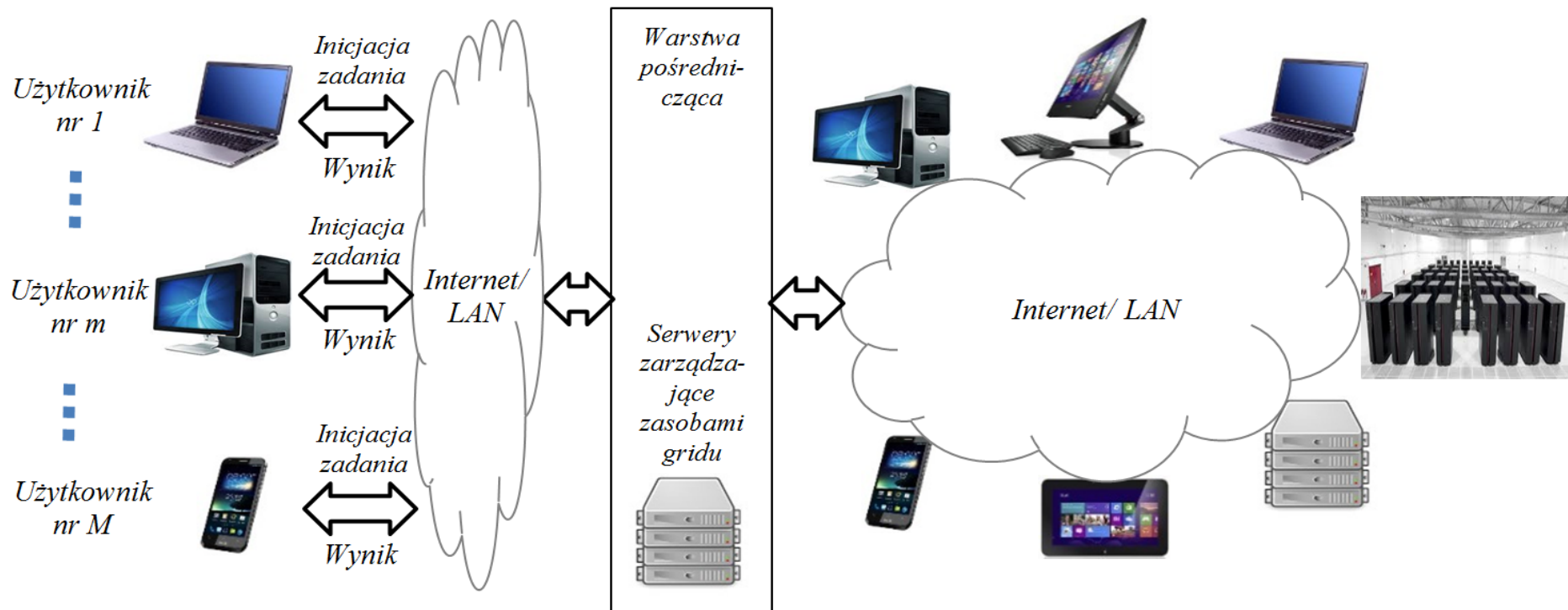


Charakterystyka wybranych gridów

Wydajność obliczeniowa wg benchmarku *Linpack*:

- *Folding@home* - 39,9 [PFLOPS];
- *BOINC* -5,6 [PFLOPS];
- *SETI@Home* - 681 [TFLOPS];
- *Einstein@Home* - 492 [TFLOPS];
- *MilkyWay@Home* - 471 [TFLOPS];
- *GIMPS* -173 [TFLOPS], co umożliwiło odkrycie 48-ej liczby pierwszej *Mersenne'a* w 2013 roku;
- *Pl-Grid* – 942,48 [TFLOPS];
- Najszybszy superkomputer *Tianhe-2* - 33,86 [PFLOPS].

Infrastruktura gridu



Grid jest rozproszonym systemem komputerowym, w którym implementuje się wirtualne środowisko obliczeniowe dla pewnej organizacji zrzeszającej użytkowników potrzebujących usług informatycznych o odpowiedniej jakości.

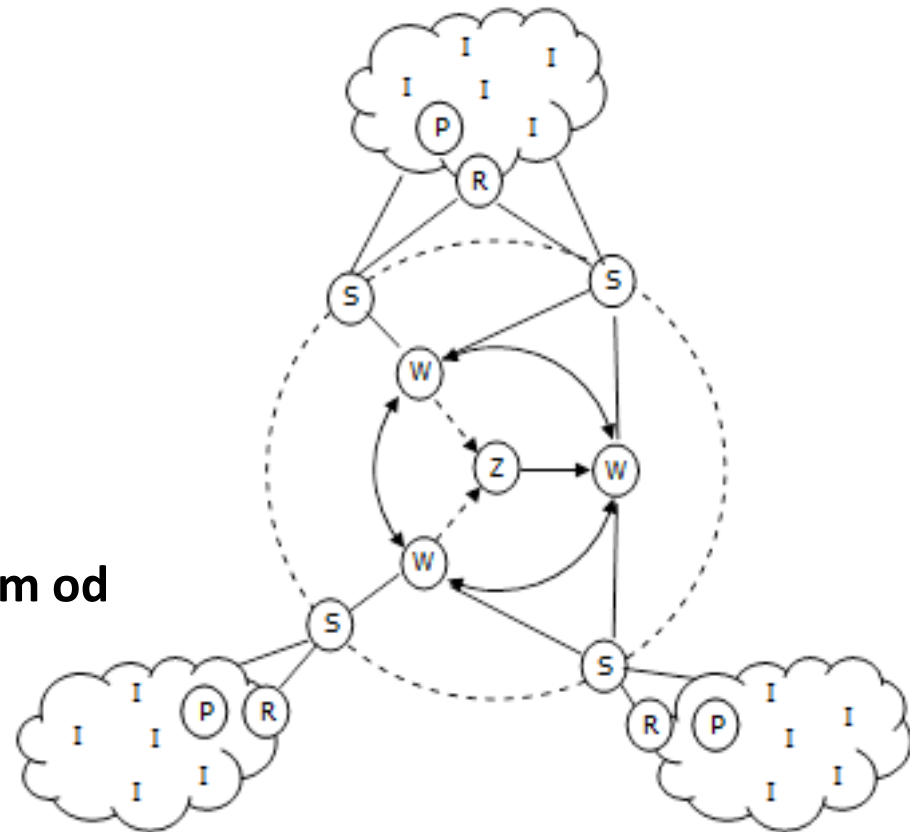
Zarządzanie zasobami odbywa się w wielu domenach z wykorzystaniem uzgodnionego zestawu protokołów i powszechnie dostępnych interfejsów.

Platformy gridowe

- *ARC Advanced Resource Connector;*
- *BOINC;*
- *DIET;*
- *Globus Toolkit;*
- **COMCUTE**
- *EMI;*
- *gLite;*
- *GridWay;*
- *Oracle Grid Engine.*

COMCUTE

1. warstwa **Z** (zleceniodawcy) - interfejs użytkownika;
2. warstwa **W** (wewnętrzna)
- **moduły W** - partycjonowanie, synchronizacja i weryfikacja wyników obliczeń;
3. warstwa **S** (dystrybucyjna, brzegowa)
- **moduły S** - dostarczanie zadań do obliczeń dla internautów i odbieraniem od nich wyników;
4. warstwa **I** (publiczna, zewnętrzna) - komputery internautów



Model optymalizacji wykorzystania zasobów w gridzie

- Podział oprogramowania gridowego na moduły, np. W i S w gridzie *Comcute*, umożliwia skrócenie czasu ich wykonania poprzez zrównoleglenie ich realizacji;
- **Redukcja kosztu modyfikacji oprogramowania.**
Wprowadzanie zmian deweloperskich odnosi się wówczas do podzbioru modułów, a nie do całego systemu;
- Dzielenie programu na moduły ułatwia zarządzanie projektem realizowanym przez zespół programistów, co skutkuje podniesieniem jakości produktu.

Optimalizacja wykorzystania zasobów

- Zasoby gridowe (procesory, RAM, dyski, podsystem komunikacyjny, usługi) należy skonfigurować do potrzeb oprogramowania pośredniczącego lub aplikacji;
- **Dopasowanie konfiguracji gridu do specyfiki oprogramowania zachodzi za pomocą optymalizacji wykorzystania zasobów przez oprogramowanie;**
- Instrukcje graficzne lub instrukcje w języku CUDA C++ zapisane w zadaniu będą szybciej wykonywane na karcie graficznej NVIDIA niż na procesorze serwera bazodanowego, do którego należałoby raczej przydzielić zadanie z instrukcjami w języku SQL.

Konfiguracja zasobów w gridzie

a) *Przydział modułów programu do węzłów*

$$X^{\alpha} = [X_{11}^{\alpha}, \dots, X_{1i}^{\alpha}, \dots, X_{1I}^{\alpha}, \dots, X_{vi}^{\alpha}, \dots, X_{VI}^{\alpha}]^T$$

gdzie

$$x_{vi}^{\alpha} = \begin{cases} 1 & \text{gdy } \alpha_v \text{ usytuowano w } w_i, \\ 0 & \text{w przeciwnym razie,} \end{cases} \text{ dla } v = \overline{1, V}, i = \overline{1, I}.$$

Konfiguracja zasobów w gridzie

b) Przydział rodzajów komputerów do węzłów

$$x^{\beta} = [x_{11}^{\beta}, \dots, x_{1j}^{\beta}, \dots, x_{1J}^{\beta}, \dots, x_{ij}^{\beta}, \dots, x_{IJ}^{\beta}]^T$$

gdzie

$$x_{ij}^{\beta} = \begin{cases} 1 & \text{gdy do } w_i \text{ przydzielono komputer typu } \beta_j, \\ 0 & \text{w przeciwnym razie,} \end{cases} \quad \text{dla } i = \overline{1, I}, j = \overline{1, J}.$$

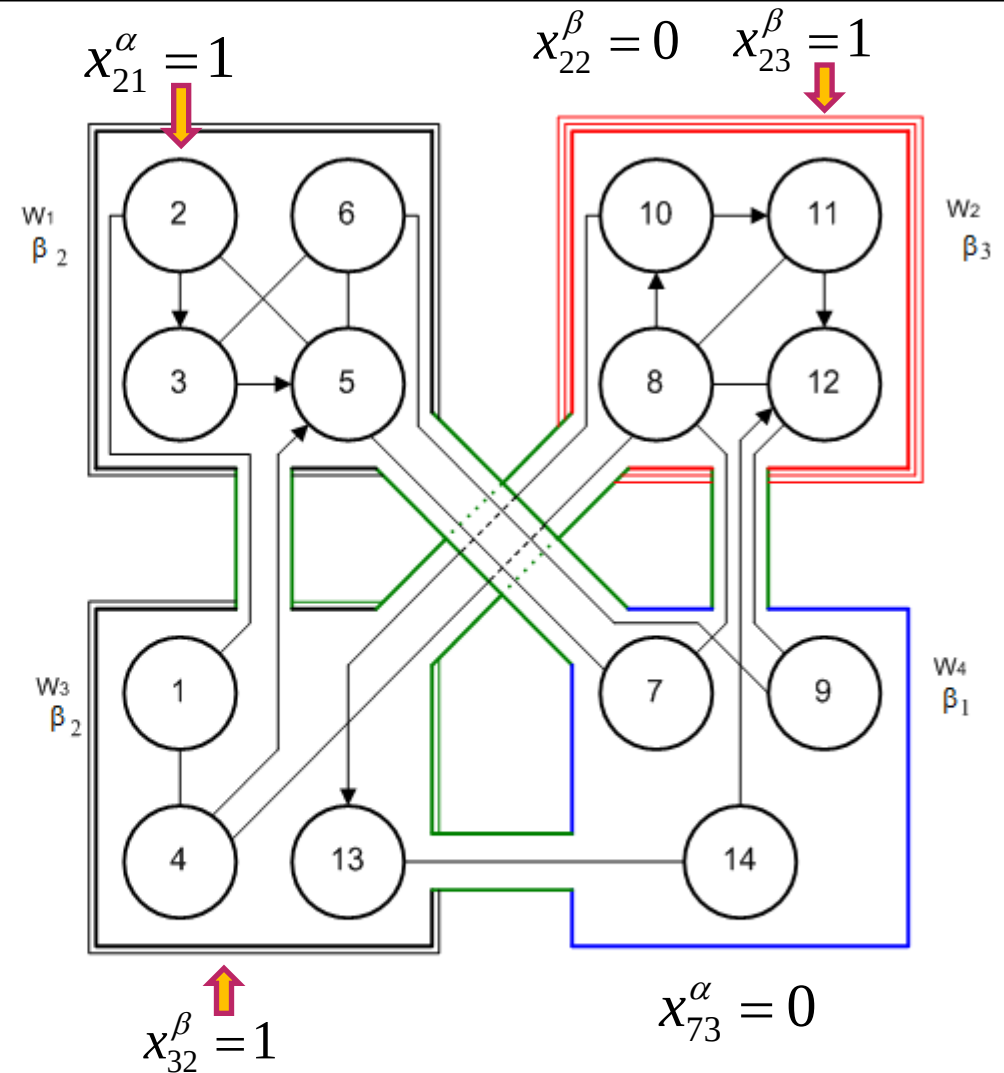
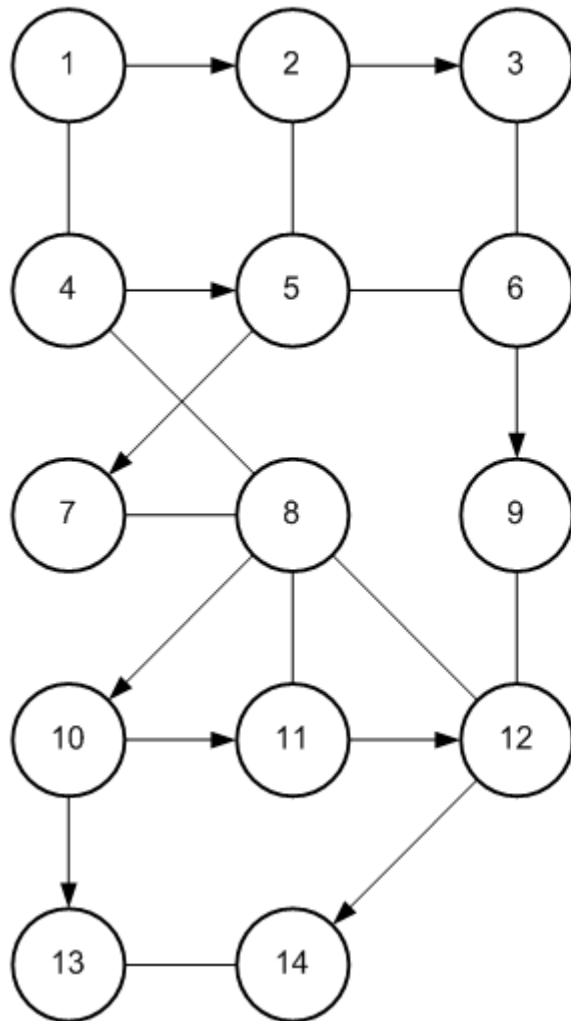
Konfiguracja zasobów w gridzie

$$X = [x_{11}^{\alpha}, \dots, x_{1i}^{\alpha}, \dots, x_{1I}^{\alpha}, \dots, x_{vi}^{\alpha}, \dots, x_{VI}^{\alpha}, x_{11}^{\beta}, \dots, x_{1j}^{\beta}, \dots, x_{1J}^{\beta}, \dots, x_{ij}^{\beta}, \dots, x_{IJ}^{\beta}]^T$$

Przyjmuje się, że konfiguracja dopuszczalna powinna spełniać:

- **postulaty alokacji każdego modułu do jednego z węzłów;**
- **warunki przydziału tylko jednego komputera do węzła.**

Konfiguracja zasobów w gridzie



Klasyfikacja problemów optymalizacji konfiguracji zasobów w gridzie

<i>Założenie</i>	<i>Atrybut</i>		
Ograniczoność zasobów	- ograniczone - nieograniczone		
Graf opisujący strukturę komunikacyjną między modułami	- dowolny - szczególny: ♦ drzewo ♦ sekwencyjno - współbieżny ♦ potok		
Struktura topologiczna sieci komunikacyjnej	- dowolna - szczególna:		
	tandem	łańcuch	gwiazda

Klasyfikacja problemów optymalizacji konfiguracji zasobów w gridzie

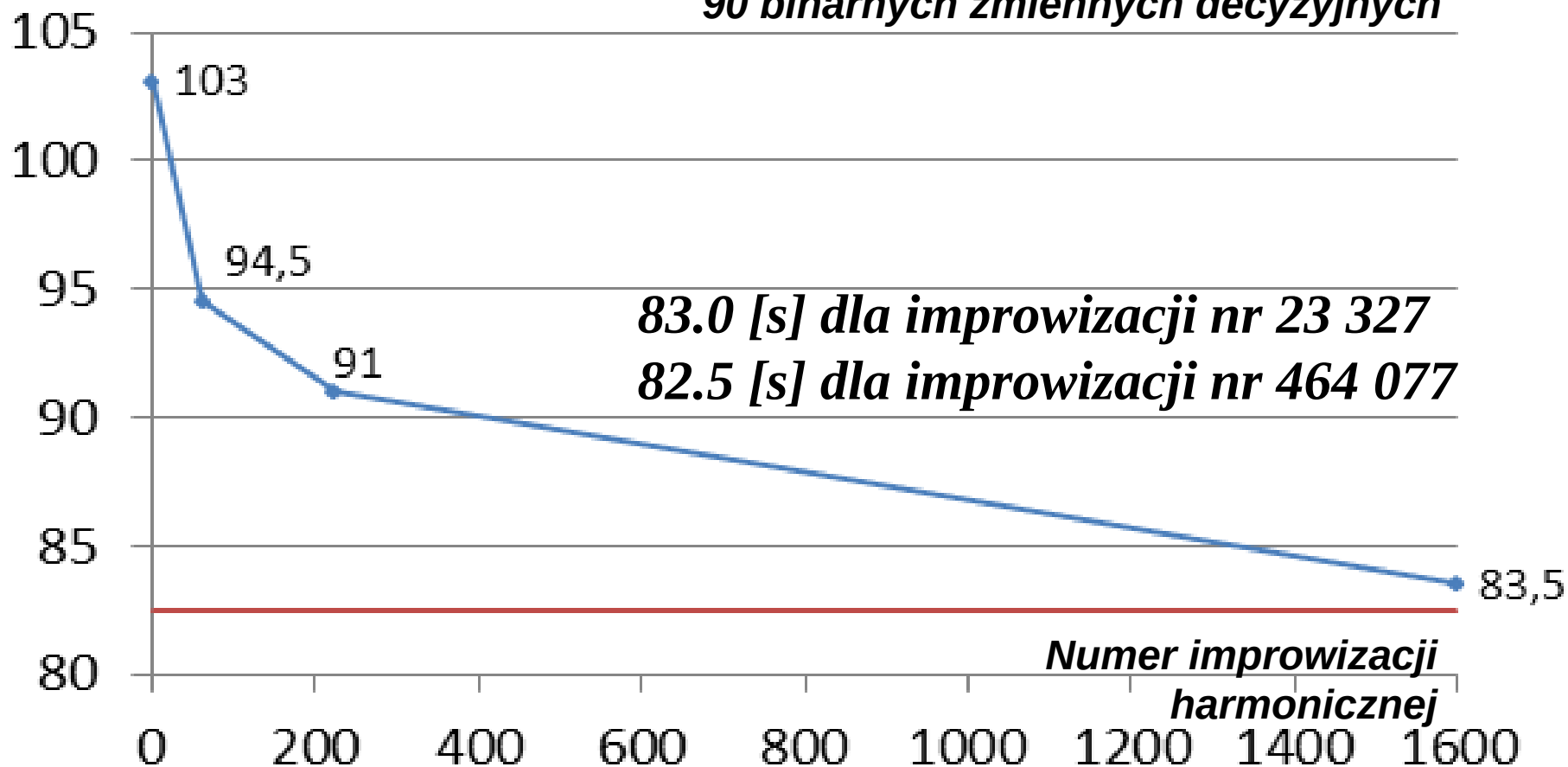
<i>Założenie</i>	<i>Atrybut</i>
Liczba kryteriów	• Jedno kryterium • Optymalizacja wielokryterialna: – dominujące – niezdominowane – hierarchiczne – kompromisowe

Minimalizacja Δ_{max} za pomocą $HS\Delta max$

Δ_{max} [s]

$V=12, I=6, J=3$

90 binarnych zmiennych decyzyjnych



**Binarna przestrzeń przeszukiwań zawiera $1,24 \cdot 10^{27}$ elementów;
Dopuszczalnych konfiguracji gridu jest nie więcej niż $1,59 \cdot 10^{12}$.**

ADAIORG'14 - Aplikacja Do Analizy I Optymalizacji Rekonfiguracji Gridu

ADAIORG'14 ver. 1.6 - ©Jacek Paluszak

ADAIORG'14 Application For Analysys And Optimization Grid Reconfiguration

harmony search

Help

About...

randomizing 100 grid configs.
randomizing time: 0.005 seconds.
minimas time: 0.0 seconds.
TOTAL time: 0.005 seconds.
display time: 0.0 seconds.
TOTAL with displ: 0.005 seconds.

randomizing 100 grid configs.
randomizing time: 0.002 seconds.
minimas time: 0.0 seconds.
TOTAL time: 0.002 seconds.
display time: 0.0 seconds.
TOTAL with displ: 0.002 seconds.

randomizing 100 grid configs.
randomizing time: 0.002 seconds.
minimas time: 0.0 seconds.
TOTAL time: 0.002 seconds.
display time: 0.0 seconds.
TOTAL with displ: 0.002 seconds.

I. grid parameters

II. objective functions

III. CALCULATIONS

Step 1,2. Grid

1. Read grid config.

2a. Read saved HM.

OR

2b. Randomize $X\alpha$ and $X\beta$

☐ with displaying all

number of random $X\alpha$ and $X\beta$

User requirements

☒ $\Theta(x) \geq \vartheta_{min} = 20.0$

☒ $\Xi(x) \leq \xi_{max} = 7000.0$

☒ $v(x) \geq v_{min} = 80.0$

☒ $\kappa_{ir}(x) \geq 0$

☒ $E(x) \leq \varepsilon_{max} = 3800.0$

Algorithm parameters

harmony memory considering rate HMCR:

pitch adjusting rate PAR:

bandwidth of generation BW1:

bandwidth of generation BW2:

harmony memory size HMS:

Minima list dependent on Z^{max} array

41. $Z^{max}(2) = 5.0$ $Z^{max}(2) = 210.0$ $\Delta_{max} = 107.5$ $Z_{suma}(x) = 93$

51. $Z^{max}(5) = 5.0$ $Z^{max}(5) = 252.0$ $\Delta_{max} = 128.5$ $Z_{suma}(x) = 91$

Minima list dependent on Z^{max} array

14. $Z^{max}(1) = 6.0$ $Z^{max}(3) = 188.0$ $\Delta_{max} = 97.0$ $Z_{suma}(x) = 988$

Minima list dependent on Δ_{max} array

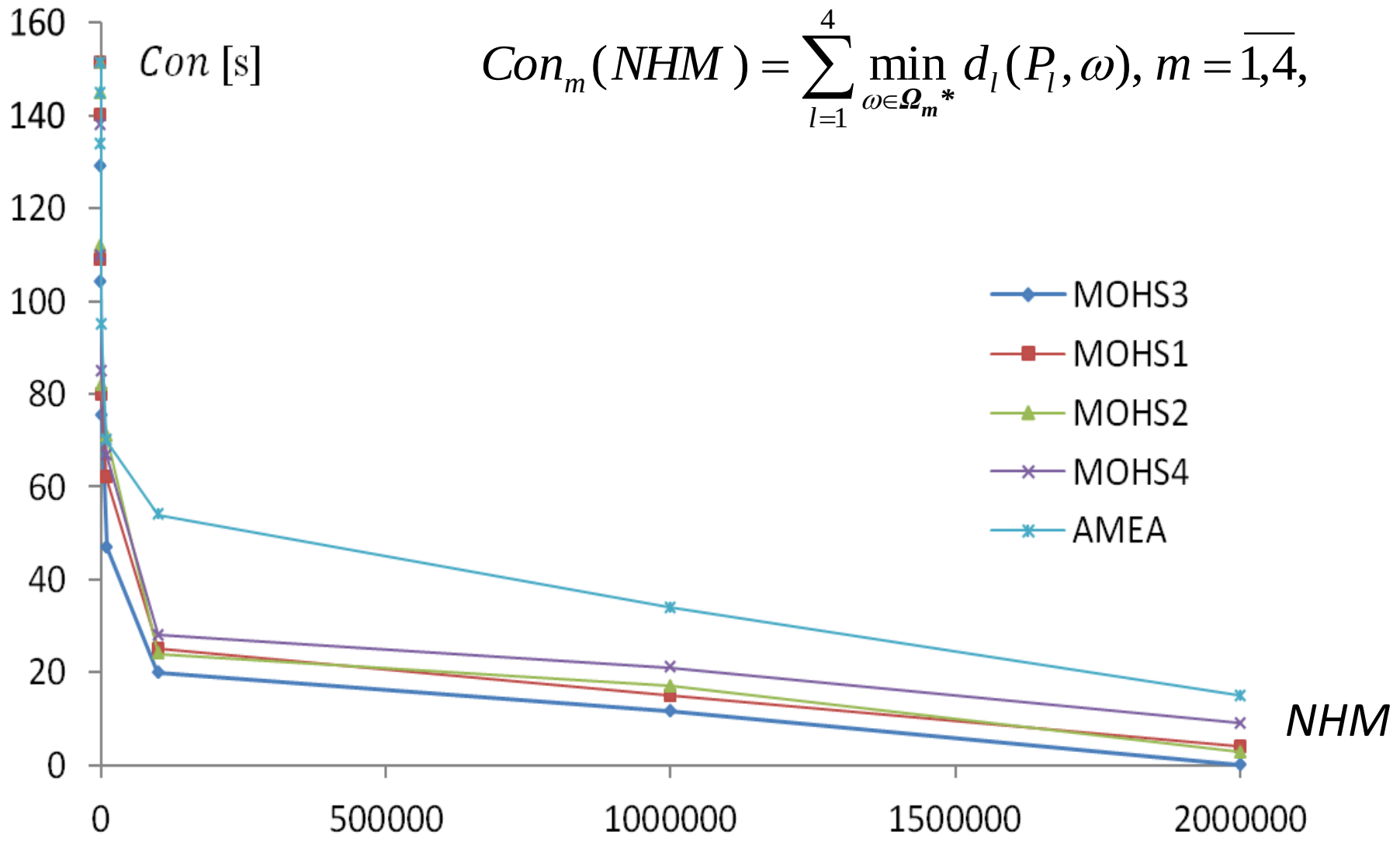
14. $Z^{max}(1) = 6.0$ $Z^{max}(3) = 188.0$ $\Delta_{max} = 97.0$ $Z_{suma}(x) = 988$

Minima list dependent on Z_{suma} array

70. $Z^{max}(3) = 16.0$ $Z^{max}(3) = 249.0$ $\Delta_{max} = 132.5$ $Z_{suma}(x) = 6$

Porównanie wielokryterialnych algorytmów harmonicznych MOHS i ewolucyjnych

$$Con_m(NHM) = \sum_{l=1}^4 \min_{\omega \in \Omega_m^*} d_l(P_l, \omega), m = \overline{1,4},$$



ADAIORG'14 - Aplikacja Do Analizy I Optymalizacji Rekonfiguracji Gridu

ADAIORG'14 ver. 1.6 - ©Jacek Paluszak

ADAIORG'14 Application For Analysys And Optimization Grid Reconfiguration

harmony search

Help

About...

harmony search ended.
generations: 302
HM size: 10
HMCr: 0.9
PAR: 0.5
BW1: 2.0
BW2: 1.0
execution time: 0.123 seconds.

I. grid parameters II. objective functions III. CALCULATIONS

Step 4. Calculations

initialize memory

start calculations

continue calculat.

100%

number of generations NGmax:
302
(max: 10 000 000)

current generation NG:
302

Diversification

all harmony memory

☐ autodiversification

grade of diversification BWN:
4
0 1 2 3 4 5 6 7 8 9 10

Options

save best Δ

saving frequency: 100

save HM

save Z^* , $Z_{\sim}$ max

☐ display temporary HM2
☐ display concat. HM+HM2

HARMONY MEMORY MATRIX

Xa1	Xa2	Xa3	Xa4	Xa5	Xa6	Xa7	Xa8	Xa9	Xa10	Xa11	Xa12	X β 1	X β 2	X β 3	X β 4	X β 5	X β 6	Z*	Z $\sim$	η
5	6	3	3	2	4	2	4	6	1	5	6	1	2	2	3	1	2	5.0	176.0	1
3	6	2	5	4	3	2	4	1	1	5	6	2	2	3	1	1	2	5.0	176.0	1
3	1	4	5	2	2	5	4	6	1	3	6	2	2	1	2	2	2	6.0	169.0	1
3	6	1	5	4	4	2	2	6	3	5	6	2	2	3	3	2	2	5.0	176.0	1
5	4	4	5	2	4	2	6	1	6	3	3	2	2	2	3	2	3	5.0	182.0	4
3	5	1	5	4	2	4	3	1	6	6	2	2	2	2	2	1	1	7.0	176.0	5
6	6	2	5	4	3	5	4	2	1	3	6	2	2	2	1	2	2	6.0	180.0	5
1	6	2	3	4	2	5	3	1	6	6	4	2	2	1	3	2	3	5.0	204.0	5
1	6	2	5	4	3	2	3	3	1	5	6	3	2	3	3	1	3	5.0	207.0	6
3	6	4	5	2	5	4	1	1	6	3	4	3	2	2	1	1	1	7.0	178.0	6

Wnioski i uwagi

- 1. Problemy optymalizacji wykorzystania zasobów w gridzie opierają się na minimalizacji ważonego obciążenia procesorów w newralgicznym komputerze i obciążenia komunikacyjnego w newralgicznym komputerze przy uwzględnieniu ograniczeń takich jak: zagwarantowanie planowanej wydajności gridu, nieprzekroczenie kosztów zakupu komputerów, wymuszenia odpowiedniego stopnia rozproszenia modułów czy też postulatu nieprzekroczenia wielkości dostępnej pamięci RAM czy pamięci dyskowej.**
- 2. Uwzględniane są także inne wymagania, takie jak żądanie nieprzekroczenia zadanego obciążenia procesorów newralgicznego komputera, czy też zapewnienie odpowiedniego poziomu dostępności gridu. Istotne jest rozważanie kwestii ekologicznej związanej z ograniczeniem poboru energii elektrycznej.**

Kierunki dalszych badań

1. Kierunkiem dalszych badań w sztucznej inteligencji, optymalizacji wielokryterialnej i systemach rozproszonych będzie rozwijanie metod *harmony search* w celu rozwiązywania innych zagadnień polioptymalizacji, w tym dla ciągłych zmiennych decyzyjnych.
2. Perspektywicznym kierunkiem badań stanowią prace nad implementacją opracowanych algorytmów *harmony search* jako algorytmy kwantowe.

Dziękuję za uwagę

