

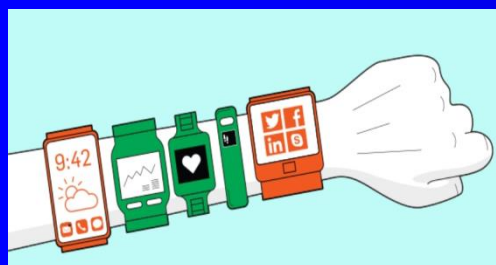
*Seminarium Inteligencji Obliczeniowej
w Zakładzie Sztucznej Inteligencji
i Metod Obliczeniowych*



Warszawa, 21 czerwca 2017

*Programowanie Genetyczne w Uczeniu Maszynowym
i Optymalizacji Systemów Rozproszonych*

Seminarium nr 3



dr hab. inż. Jerzy Balicki, prof. PW

Politechnika Warszawska

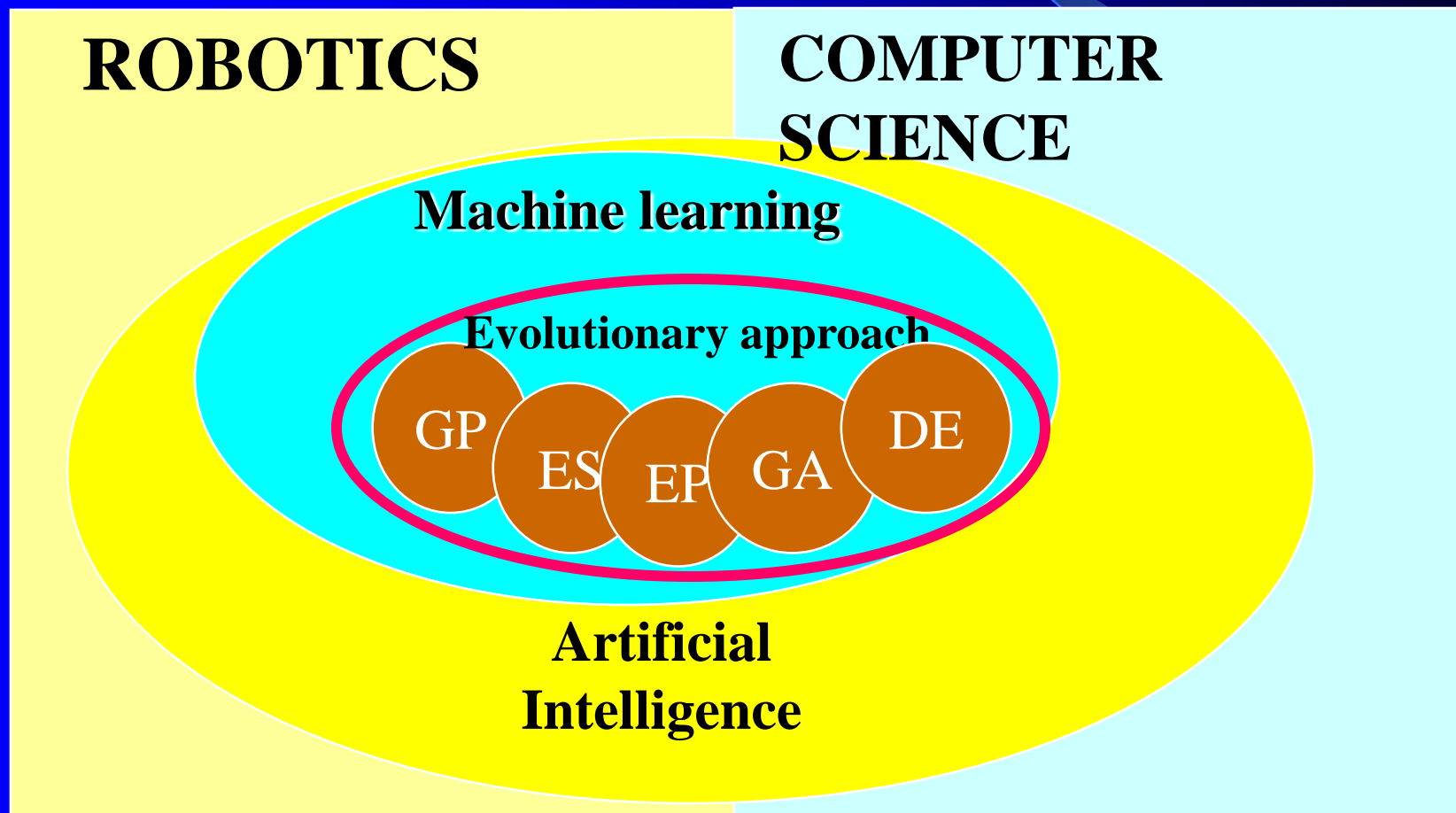
Wydział Matematyki i Nauk Informacyjnych

Plan

- 1.** Wprowadzenie do programowania genetycznego GP;
- 2.** Potencjalne zastosowania GP;
- 3.** Inteligentne metody w uczeniu maszynowym i optymalizacji systemów rozproszonych;
- 4.** Wybrane zastosowania równoległych metod sztucznej inteligencji.



What is Genetic Programming (GP)?



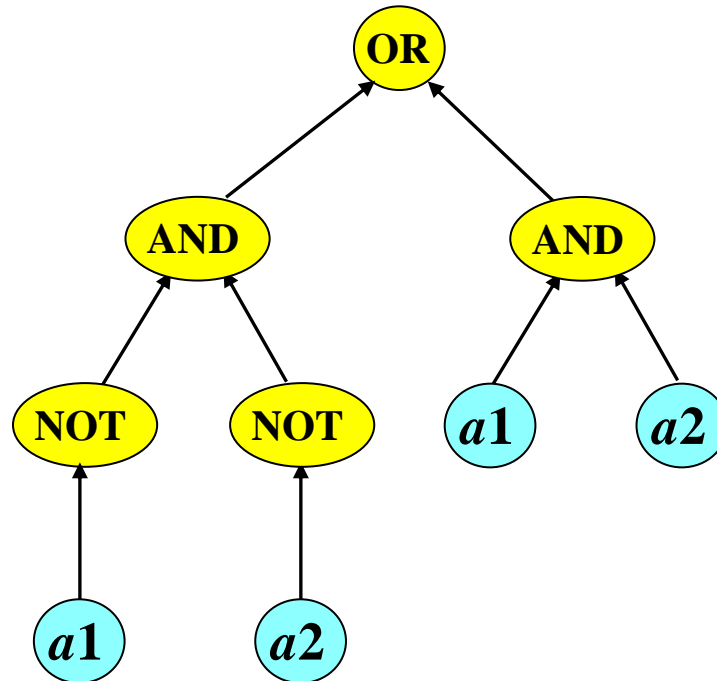
Constrained evolution

1. The size, structure and contents of a program may be dynamically changed during evolution;
2. The program size is constrained by the maximal number of tree nodes or the maximal number of the tree levels;
3. Nodes of a program tree are selected from sets:
 - a) the function set $\mathcal{F} = \{f_1, \dots, f_n, \dots, f_N\}$
 - b) the terminal set $\mathcal{T} = \{a_1, \dots, a_m, \dots, a_M\}$
4. Vector of input argument number
$$Z = [z_1, \dots, z_n, \dots, z_N]$$

Finding the boolean function

$\mathcal{F} = \{\text{AND, OR, NOT}\}$, $\mathcal{V} = \{a_1, a_2\}$, $a_1, a_2 \in \{0,1\}$; $Z = [2, 2, 1]$,

Example: Even-2-parity function



S-expression in Common Lisp

(OR (AND (NOT a1) (NOT a2)) (AND (a1 a2)))

Closure feature of sets of functions and terminals

Each of the functions in \mathcal{F} is able to accept, as its argument, any value and data type that may possibly be:

- returned by any function in \mathcal{F} ;*
- assumed by any terminal in \mathcal{T} .*

Some cases:

- 1. If \mathcal{F} consists of Boolean operators and arithmetic operations, then logical functions are supposed to be re-defined to accept numerical values **0** and **1**;**
- 2. To avoid the overflow error during program run like $x/0$ or $\exp(1000)$, some functions like $/$, \exp should be re-defined to return outcome **inf**;**
- 3. An outcome of operations, that are not defined in mathematical sense, like $0/0$ or inf-inf are denoted by **NaN** (not a number)**

Sufficiency feature of sets of functions and terminals

The set of functions and the set of terminals are supposed to be capable of expressing a solution to the problem;

An example:

Finding trajectory for the artificial ant for the Santa fe trail

How to find 89 pellets of food among 1024 fields in the matrix 32 x 32 by 400 movements of a microrobot?

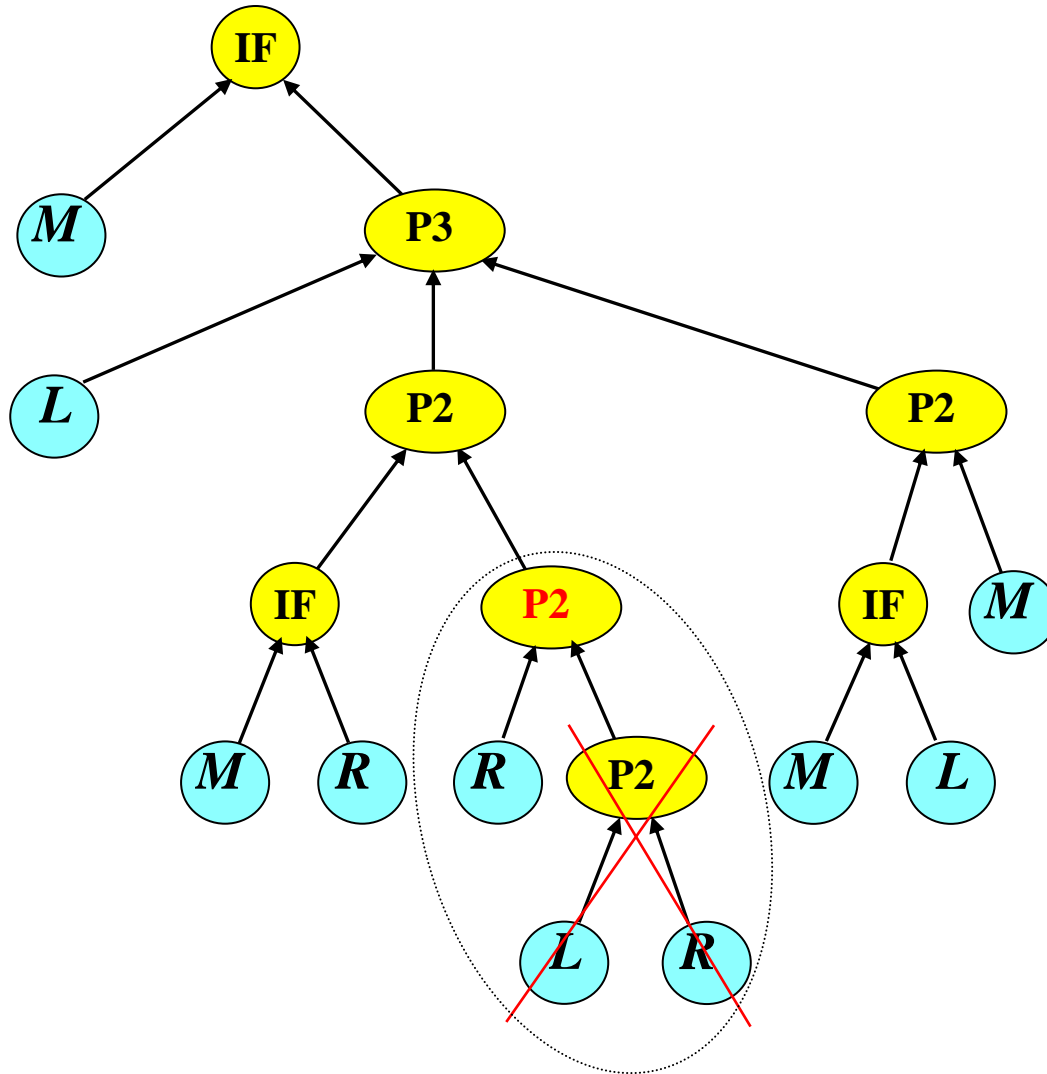
$$\mathcal{F} = \{\text{MOVE, RIGHT, LEFT}\}$$

$$\mathcal{Z} = \{\text{IF - FOOD - AHEAD, P2, P3}\}$$

$$Z = [2, 2, 3]$$

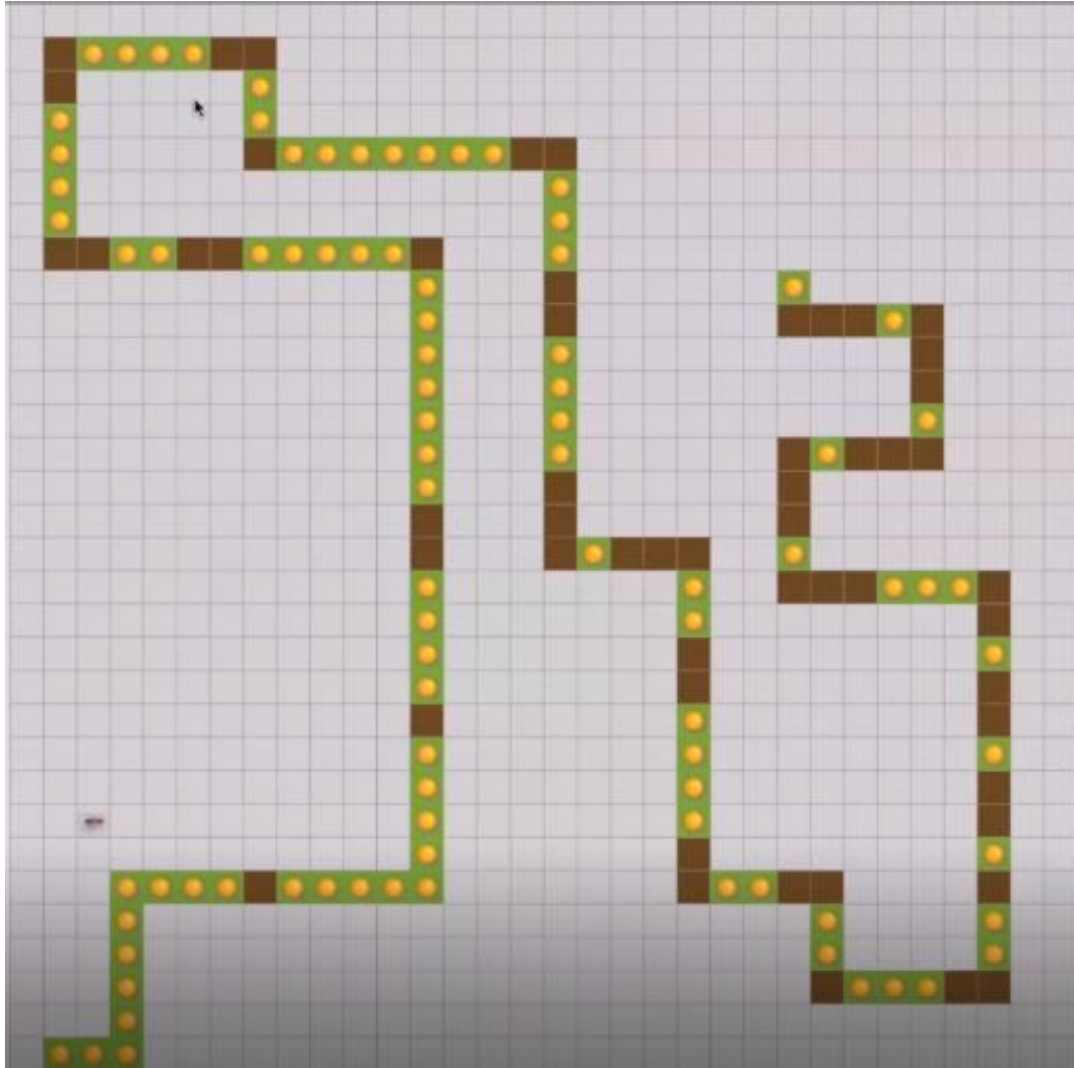
Are sets \mathcal{Z} and \mathcal{F} sufficient or not?

Strategy for the artificial ant - heuristic



The shortest trajectory consists of 55 gaps and 89 pieces of food

Santa Fe Trail Problem

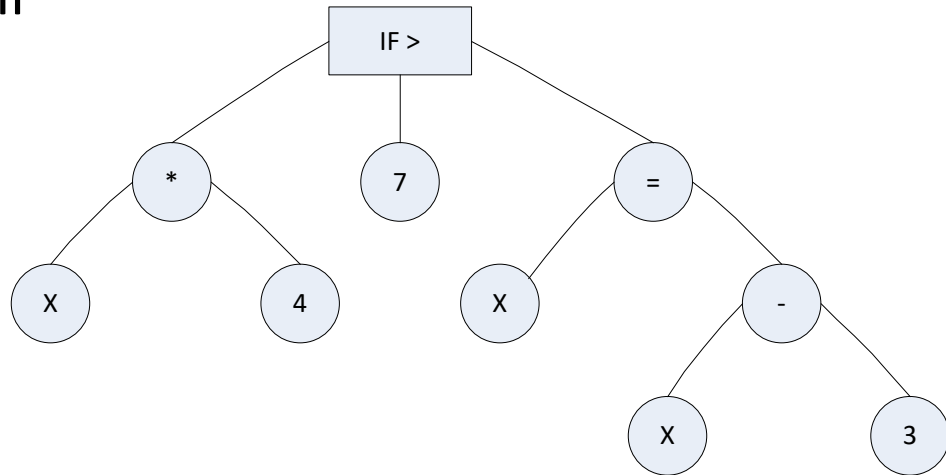


See it on You Tube: Uploaded on Jan 3, 2012

The "Santa Fe Trail" problem solved automatically using Genetic Programming and the GP.Lab toolkit <https://www.youtube.com/watch?v=BKF7pGw8qbY&t=94s>

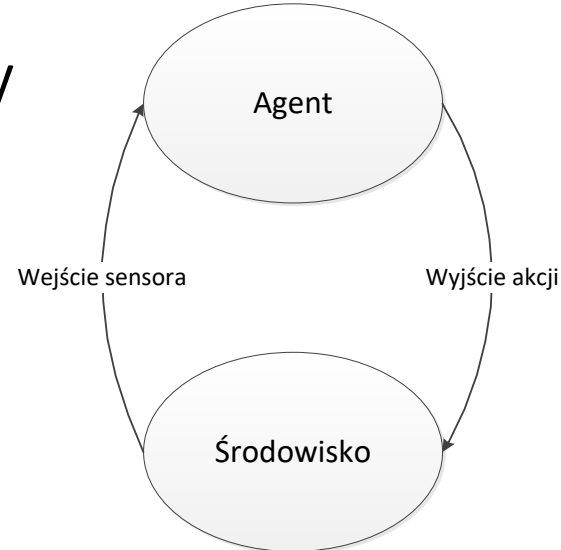
Agentowe GP

- Połączenie dwóch pojęć
 - Programowanie genetyczne
 - Osobniki to programy
 - Losowość
 - Dobór terminali

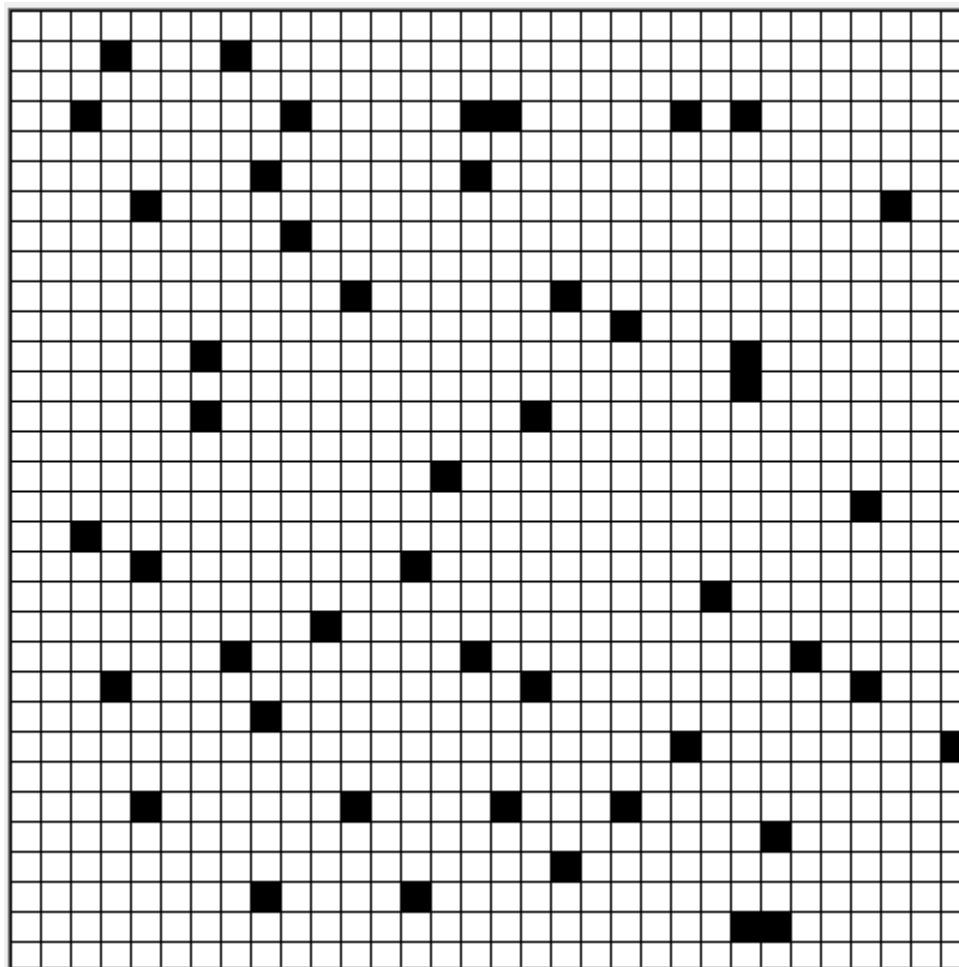


Agentowe GP

- Połączenie dwóch pojęć
 - Agenty
 - Umieszczony w środowisku
 - Autonomiczny
 - Możliwość współpracy

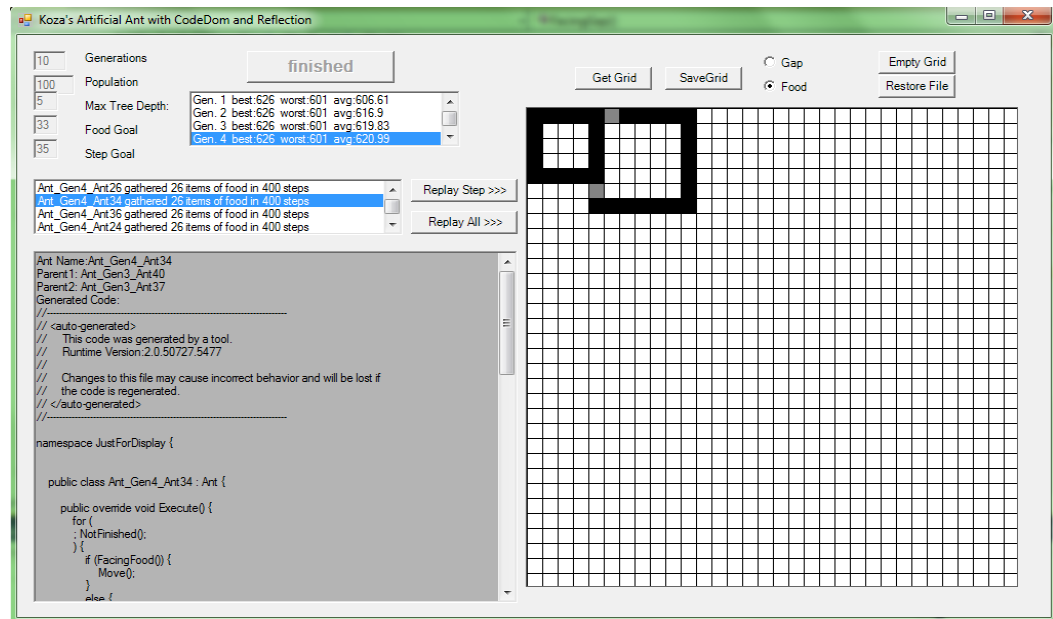


Plansza



Implementacja

- Język C# (biblioteka CodeDom)
- Oparte na programie Braina Connolly'iego (mrówka Kozy)
- Wymagane zmiany by dopasować program do problemu



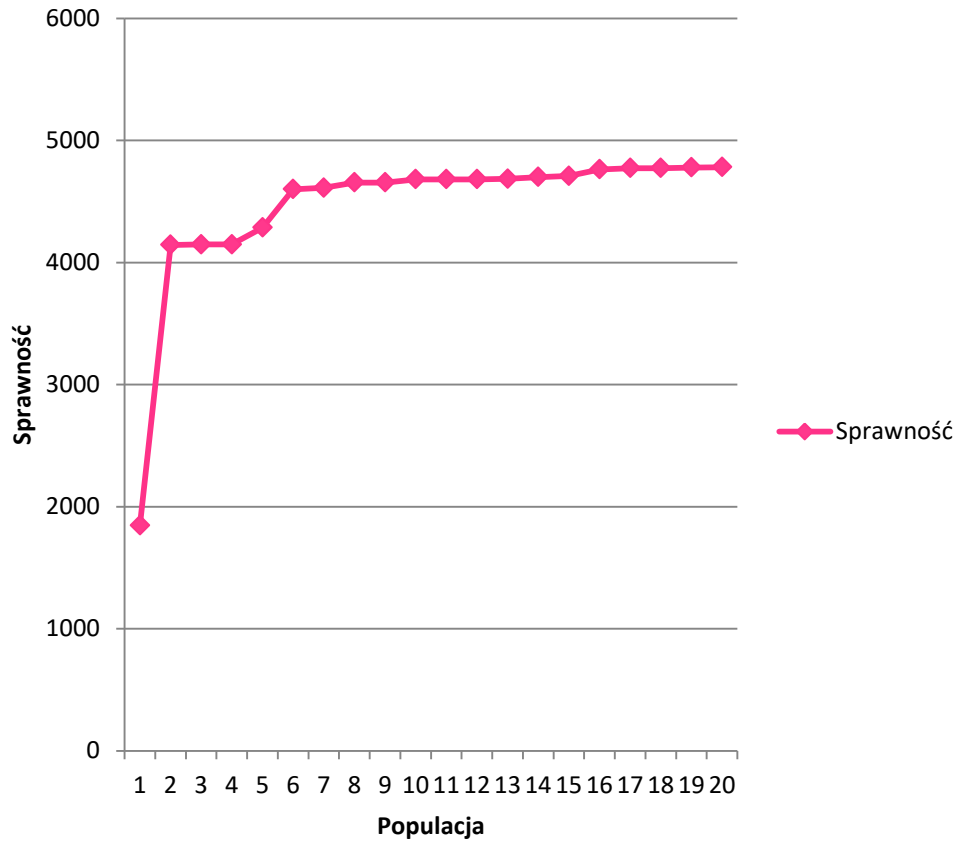
Model A – Bez wiedzy

- Agenty bez wiedzy o rozmieszczeniu elementów na planszy;
- Tylko przylegające pola;
- Wiedza o odwiedzonych polach;

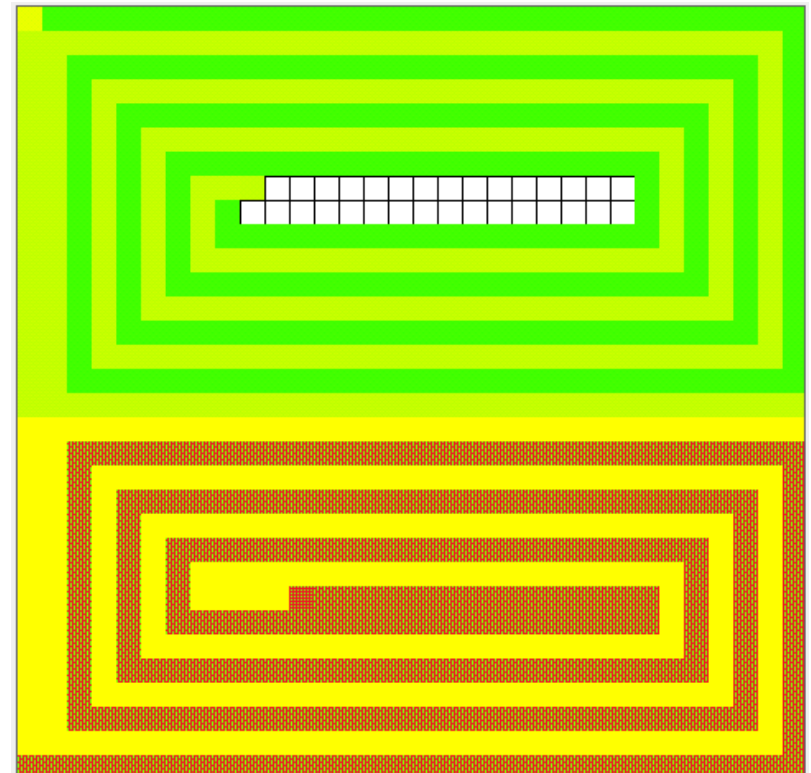
- Wnioski:
 - Współpraca;
 - Różne strategie;
 - Wpływ funkcji przystosowania.

Model A – bez wiedzy

Sprawność najlepszych mrówek



Agent 1	10 dań
Agent 2	10 dań
Agent 3	13 dań
Agent 4	14 dań

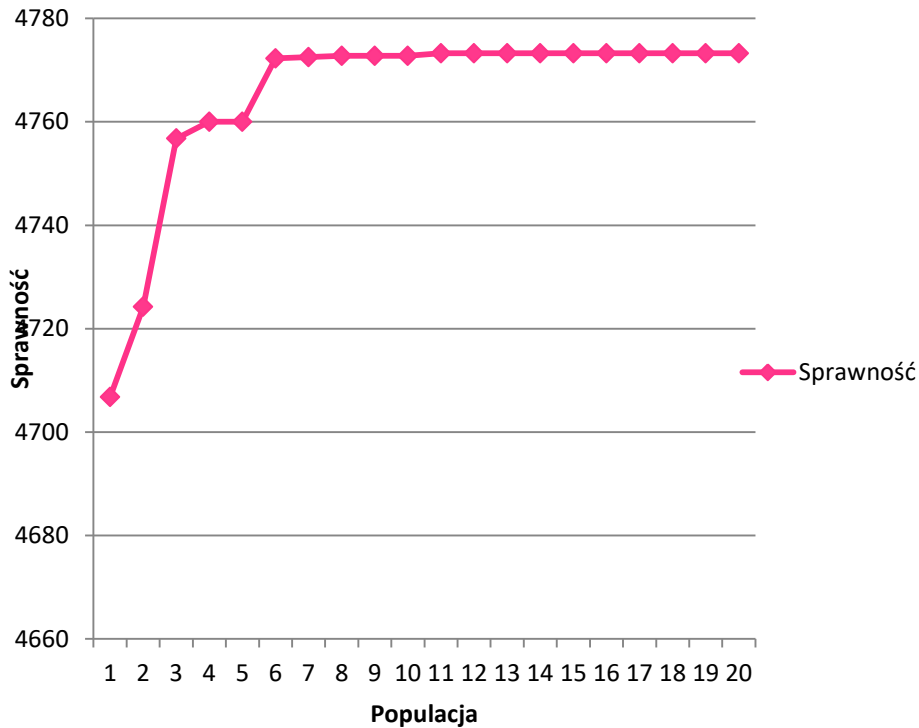


Model B - Wiedza

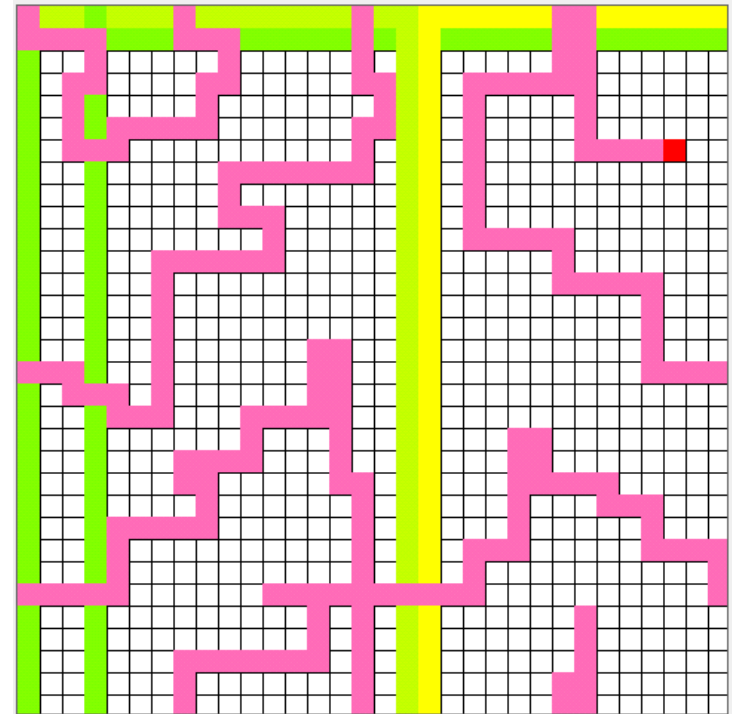
- Agenty mają pewną wiedzę o planszy:
 - Najbliższy element
- Spodziewana poprawa wyniku;
- Wnioski:
 - Brak poprawy;
 - Rywalizacja;
 - Wiedza nie musi się równać osiągnięciom.

Model B - Wiedza

Sprawność najlepszych mrówek



Agent 1	42 dania
Agent 2	1 dania
Agent 3	2 dania
Agent 4	2 dania

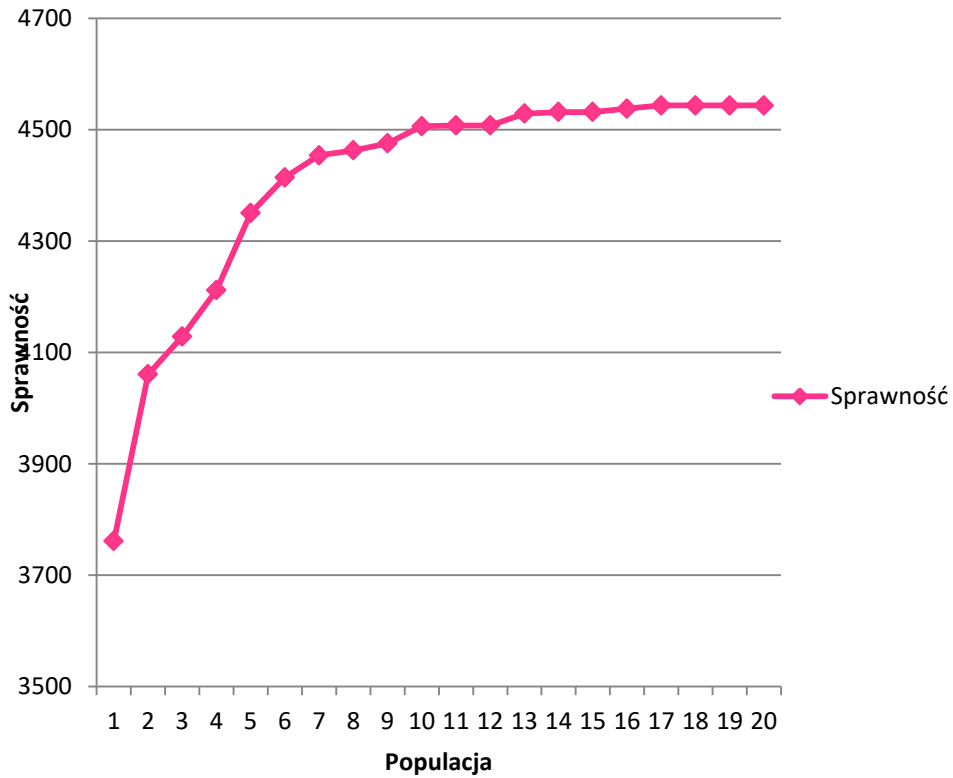


Model C - Komunikacja

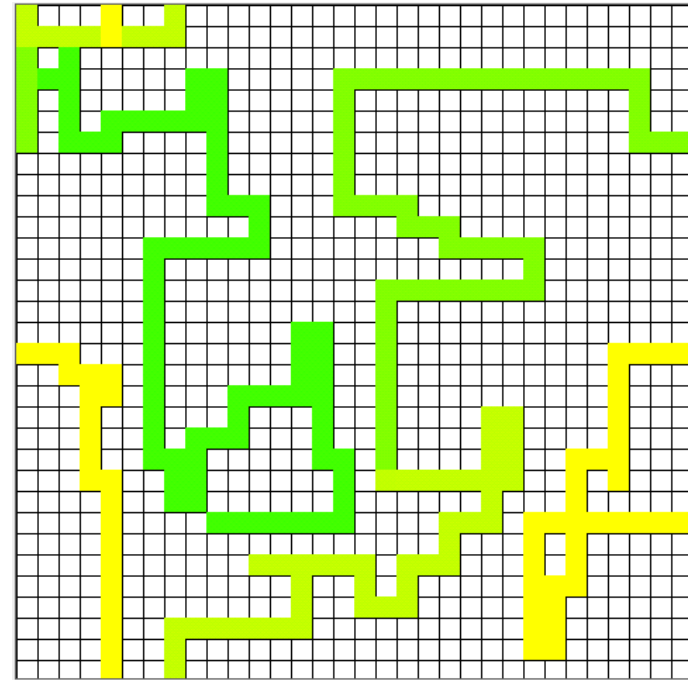
- Funkcje i terminale związane są z komunikacją między agentami;
- Przekazywanie informacji o polach - „celach”;
- Wnioski:
 - Znaczna poprawa działania;
 - Rozwiązanie problemów z dynamicznym środowiskiem;
 - Komunikacja umożliwia wykorzystanie wiedzy.

Wyniki - Komunikacja

Sprawność najlepszych mrówek



Agent 1	14 dań
Agent 2	11 dań
Agent 3	11 dań
Agent 4	11 dań



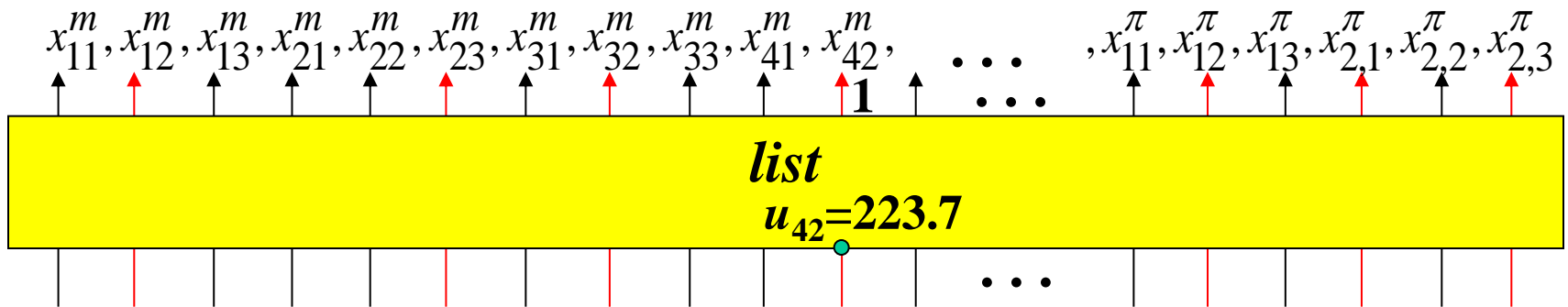
Random construction of an initial program

1. SET OF FUNCTIONS $\mathcal{F} = \{list, +, -, *, /\}$ $Z = [M, 2, 2, 2, 2]$

where *list* – a procedure with M real input parameters and M binary output parameters, $list: \mathcal{R}^M \rightarrow \{0,1\}^M$

$$x_m = \begin{cases} 1 & \text{if } u_m > 0 \\ 0 & \text{if } u_m \leq 0 \end{cases} \quad u_m \in \mathcal{R}, m = \overline{1, M}$$

- The root of each program tree is the procedure *list*
- *list* may be merely a root of a tree, not the other node in a tree
- M is the number of binary decision variables, $M=I(V+J)$



Random construction of an initial program

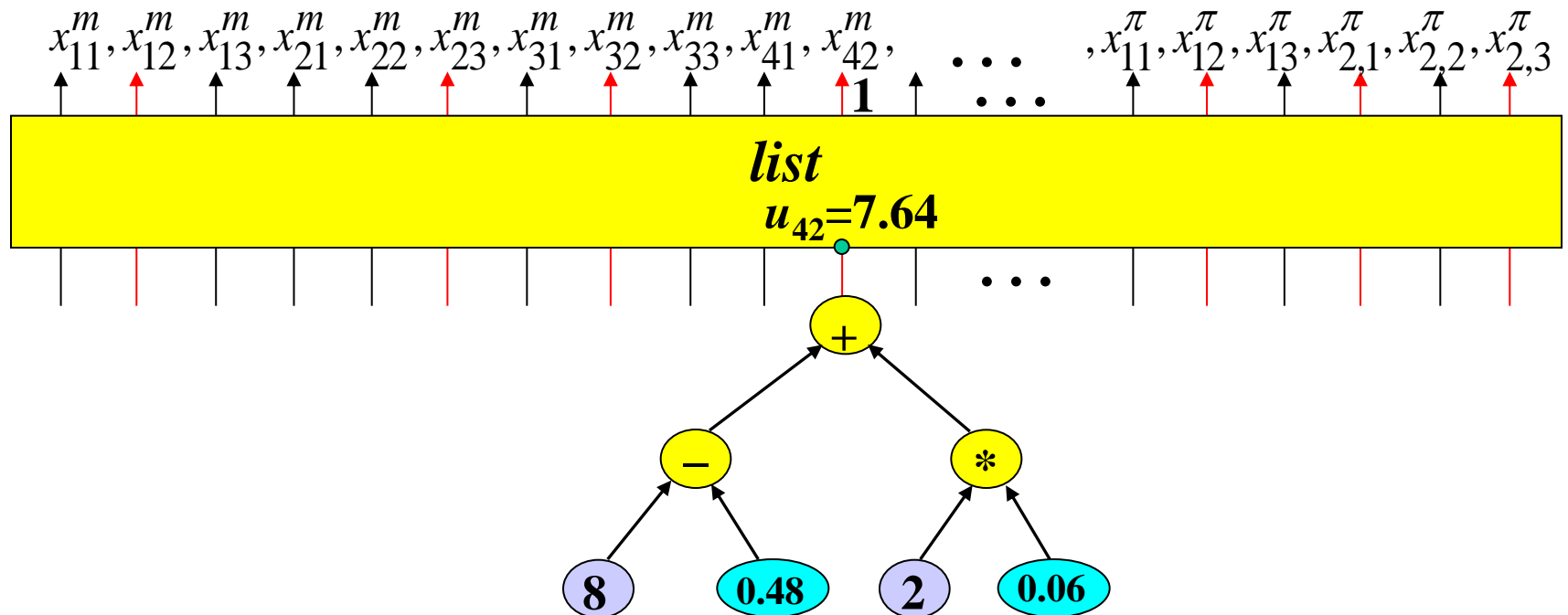
2. SET OF TERMINALS $\mathcal{T} = \mathcal{T}' \cup Q$

Let set of data \mathcal{T}' consists of elements from T, τ, κ, θ

Q – set of n random numbers, $n = |\mathcal{T}'|$

3. The other nodes of tree are generated, as follows:

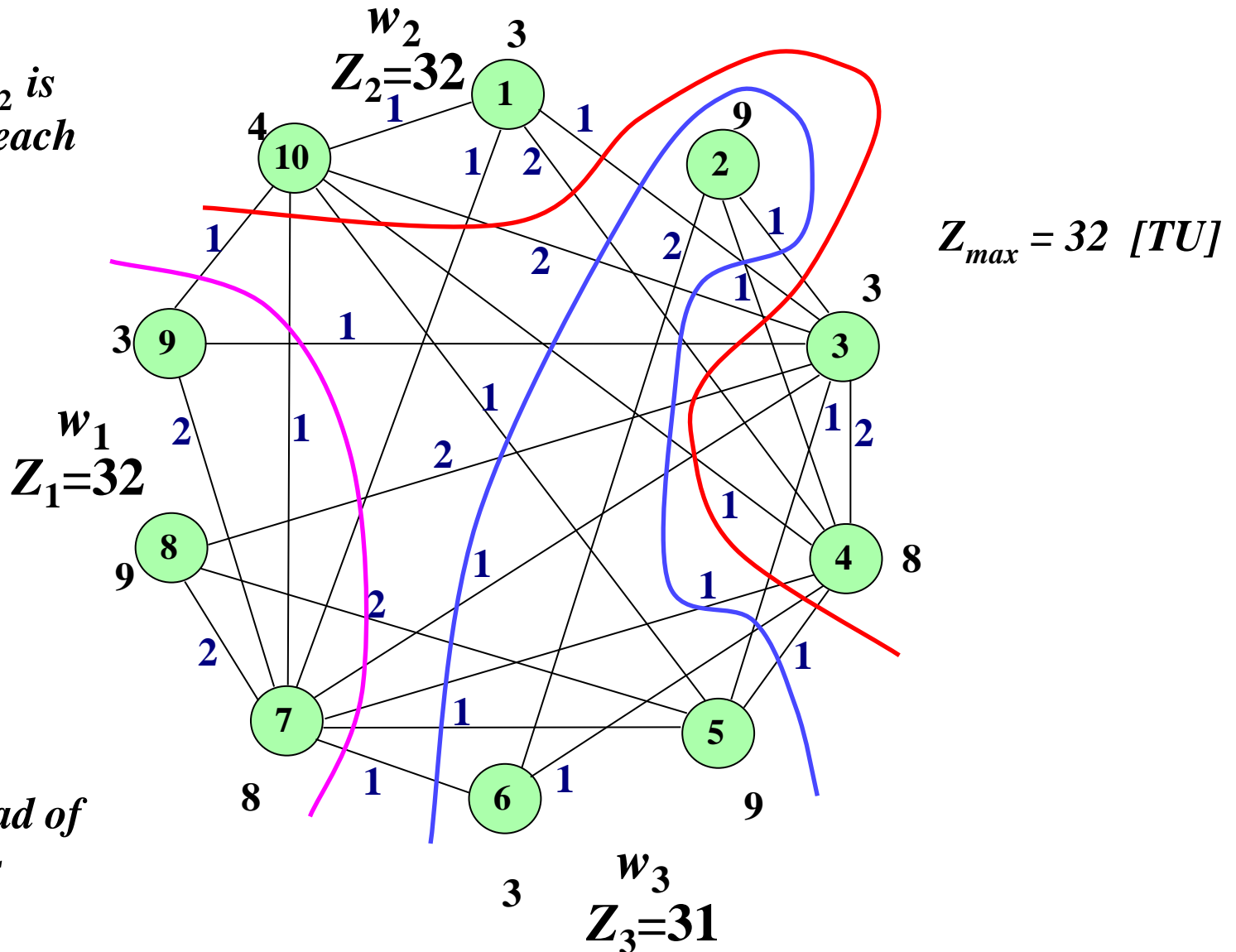
- name of function with the probability 0.5
- element of data matrix or vector with the probability 0.25
- Random number from the period $[0; 1]$ with the probability 0.25



TEST INSTANCE

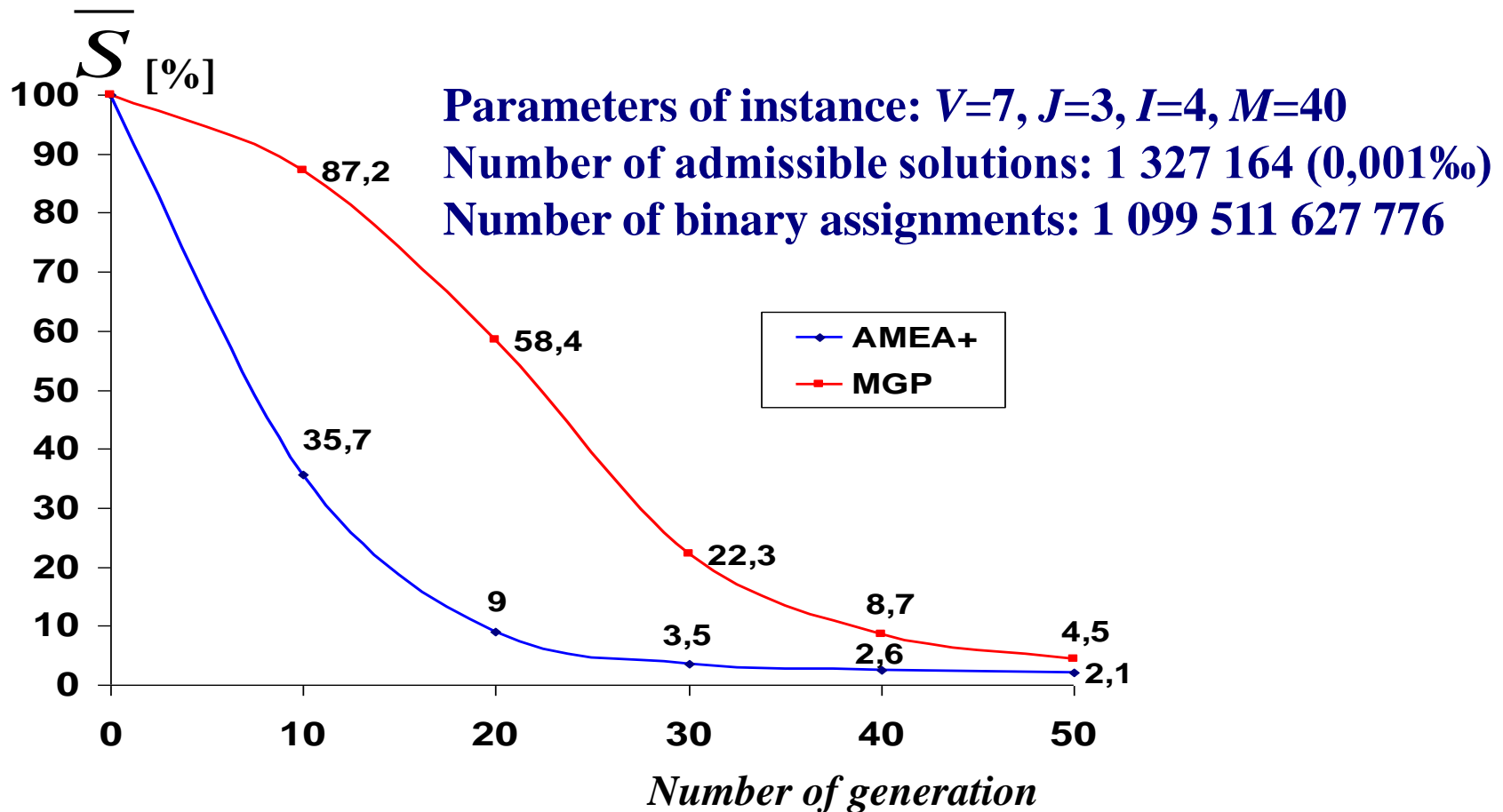
task assignment for the balanced load

Computer π_2 is assigned to each node



Balanced load of workstations

MINIMIZATION OF AN AVERAGE CONVERGENCE DEGREE TO THE PARETO FRONT



Parameters of *MGP*: $L=50, p_m=10^{-5}$

2 500 programs have been designed with at least $K=7\ 840$ nodes in a tree (196 nodes in a subtree)

Parameters of *AMEA+*: $L=50, p_m=0.15$

Building a Better Mouse

- Design a program for navigating a mouse in a maze;
- What are terminal and function sets?

Function Set =

{If-Movement-Blocked, While-Not-At-Cheese}

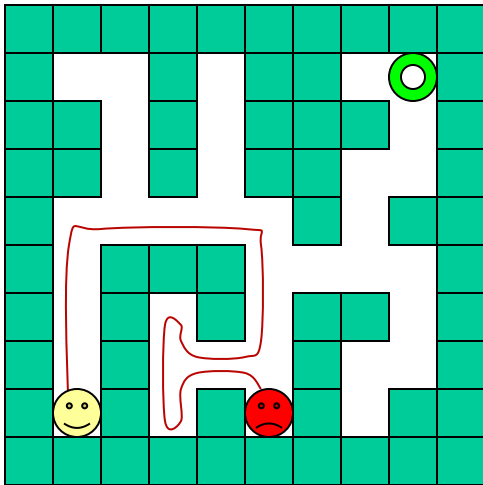
Terminal Set =

{Move-Forward, Turn-Left, Turn-Right}

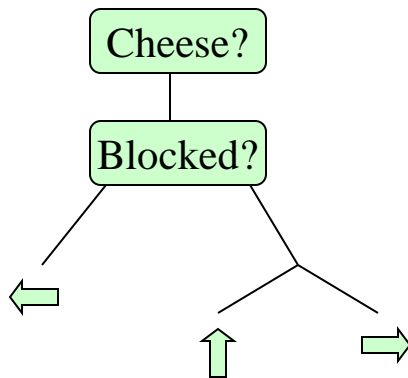
While-Not-At-Cheese will be used exclusively as the root node of the parse tree

Building a Better Mouse

A fitness function:



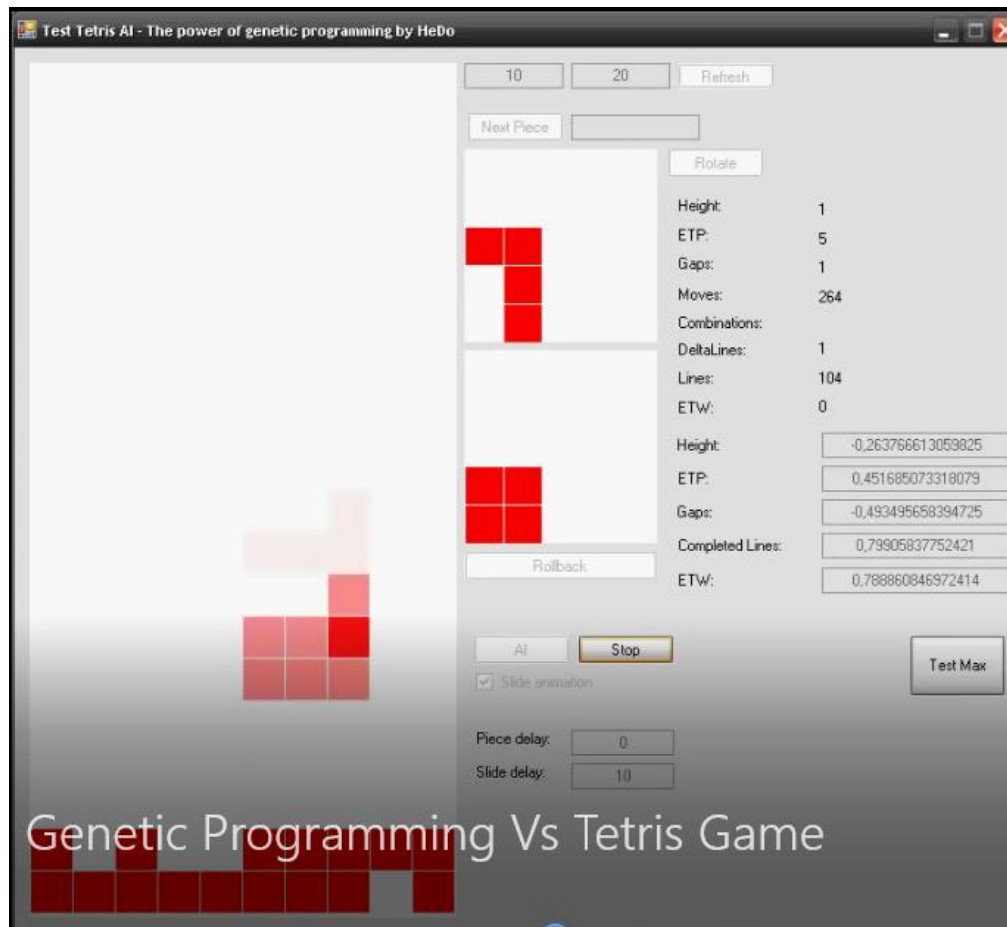
- Each function and terminal other than the root node costs one unit to execute;
- If the mouse spends more than 100 units, it dies of hunger/lack of energy;
- The fitness measure for a program is determined by executing the program, then **squaring the sum of the total units spent and the final distance from the exit**;
- A lower fitness measure is preferable to a higher fitness measure;



If a mouse dies 10 moves from the exit after spending 100 units, the fitness of the mouse control program is **12,100**

Artificial player in Tetris

More than 2kk+ pieces are required to defeat the GP strategy;



See it on You Tube

<https://www.youtube.com/watch?v=nkIY5IFZAY>

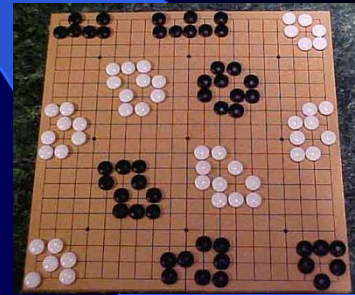


2. Potencjalne zastosowania GP



AlphaGo

- **Inteligentny system rozproszony AlphaGo firmy Google DeepMind z Londynu wygrywa 4:1 z Lee Sedolem - mistrzem świata w Go – w marcu 2016 r.**
- **Ke Jie nr 1 w grę Go pokonany w maju 2017 r.**
- **Od czerwca 2017 projekt został zakończony**
- **Kombinacja algorytmów:**
 - **Sztuczne sieci neuronowe,**
 - **Uczenie maszynowe;**
 - **Uczenie ze wzmocnieniem - gry treningowe z ludźmi i programami, w tym z samym sobą;**
 - **Tree search;**
 - **Monte Carlo.**





AlphaGo

Architektura i wydajność

Configuration	Search threads	No. of CPU	No. of GPU	Elo rating
Asynchronous	40	48	1	2,151
Asynchronous	40	48	2	2,738
Asynchronous	40	48	4	2,850
Asynchronous	40	48	8	2,890
Distributed	12	428	64	2,937
Distributed	24	764	112	3,079
Distributed	40	1,202	176	3,140
Distributed	64	1,920	280	3,168

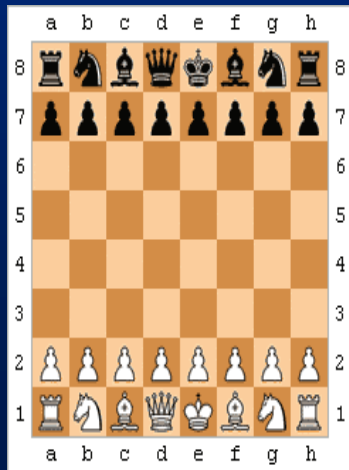


AlphaGo

Wersja	Architektura	Elo	Mecze
AlphaGo Fan	rozproszona	prawie 3,000	5:0 Fan Hui
AlphaGo Lee	50 TPUs, rozproszona	ok. 3,750	4:1 Lee Sedol
AlphaGo Master	CPU wraz z TPU v2	ok. 4,750	60:0 najlepsi gracze, maj 2017;

Programy do gry w szachy

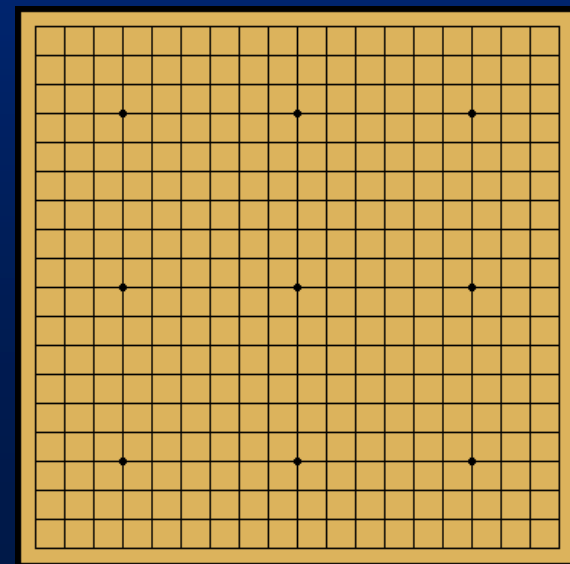
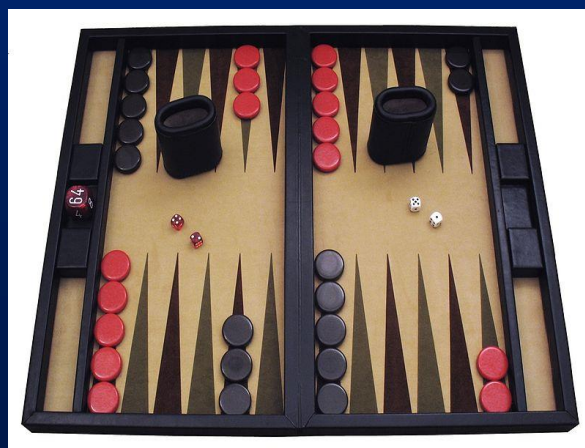
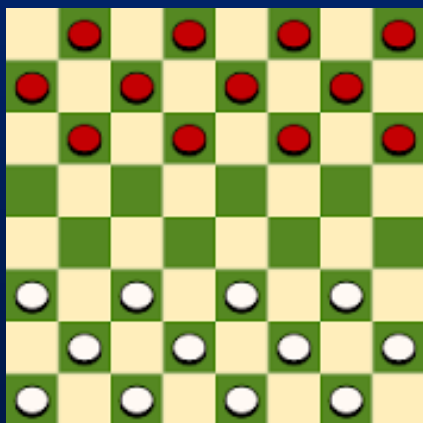
- 1959 - pierwszy program do gry w szachy (Samuel –1959);
- 1997 – **Deep Blue** na superkomputerze (32 procesory IBM RS6000/SP2 + 256 ASIC) wygrał z mistrzem świata G. Kasparowem 3.5:2.5;
- 2002 – program **Deep Fritz** na PC remisuje z V. Kramnikiem;
- 2003 – remis Kasparowa z X3D **Deep Fritz junior** (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.



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...*

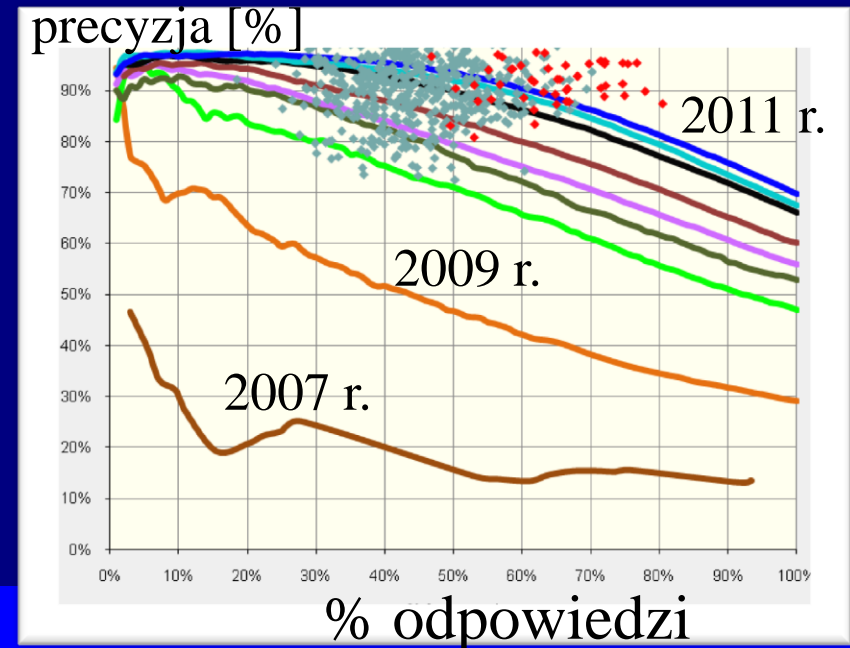
Scott: How many legs does a camel have?

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

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

IBM Watson

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



Emily Dickinson		99%	
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 .

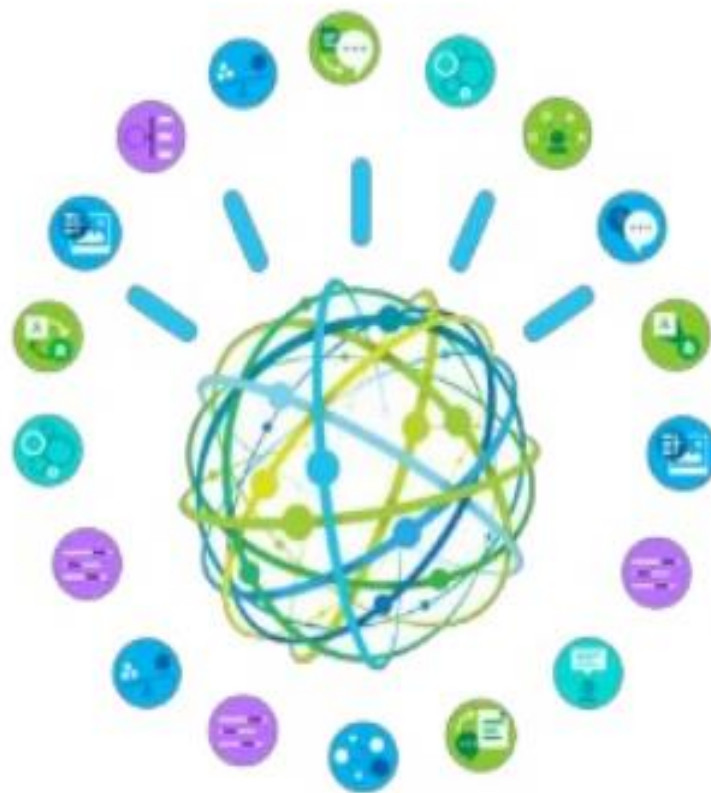
IBM Watson –zastosowania

- **Analiza biznesowa;**
- **Wyszukiwanie w rozległych bazach wiedzy;**
- **Wspomaganie podejmowania decyzji;**
- **Diagnostyka medyczna (onkologia);**
- **Edukacja.**

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



Sunway TaihuLight

93 PF/s; 15,4 MW; 5 ton

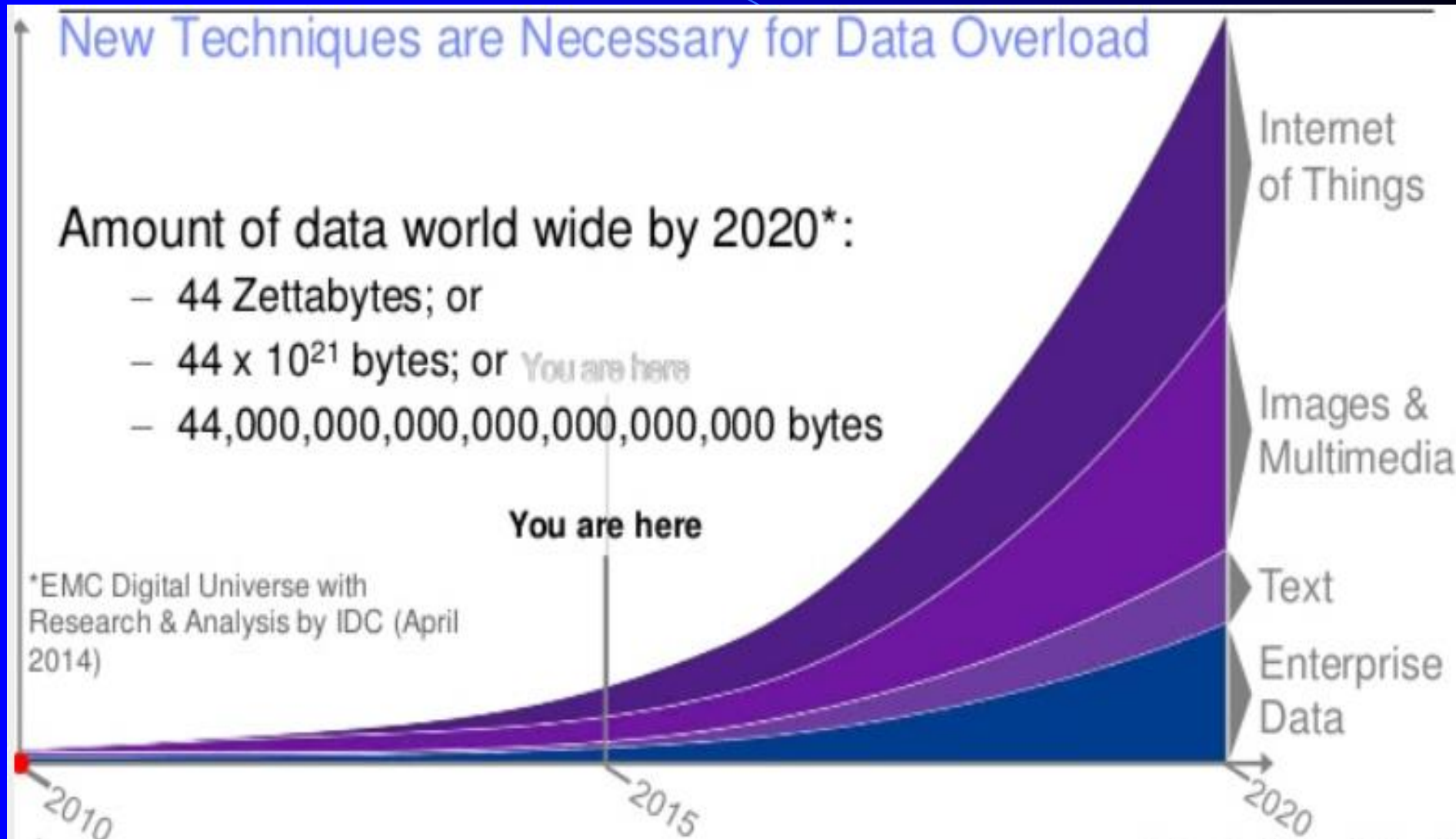


China's National Research Center of Parallel Computer Engineering & Technology (NRCPC)

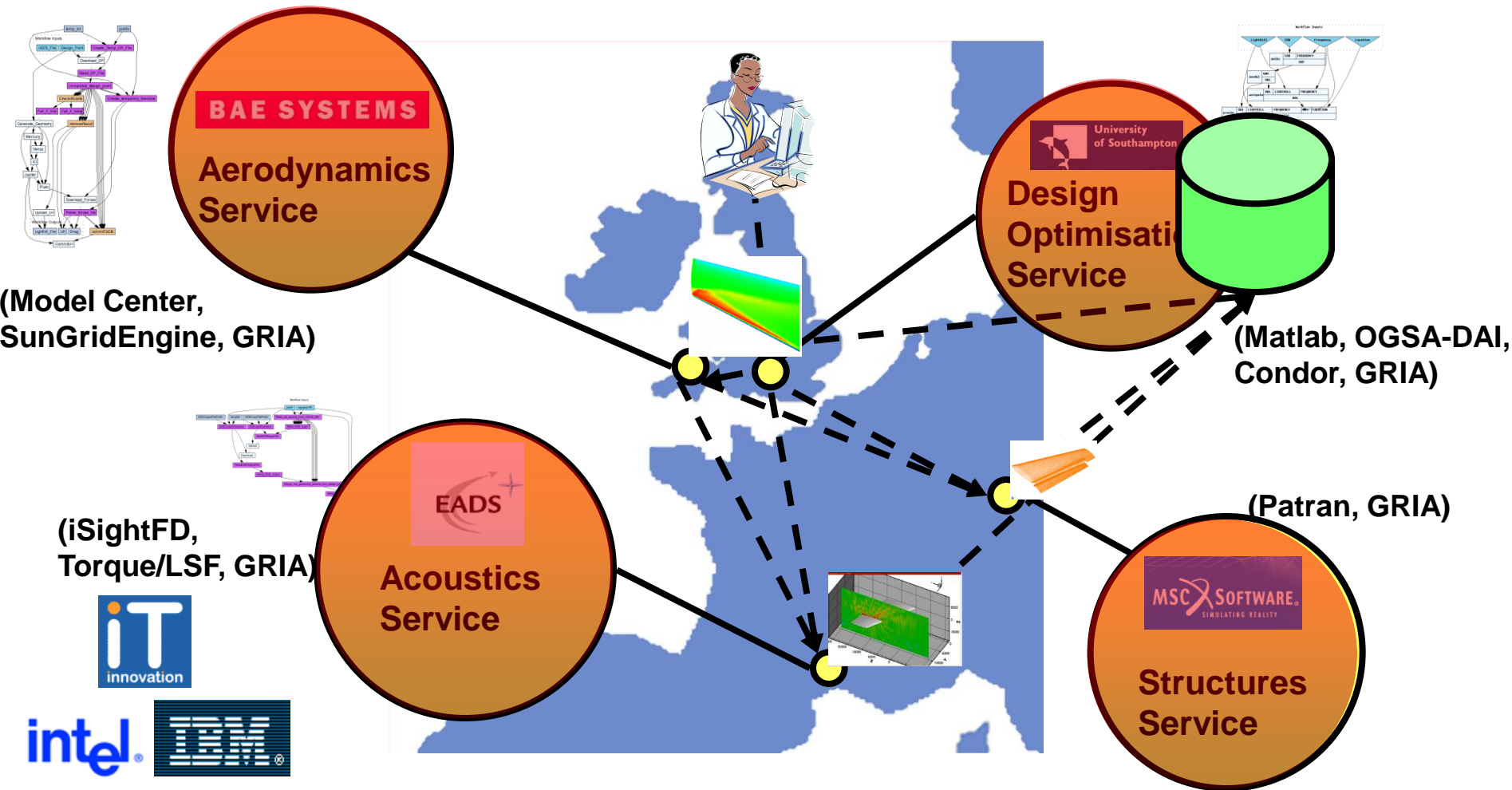
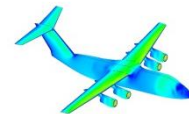
Installed at the National Supercomputing Center in Wuxi

Procesory	Sunway SW26010 260C 1,45 GHz 260-core RISC 64 bity (łącznie 10 649 600 rdzeni)
RAM	1,3 PB
Sieć	Sunway
HDD	20 PB (256 węzłów I/O, 65 serwerów HDD)
Szafy	170

Wykładniczy wzrost intensywności przetwarzania danych



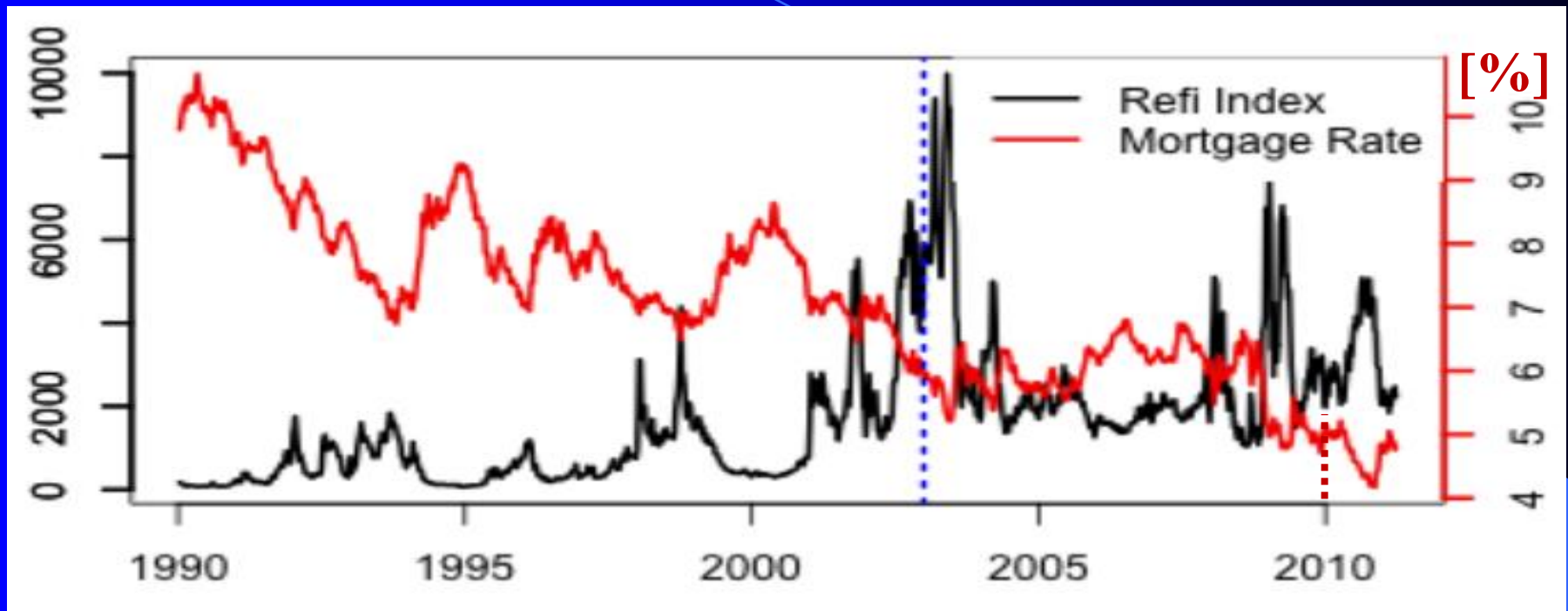
Typowy scenariusz biznesowy



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,
- **Nauki przyrodnicze, modelowanie molekularne;**
- Kryptologia;
- **Zaawansowane animacje graficzne;**
- Analiza danych geologicznych;
- **Zarządzanie energią;**
- Poszukiwanie ropy i gazu łupkowego;
- **Astronautyka;**
- Badania farmaceutyczne;
- **Wzornictwo przemysłowe.**

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

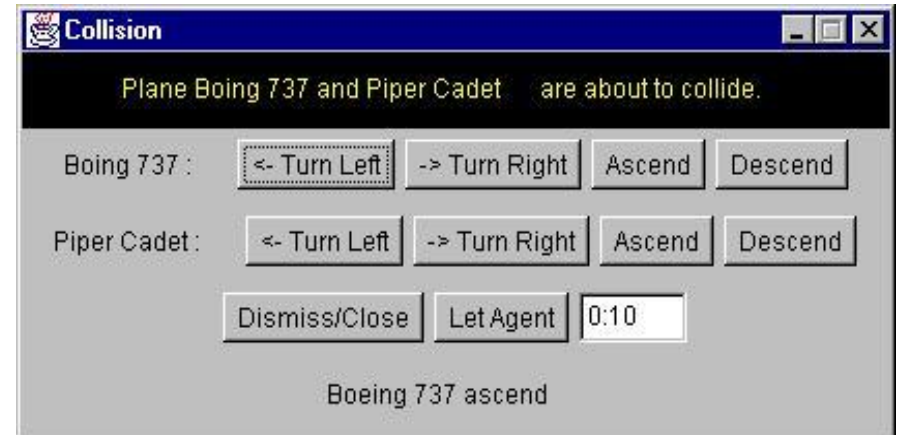
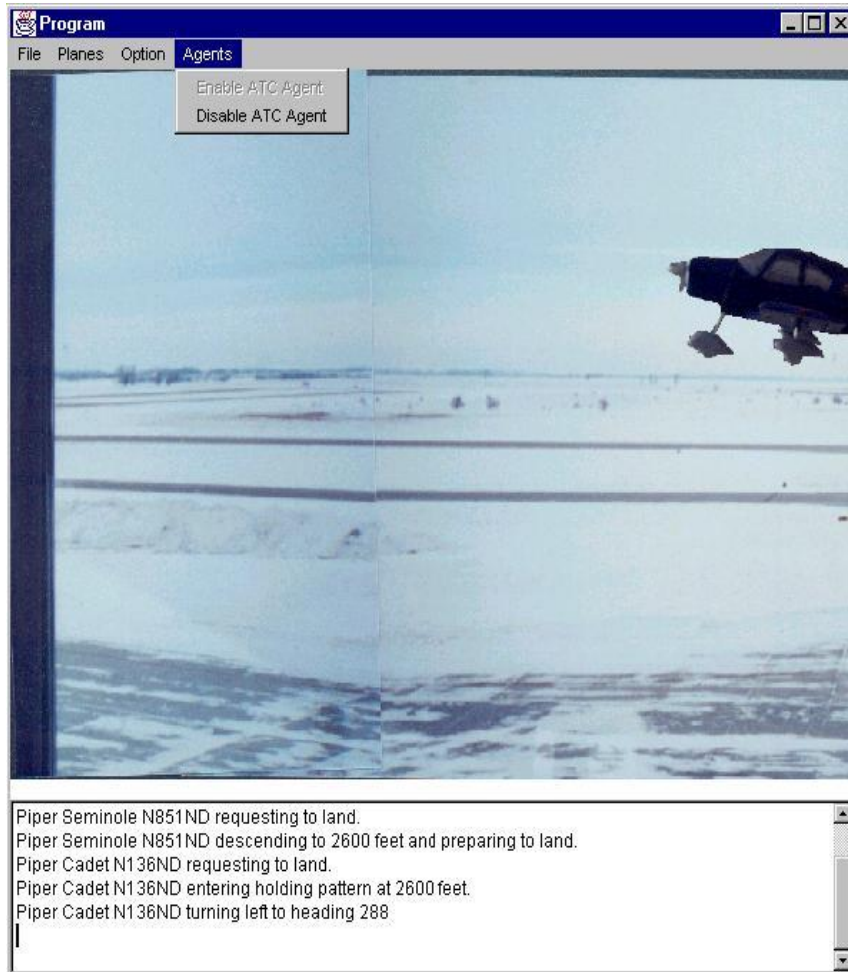


- 10 000 neuronów -> 100 000 000 000;

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* .

Rozbieżność między kontrolerem lotu a programistycznym agentem Air Traffic Control



Masdar city - niedaleko Abu Dhabi (ZEA)



Gospodarka oparta na wiedzy

Gospodarka przemysłowa

Gospodarka oparta na wiedzy

Status Quo

Robotnicy vs
Zarząd

Koszt

Lokalna/Krajowa

Ten sam rozmiar

Just in case

Odizolowana

Studia pięcioletnie

Dynamiczna

Współpraca

Zwrot z inwestycji

Globalna sieć

Na miarę

Just in time

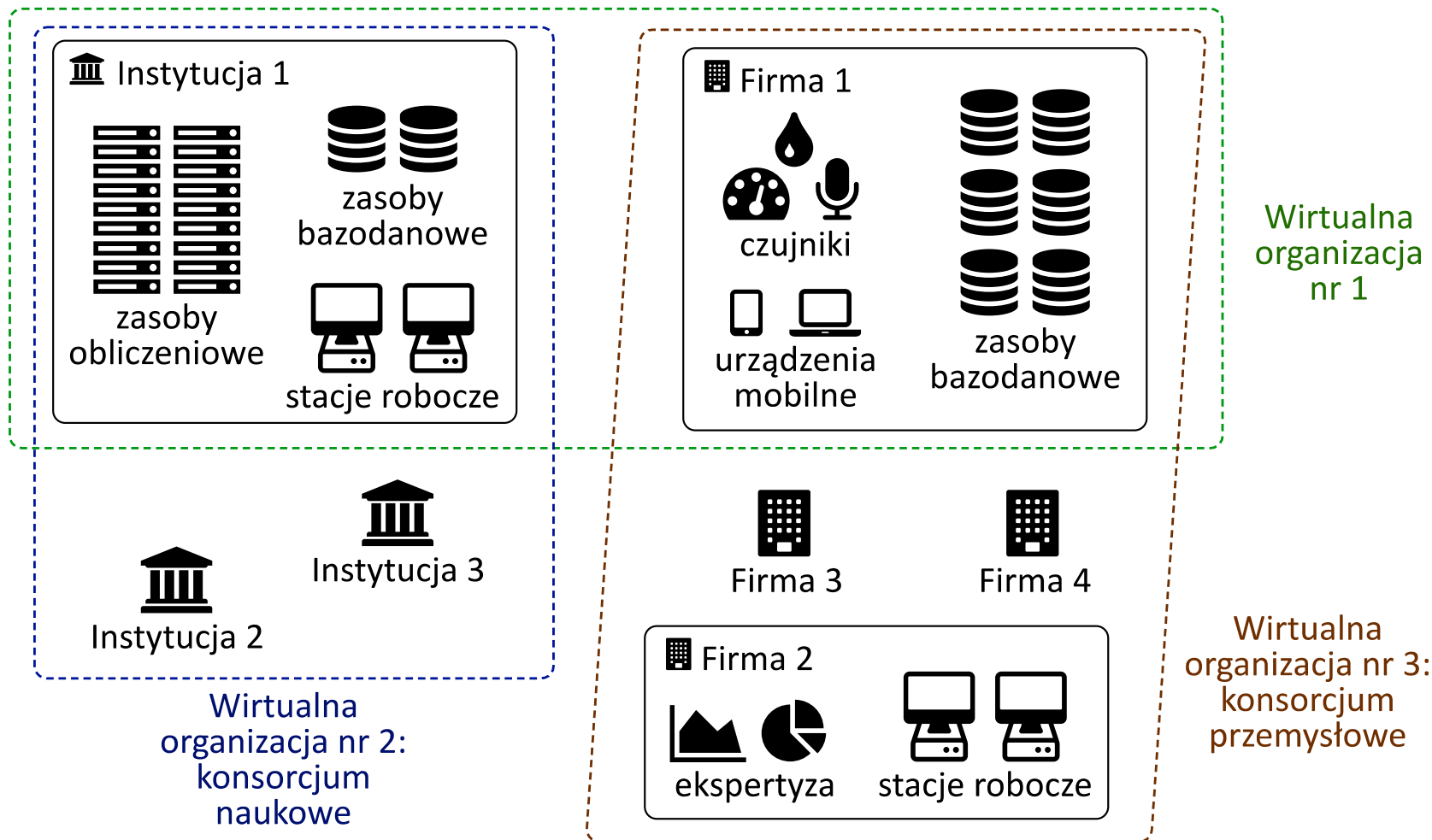
Wirtualne społeczności

Lifelong Learning



3. GP w optymalizacji i uczeniu maszynowym gridów i chmur

2. Charakterystyka wybranych gridów



Współdzielenie i koordynacja rozproszonych zasobów w wirtualnych organizacjach



Nazwa waluty.

Realizacja płatności na całym świecie.

Brak banku centralnego, operacje potwierdzane przez system.

Znikomy koszt opłat transferowych w porównaniu do tradycyjnej bankowości.

Brak możliwości wycofania operacji



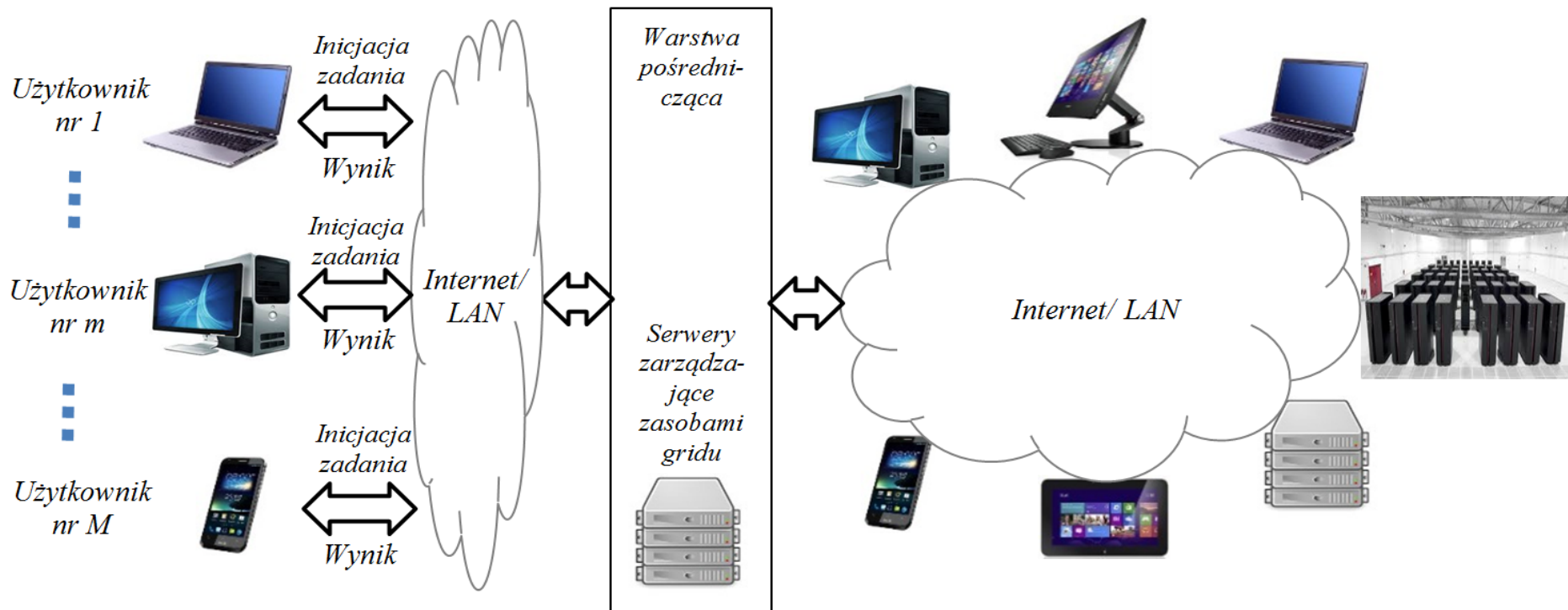
Ponadto:

- **najpotężniejszy GRID na świecie, którego prędkość wciąż wzrasta,**
- **system opiera się na architekturze P2P (brak centralnej jednostki zarządzającej),**
- **oprogramowanie typu *open-source*,**
- **system wykorzystuje mechanizmy kryptografii.**

Wybrane platformy gridowe (ang. *middleware*)

- ***Globus Toolkit;***
- ***BOINC;***
- ***gLite;***
- ***ARC Advanced Resource Connector;***
- ***UNICORE;***
- ***dCache;***
- ***Comcute PG middleware.***

Infrastruktura gridu

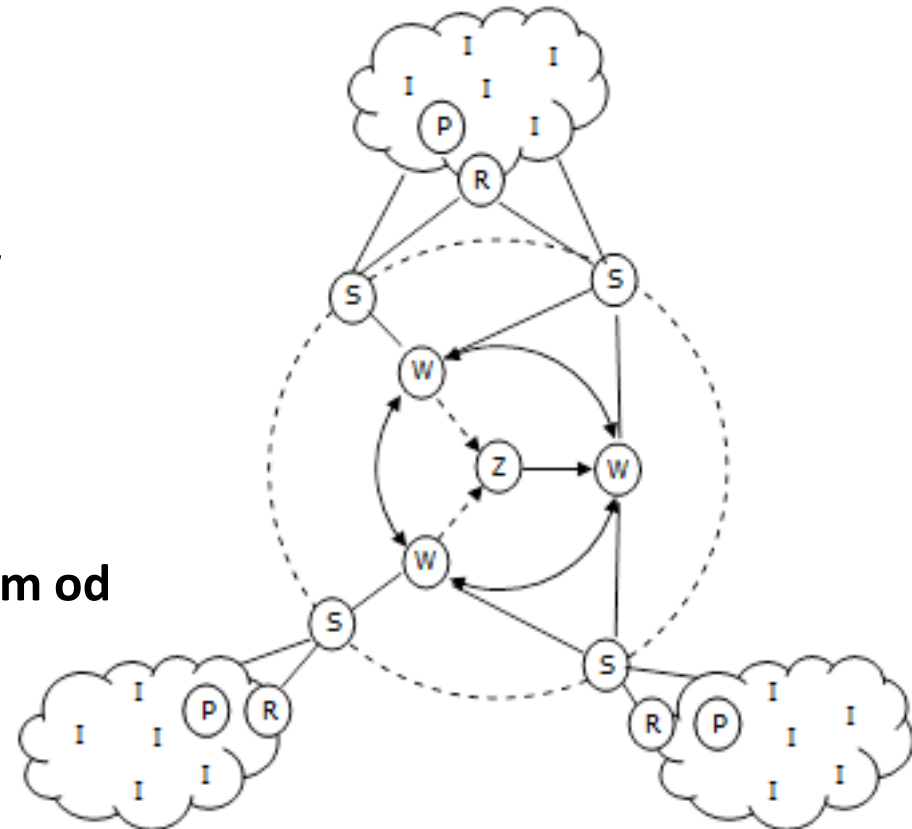


Grid jest rozproszonym systemem komputerowym, w którym implementuje się wirtualne środowisko obliczeniowe dla pewnej organizacji zrzeszającej użytkowników potrzebujących usług informatycznych o odpowiedniej jakości.

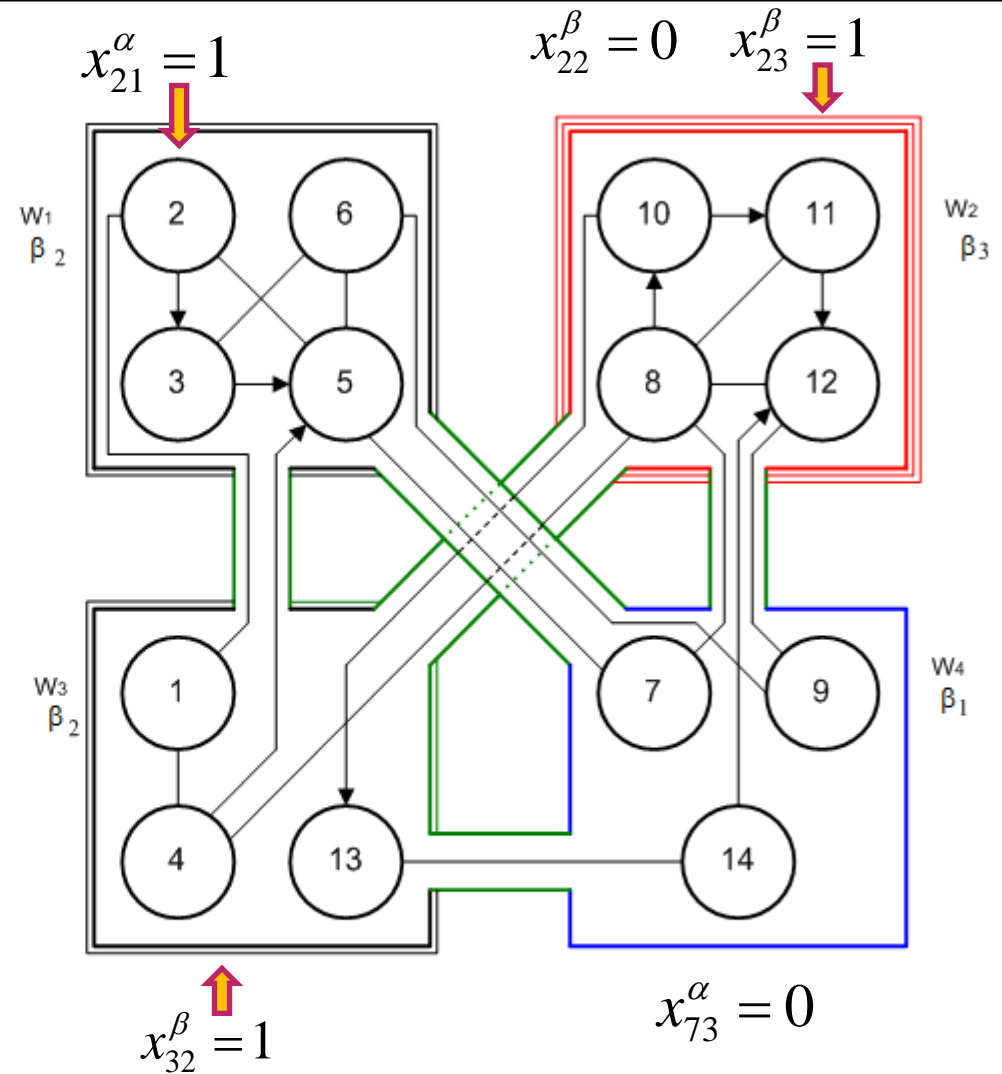
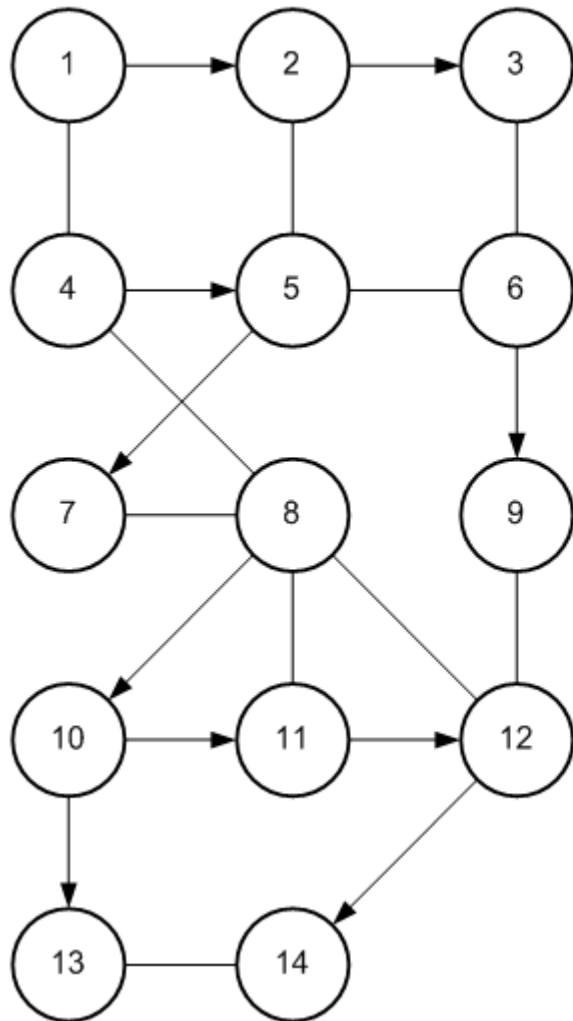
Zarządzanie zasobami odbywa się w wielu domenach z wykorzystaniem uzgodnionego zestawu protokołów i powszechnie dostępnych interfejsów.

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



Konfiguracja zasobów w gridzie



Klasyfikacja problemów optymalizacji konfiguracji zasobów w gridzie

<i>Założenie</i>	<i>Atrybut</i>		
Ograniczoność zasobów	<ul style="list-style-type: none"> - ograniczone - nieograniczone 		
Graf opisujący strukturę komunikacyjną między modułami	<ul style="list-style-type: none"> - dowolny - szczególny: <ul style="list-style-type: none"> ◆ drzewo ◆ sekwencyjno - współbieżny ◆ potok 		
Struktura topologiczna sieci komunikacyjnej	<ul style="list-style-type: none"> - dowolna - szczególna: 		
	tandem	łańcuch	gwiazda

Klasyfikacja problemów optymalizacji konfiguracji zasobów w gridzie

<i>Założenie</i>	<i>Atrybut</i>
Liczba kryteriów	<ul style="list-style-type: none">• Jedno kryterium• Optymalizacja wielokryterialna:<ul style="list-style-type: none">– dominujące– niezdominowane– hierarchiczne– kompromisowe

Comcute – wersja rozszerzona

- **W gridzie Comcute wprowadzono paradygmat agentowy;**
- **Agenty zrealizowano jako maszyny wirtualne działające w środowisku chmury prywatnej OpenStack;**
- **Istniejące moduły programistyczne przeniesiono do maszyn wirtualnych zgodnie z przeznaczeniem agentów;**
- **Środowisko OpenStack umożliwia migrację agentów między węzłami fizycznymi systemu i komunikację pomiędzy agentami niezależnie od ich ulokowania.**

OpenStack

Oprogramowanie do budowy chmury prywatnej:

- **Pozwala wykorzystać zalety chmury obliczeniowej w obrębie własnej infrastruktury;**
- **Kontrola nad serwerami;**
- **Wymagania w zakresie bezpieczeństwa danych.**

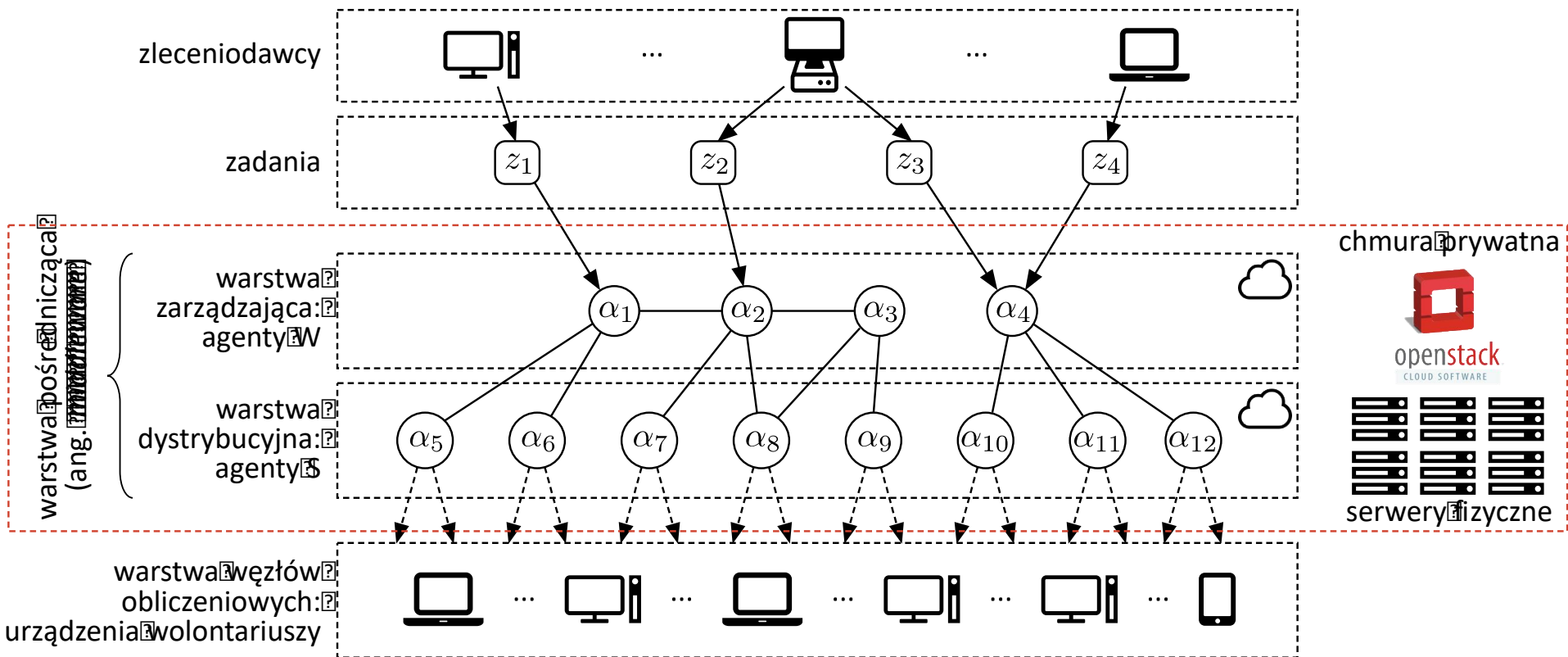
Wykorzystywane przez wiele firm na całym świecie:

- **Volkswagen Group;**
- **China Mobile;**
- **AT&T;**
- **Walmart;**
- **Bloomberg;**
- **PayPal;**
- **Santander Bank;**
- **State Grid Corporation of China (największa firma energetyczna na świecie).**



Comcute – wersja rozszerzona

Możliwe do zagwarantowania parametry wykonania zadań wynikają z powiązań pomiędzy agentami warstwy pośredniczącej.



Środowisko agentowe w chmurze Comcute

Agenty w gridzie Comcute cechują się różnym poziomem inteligencji:

- **Agenty klasy W organizują się w zespoły robocze do realizacji zadań obliczeniowych;**
- **Agenty klasy S negocjują z komputerami internautów możliwość ich wykorzystania do obliczeń gridowych;**
- **Agenty typu W i S migrują autonomicznie pomiędzy węzłami w celu optymalizacji wykorzystania zasobów zgodnie z przyjętą strategią zarządzania gridem;**
- **Agenty-solwery do samoopptymalizacji gridu wyznaczają strategię działania agentów klasy W i S;**
- **Potencjalne agenty pomocnicze: zbierają dane o zasobach w gridzie lub rekonfigurują węzły gridu.**

4. Model optymalizacji strategii zespołu agentów warstwy zarządzającej systemu Comcute

Podział funkcji warstwy pośredniczącej pomiędzy agenty pozwala na:

- **Rozproszenie systemu;**
- **Zapewnienie redundancji;**
- **Skalowanie systemu w zależności od obciążenia;**
- **Efektywniejsze wykorzystanie dostępnych zasobów.**

Strategia zespołu inteligentnych agentów obejmuje:

- **Wybór zadań użytkowników przez agenty;**
- **Określenie miejsc migracji agentów;**
- **Wyznaczenie konfiguracji zasobów niezbędnych do działania agentów.**

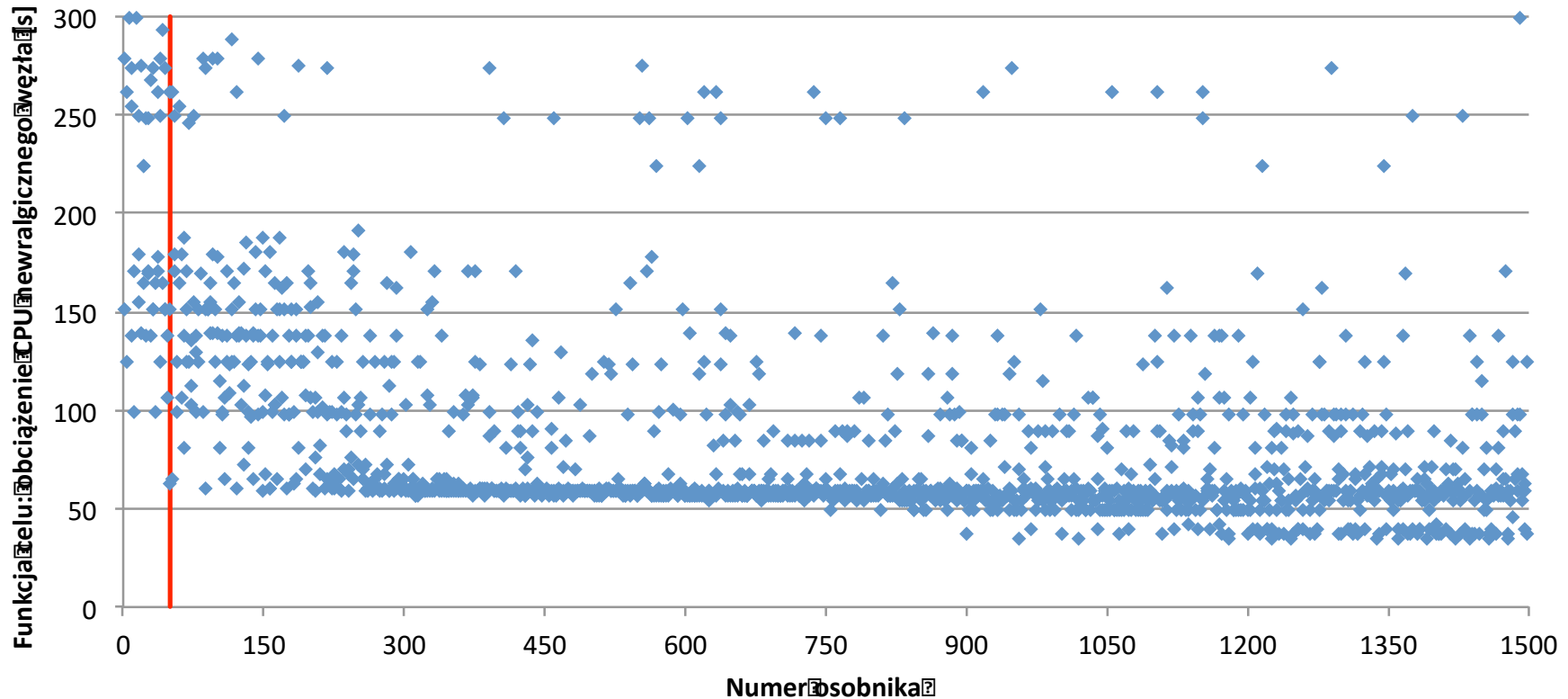
Kryteria oceny jakości strategii zespołu agentów

1. \hat{Z}_{\max} – obciążenie procesorów newralgicznego komputera [s];
2. \tilde{Z}_{\max} – obciążenie komunikacyjne newralgicznego komputera [s];
3. \hat{Z}_{suma} – łączne obciążenie procesorów w gridzie [s];
4. \tilde{Z}_{suma} – łączne obciążenie komunikacyjne w gridzie [s];
5. Θ – łączna wydajność gridu, np. [FLOPS];
6. Ξ – koszt zakupu komputerów [JM];
7. K – koszt przetwarzania danych i komunikacji [JM];
8. η – stopień rozproszenia modułów;
9. $\kappa_{\min}^{\text{RAM}}$ – wielkość dostępnej pamięci RAM w newralgicznym węźle [GB];

Kryteria oceny jakości strategii zespołu agentów

10. $\kappa_{\min}^{\text{HDD}}$ – wielkość dostępnej pamięci HDD w newralgicznym węźle [TB];
11. $\kappa_{\min}^{\text{SSD}}$ – wielkość dostępnej pamięci SSD w newralgicznym węźle [TB];
12. M^{RAM} – wielkość łącznej rezerwy RAM [GB];
13. M^{HDD} – wielkość łącznej rezerwy HDD [TB];
14. M^{SSD} – wielkość łącznej rezerwy SSD [TB];
15. E – łączny pobór mocy elektrycznej przez komputery gridu [W];
16. Γ – dostępność gridu;
17. \bar{T} – oczekiwany czas realizacji zadań [s].

Eksperyment: optymalizacja jednokryterialna

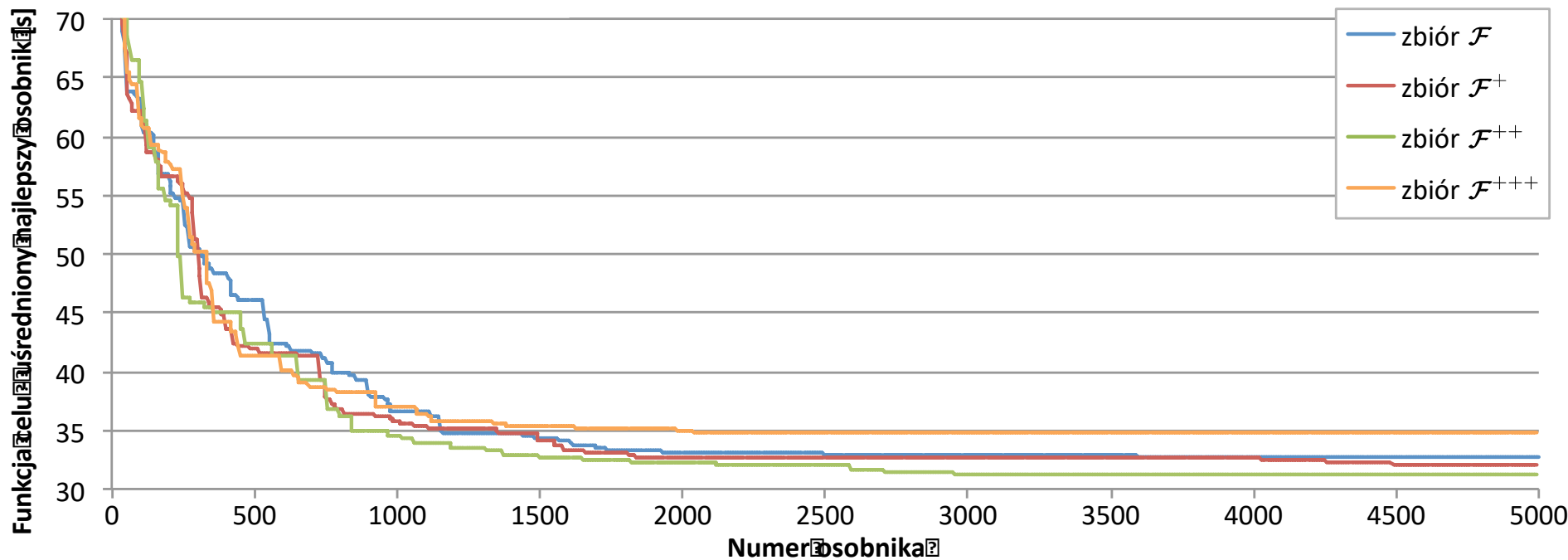


Oceny strategii wyznaczonych przez programowanie genetyczne
(30 początkowych epok)

Rozmiar populacji: 50 osobników

- **152 binarnych zmiennych decyzyjnych;**
- **Binarna przestrzeń przeszukiwań zawiera $5,7 \cdot 10^{45}$ elementów;**

Wpływ zbioru procedur na zbieżność programowania genetycznego



Oceny uśrednionych najlepszych osobników dla czterech zbiorów procedur

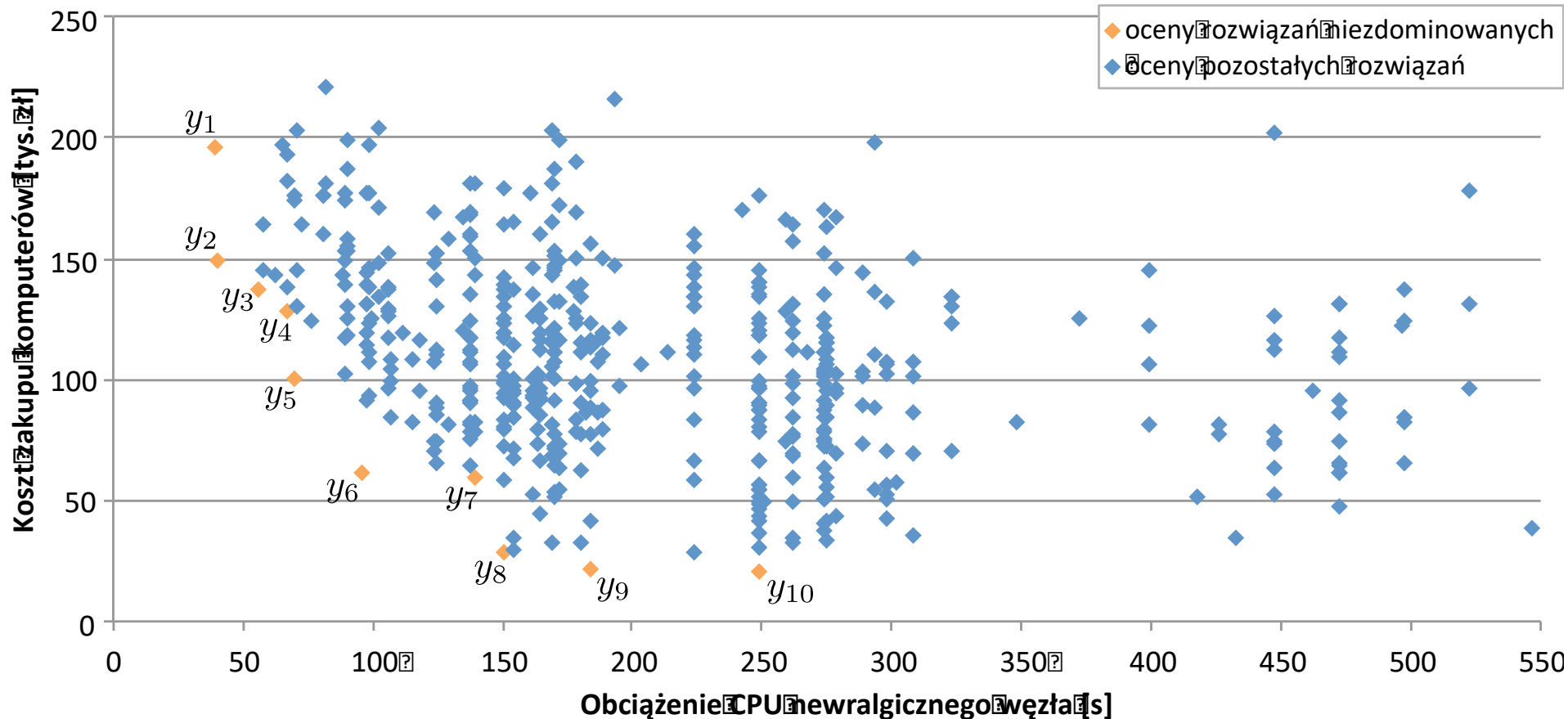
$$\mathcal{F} = \{+, -, *, /\}$$

$$\mathcal{F}^+ = \{+, -, *, /, IF, OR, AND, NOT\}$$

$$\mathcal{F}^{++} = \mathcal{F}^+ \cup \{WHILE, SET_{var_1}, \dots, SET_{var_{D_{var}}}, GET_{var_1}, \dots, GET_{var_{D_{var}}}\}$$

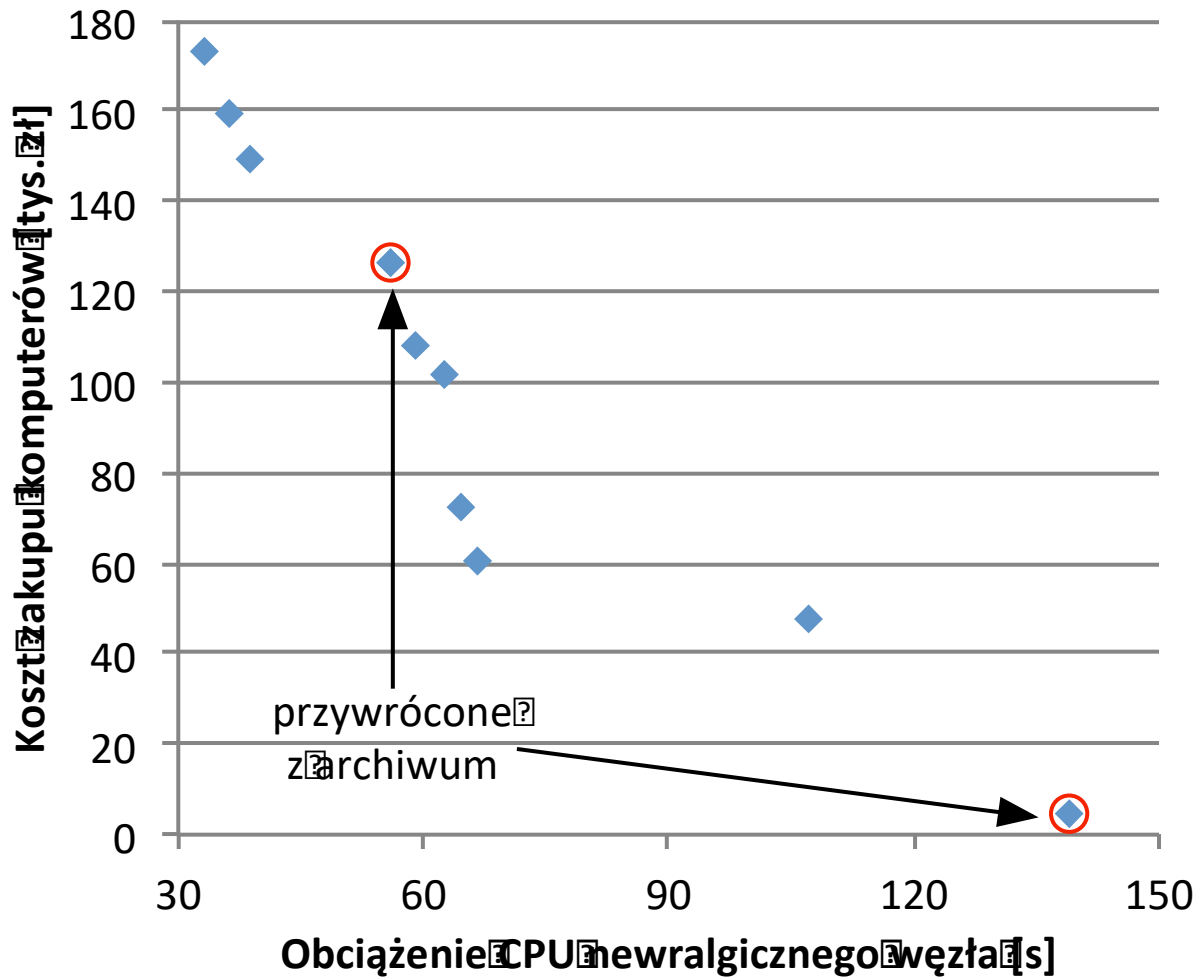
$$\mathcal{F}^{+++} = \mathcal{F}^+ \cup \{FOR, SET_{var_1}, \dots, SET_{var_{D_{var}}}, GET_{var_1}, \dots, GET_{var_{D_{var}}}\}$$

Sytuacja konfliktowa przy wyborze strategii



Wybrane oceny strategii zespołu 12 agentów realizujących dwa zadania obliczeniowe w gridzie o sześciu węzłach

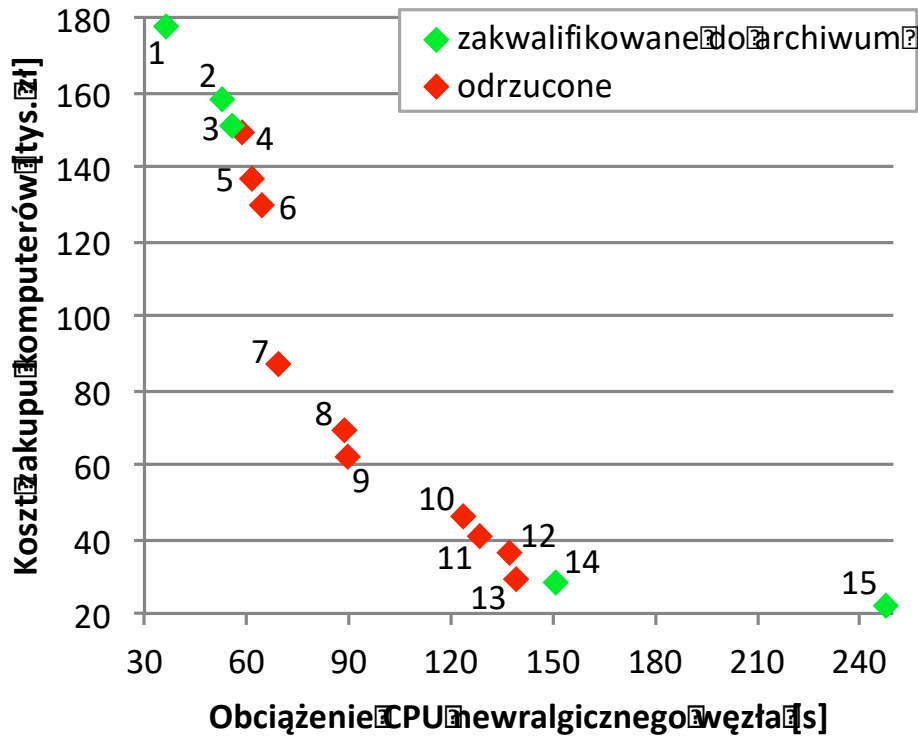
Przywrócenie rozwiązań z archiwum



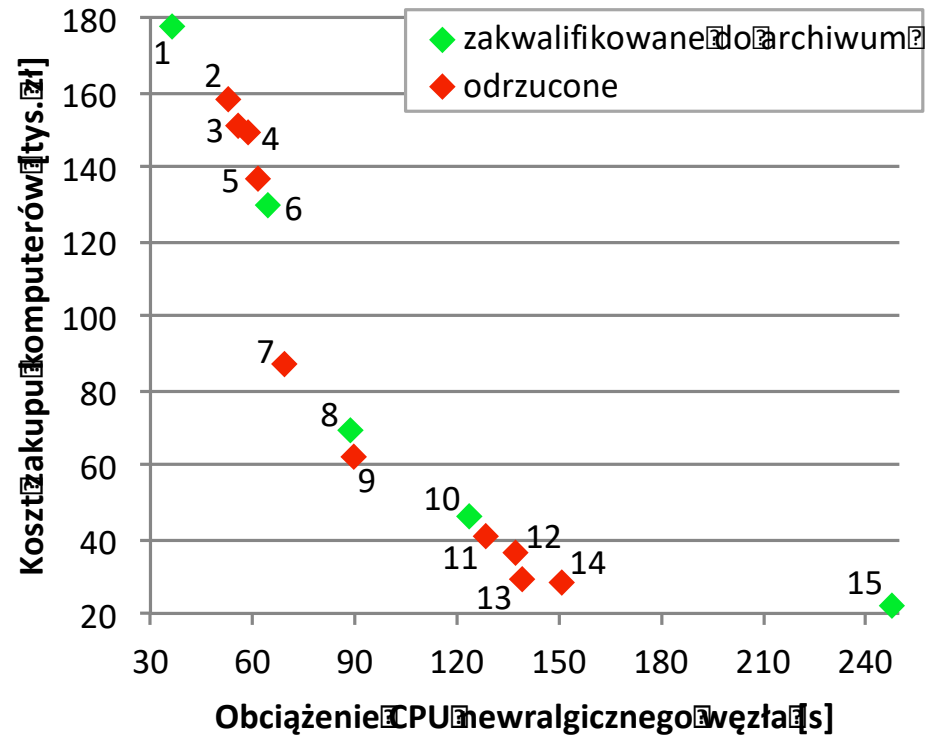
Oceny niezdominowane w generacji $g + \lambda_{epoch}$ (interwał weryfikacji)

Redukcja nadmiarowych rozwiązań na podstawie zagęszczenia

Wybór $\Lambda = 5$ spośród 15 rozwiązań kandydujących do zapisania w archiwum.



Na podstawie miary GCD



Na podstawie miary DGCD

AAG'16 – Aplikacja Administratora Gridu



Aplikacja Administratora Gridu

Aktualny scenariusz:

#2: Optymalizacja w ↕

Kokpit administratora

Zagadnienie

#2: Optymalizacja wielokryterialna gridu - algorytm MOGPA

Grid

Kryteria i ograniczenia

Konfiguracja fizyczna

Zadania i agenty

Programowanie genetyczne

Parametry ewolucji

Operatory i terminale

Kryteria optymalizacji

<input type="checkbox"/>	n	Kryterium cząstkowe F_n	Oznaczenie	Optymalizacja
<input checked="" type="checkbox"/>	1	Obciążenie procesorów newralgicznego węzła	\hat{Z}_{\max}	MIN
<input type="checkbox"/>	2	Obciążenie komunikacyjne newralgicznego węzła	\tilde{Z}_{\max}	MIN
<input type="checkbox"/>	3	Łączne obciążenie procesorów CPU w gridzie	\hat{Z}_{suma}	MIN
<input type="checkbox"/>	4	Łączne obciążenie komunikacyjne w gridzie	\tilde{Z}_{suma}	MIN
<input type="checkbox"/>	5	Łączna wydajność gridu	$\Theta(x)$	MAX
<input checked="" type="checkbox"/>	6	Koszt zakupu komputerów	Ξ	MIN
<input type="checkbox"/>	7	Koszt przetwarzania danych i komunikacji	$K(x)$	MAX
<input type="checkbox"/>	8	Stopień rozproszenia modułów	$\eta(x)$	MAX
<input type="checkbox"/>	9	Wielkość dostępnej pamięci RAM w newralgicznym hoście	$\kappa_{\min}^{\text{RAM}}(x)$	MAX
<input type="checkbox"/>	10	Wielkość dostępnej pamięci HDD w newralgicznym hoście	$\kappa_{\min}^{\text{HDD}}(x)$	MAX
<input type="checkbox"/>	11	Wielkość dostępnej pamięci SSD w newralgicznym hoście	$\kappa_{\min}^{\text{SSD}}(x)$	MAX
<input type="checkbox"/>	12	Wielkość łącznej rezerwy pamięci RAM	$M^{\text{RAM}}(x)$	MAX
<input type="checkbox"/>	13	Wielkość łącznej rezerwy pamięci HDD	$M^{\text{HDD}}(x)$	MAX
<input type="checkbox"/>	14	Wielkość łącznej rezerwy pamięci SSD	$M^{\text{SSD}}(x)$	MAX

AAG'16 – Aplikacja Administratora Gridu



Aplikacja Administratora Gridu

Aktualny scenariusz: #2: Optymalizacja w ↕

Kokpit administratora

Zagadnienie

#2: Optymalizacja wielokryterialna gridu - algorytm MOGPA

Grid

Kryteria i ograniczenia

Konfiguracja fizyczna

Zadania i agenty

Programowanie genetyczne

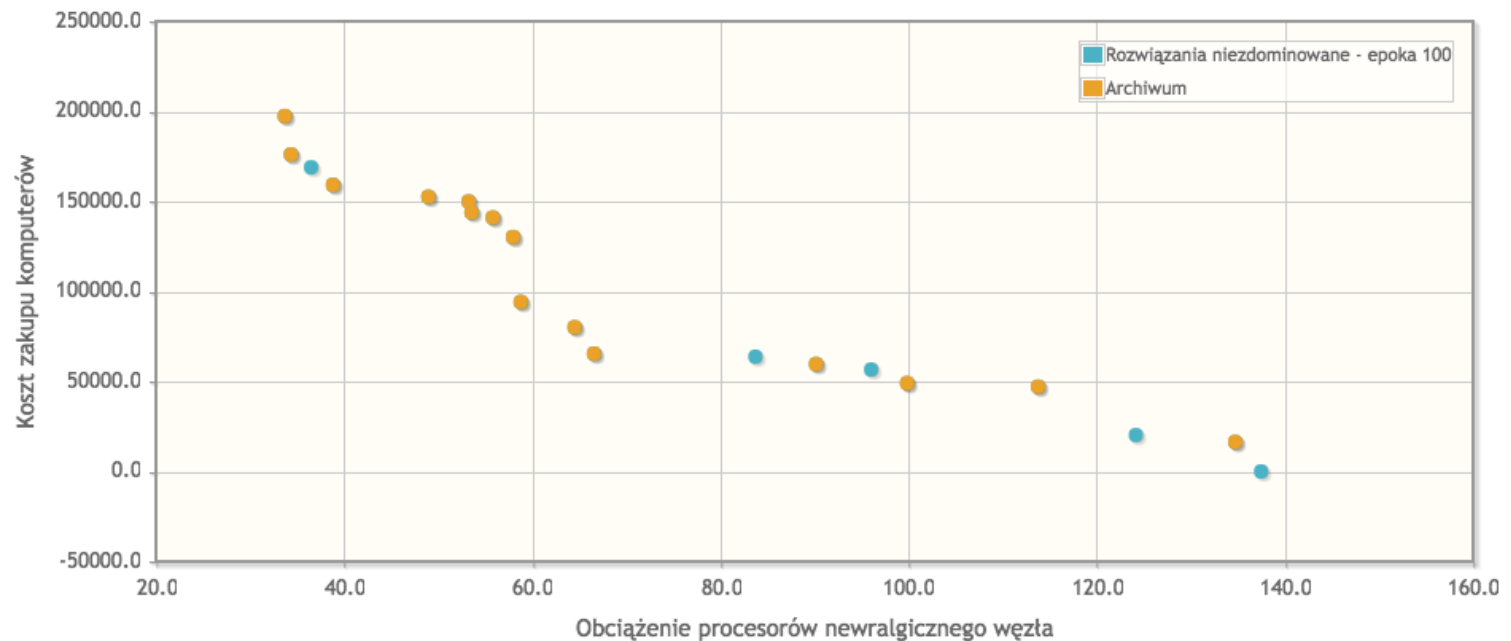
Parametry ewolucji

Operatory i terminale

Symulacja #175 #2: Optymalizacja wielokryterialna gridu - algorytm MOGPA

Rozwiązania niezdominowane

Szczegóły wybranego rozwiązania po kliknięciu na punkt wykresu



Epoki

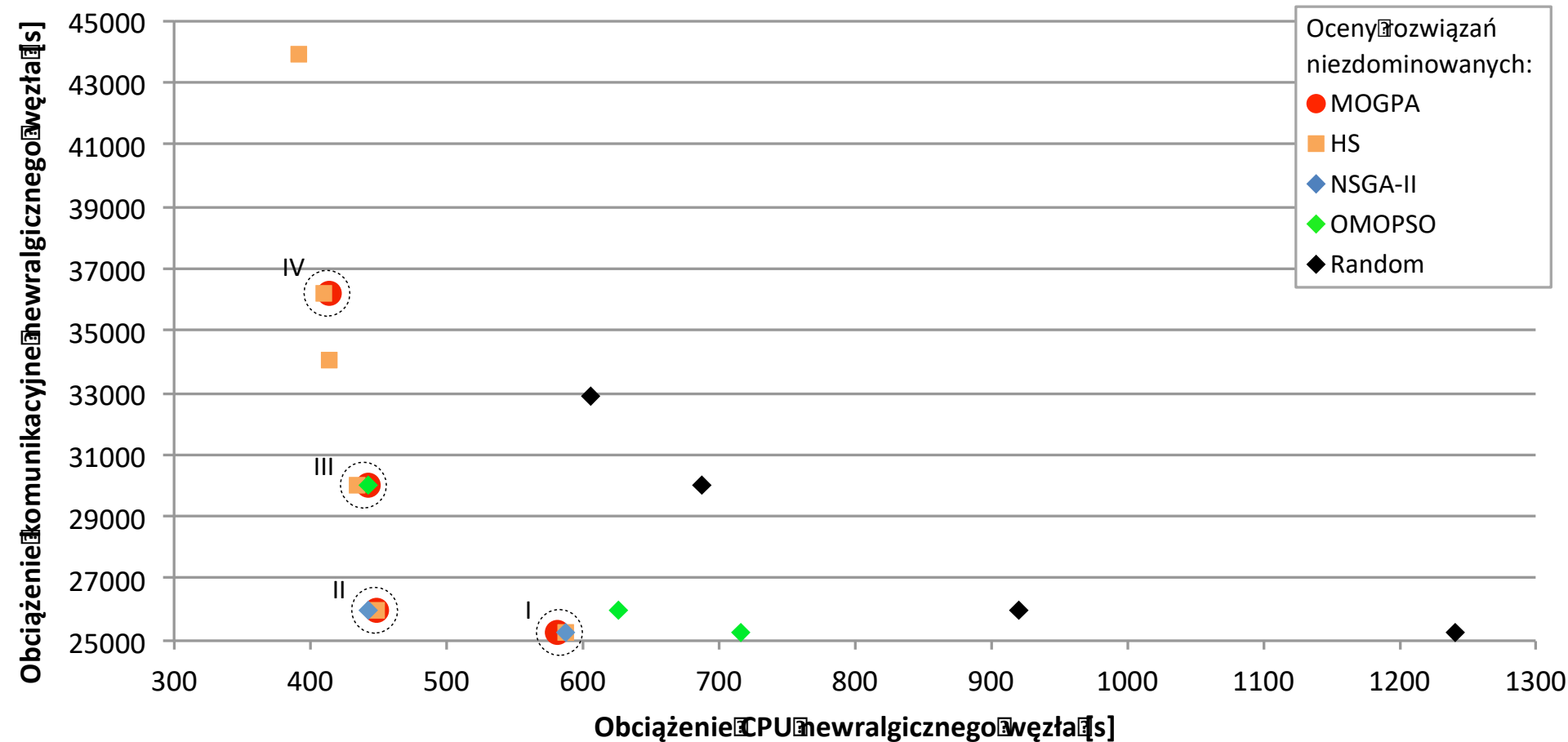


✓ Archiwum widoczne

Export

Reset zoom

Porównanie z innymi metaheurystykami



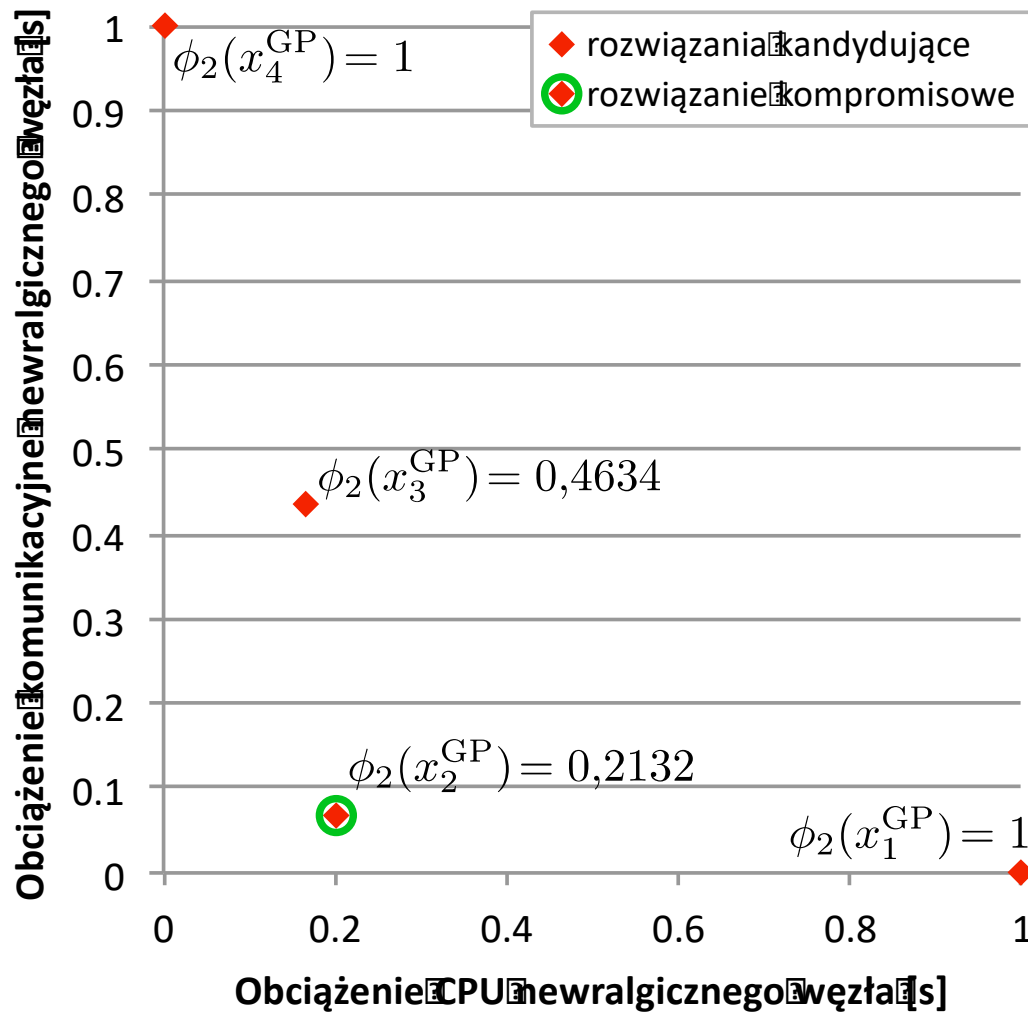
Oceny rozwiązań końcowych wyznaczone przez cztery metaheurystyki

Oceny *Pareto*-optymalne wyznaczone za pomocą metaheurystyk

Obszar	Oceny rozwiązań $F(x) = [\hat{Z}_{\max}(x), \tilde{Z}_{\max}(x)]^T$			
	MOGPA	NSGA-II	OMOPSO	HS
I	$x_1^{\text{GP}}: (581; 25221)$	(587, 25221)	–	(587, 25221)
II	$x_2^{\text{GP}}: (448; 25952)$	(442; 25952)	–	(448; 25952)
III	$x_3^{\text{GP}}: (442; 29973)$	–	(442; 29973)	(435; 29973)
IV	$x_4^{\text{GP}}: (414; 36187)$	–	–	(411; 36187)

Źródło: opracowanie własne

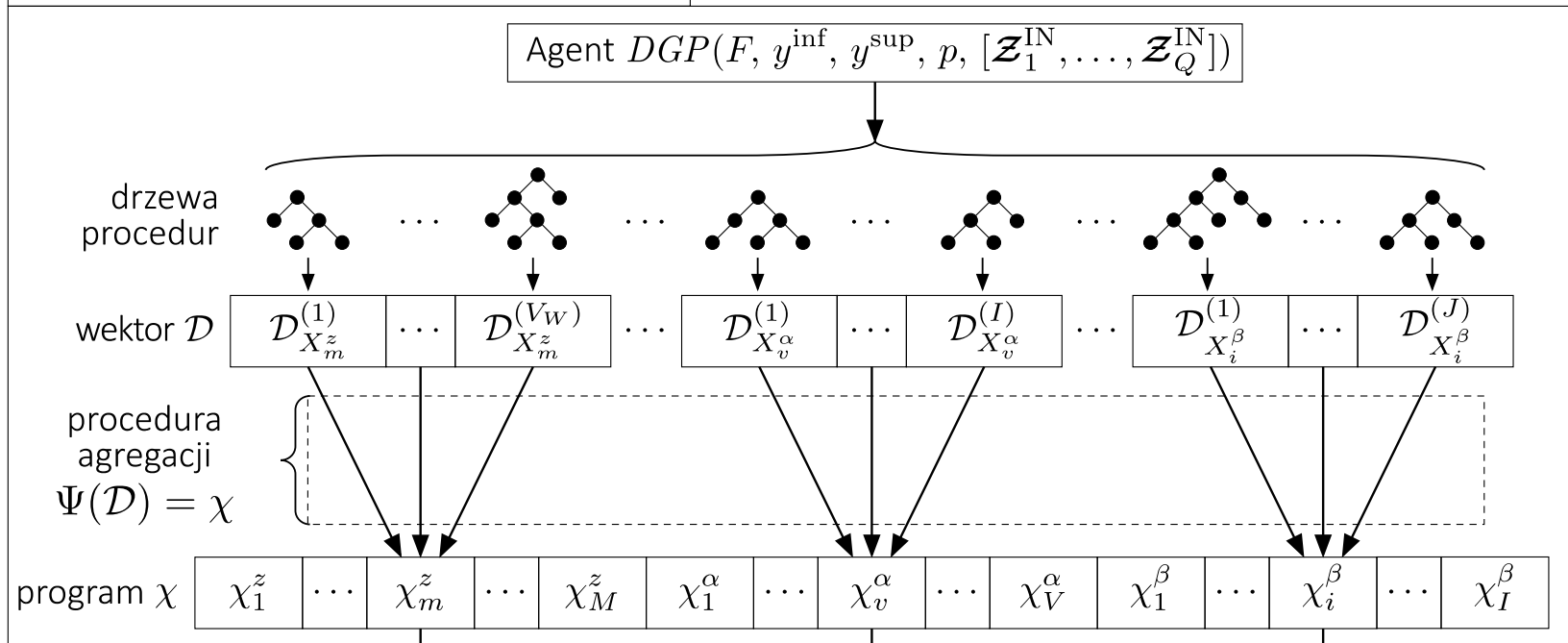
Wybór strategii kompromisowej dla zagadnienia rozbudowy gridu *Comcute*



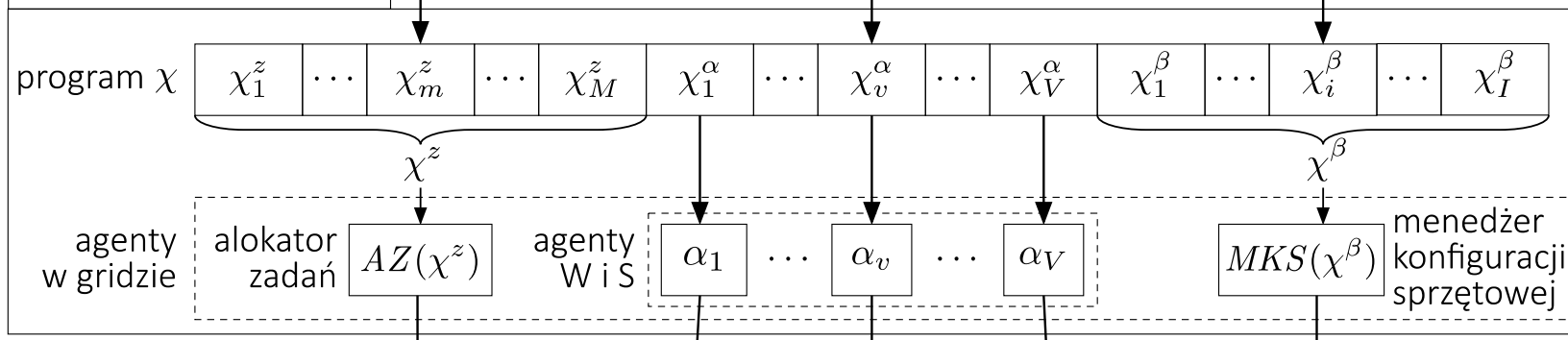
Oceny rozwiązań wyznaczonych przez algorytm MOGPA
w znormalizowanej przestrzeni kryterialnej

Wdrożenie rozwiązania w gridzie Comcute

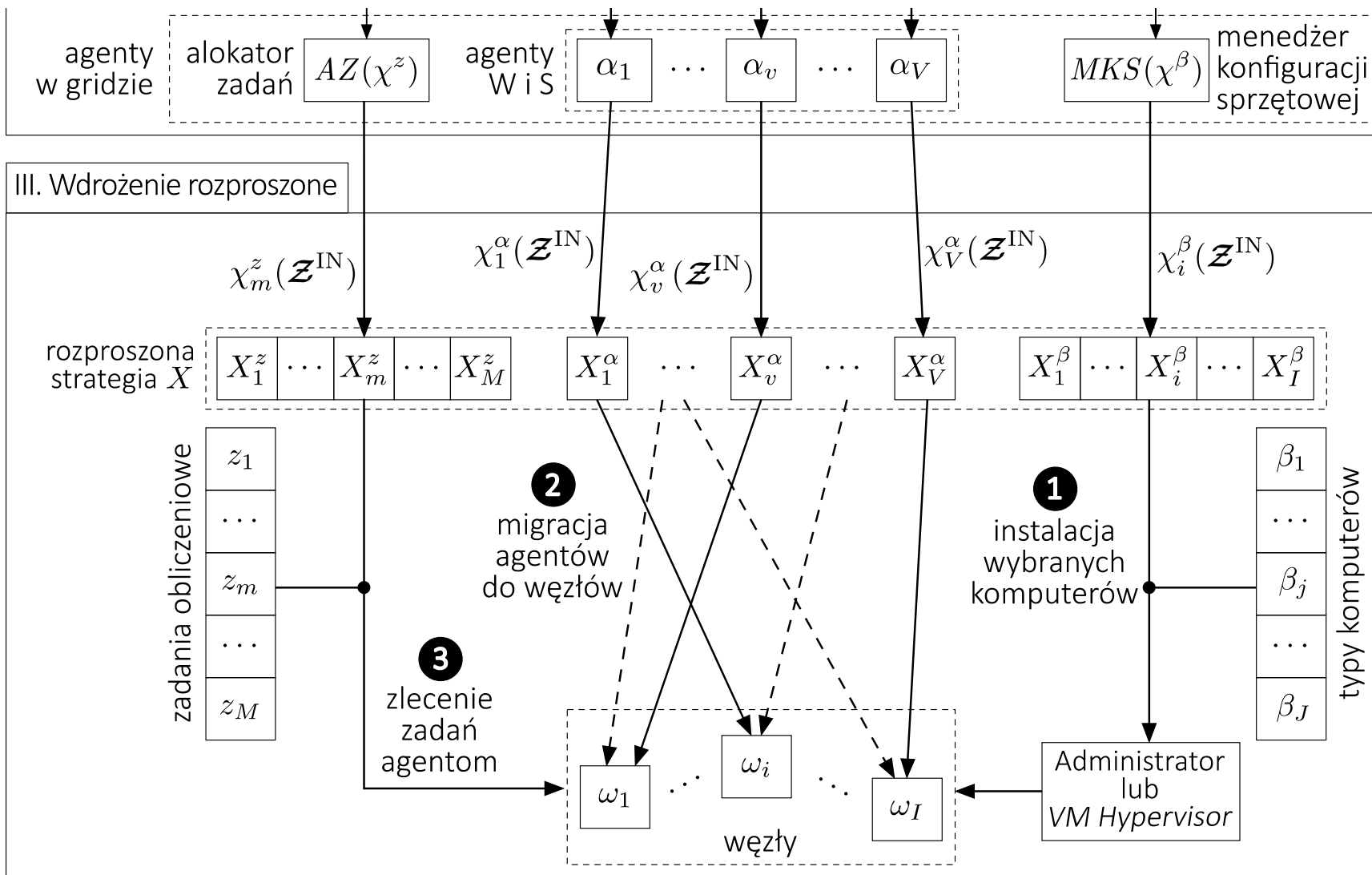
I. Wybór programu do wyznaczania strategii



II. Propagacja programu



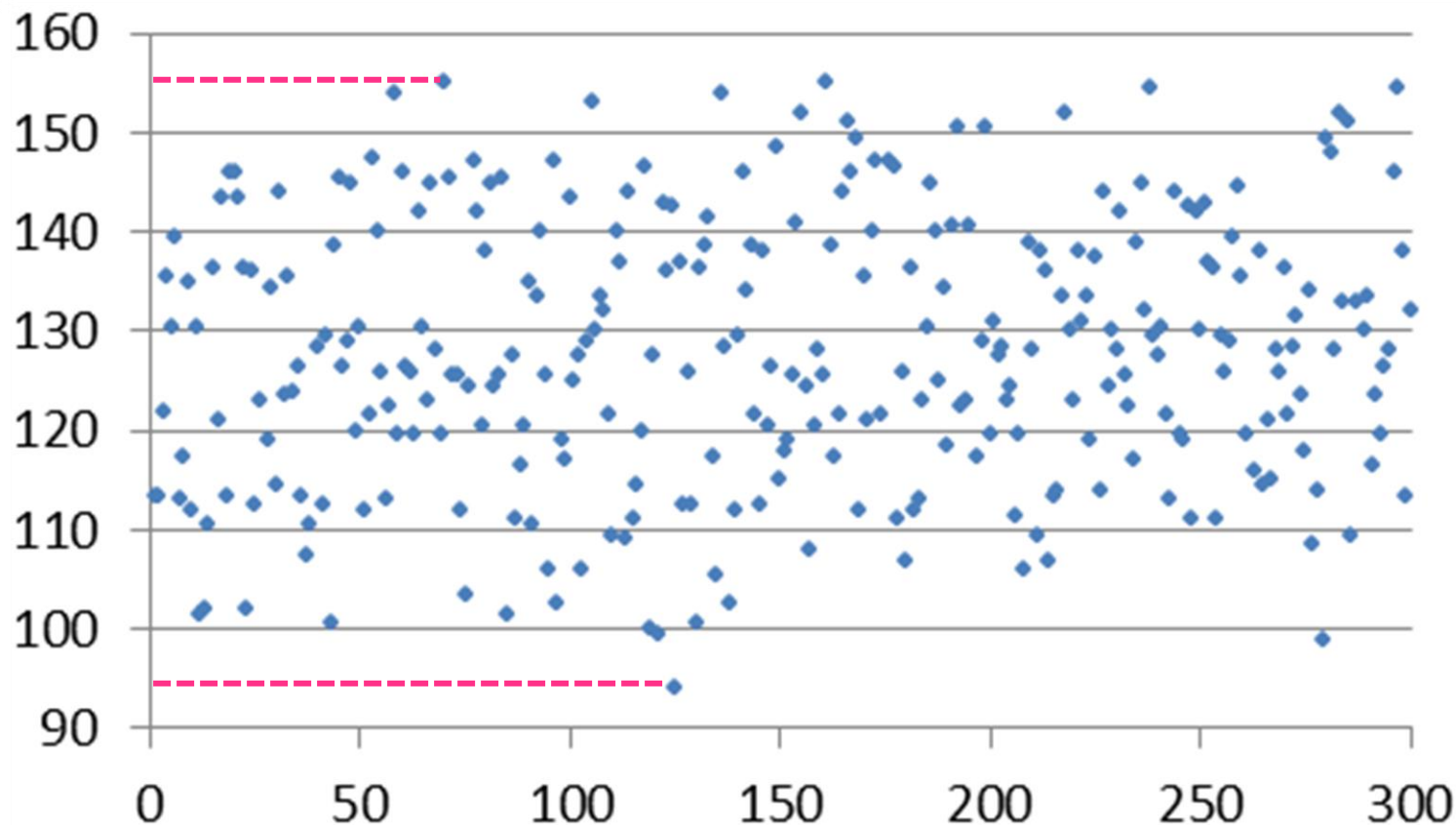
Wdrożenie rozwiązania w gridzie Comcute



Wrażliwość ważonego obciążenia Δ_{max}

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

Δ_{max} [s]



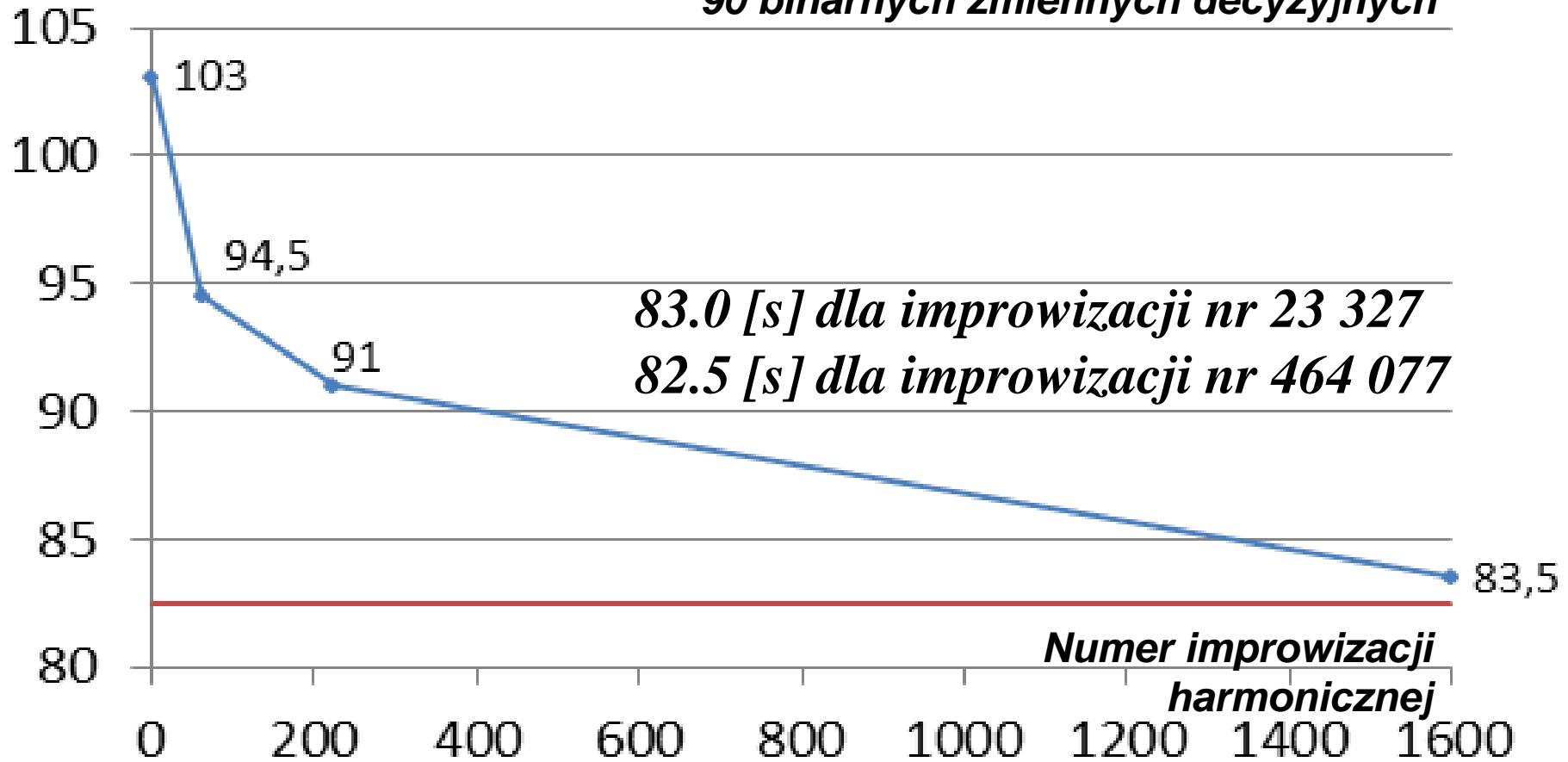
Numer losowo wygenerowanej konfiguracji dopuszczalnej

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

Δ_{max} [s]

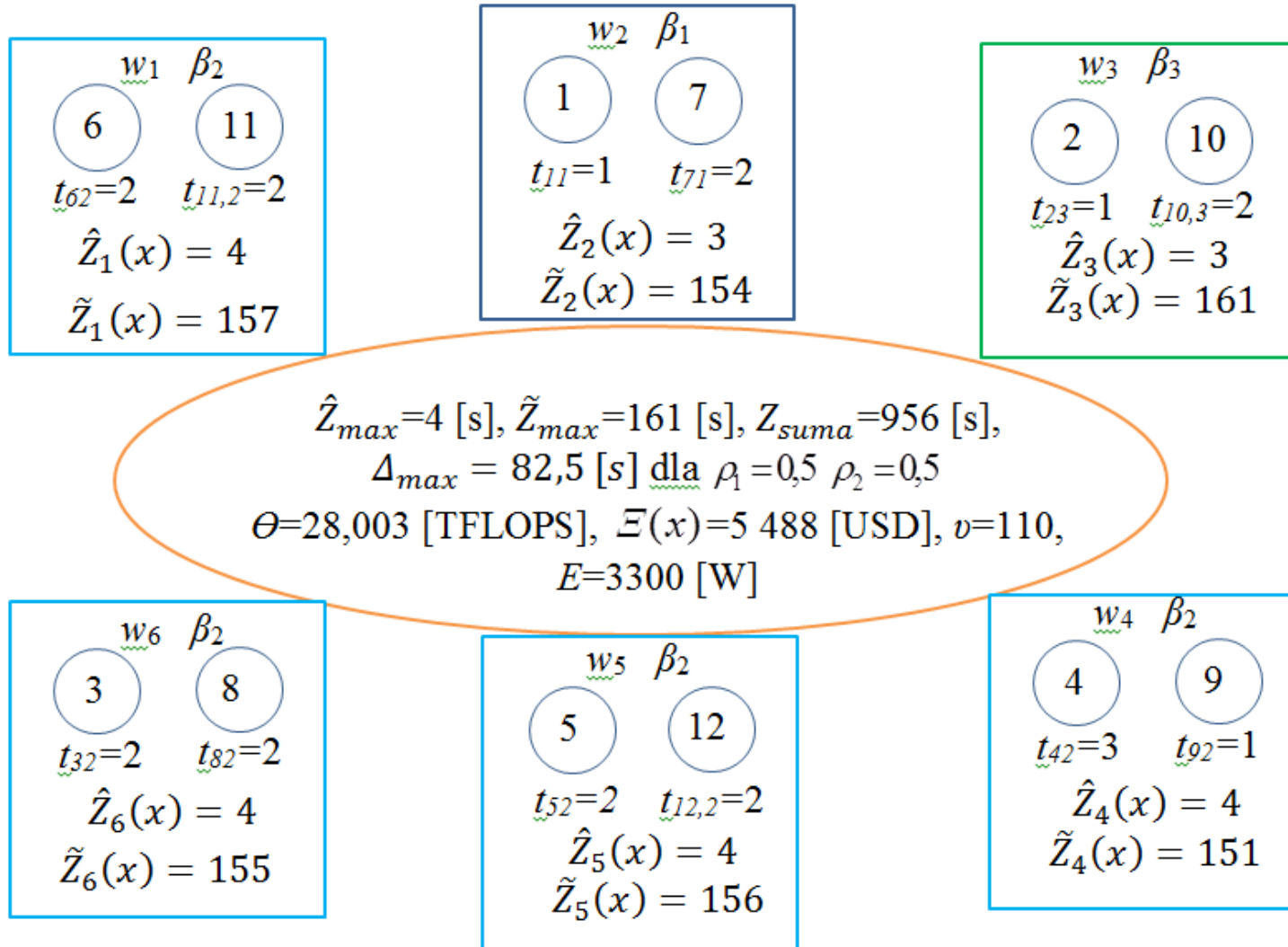
$V=12, l=6, J=3$

90 binarnych zmiennych decyzyjnych



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

Diagram suboptymalnej konfiguracji gridu pod względem ważonego obciążenia $\Delta_{max} = 82,5$ [s], przy czym $V=12, l=6, J=3$



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^{\wedge}, Z^{\sim} max

display temporary HM2

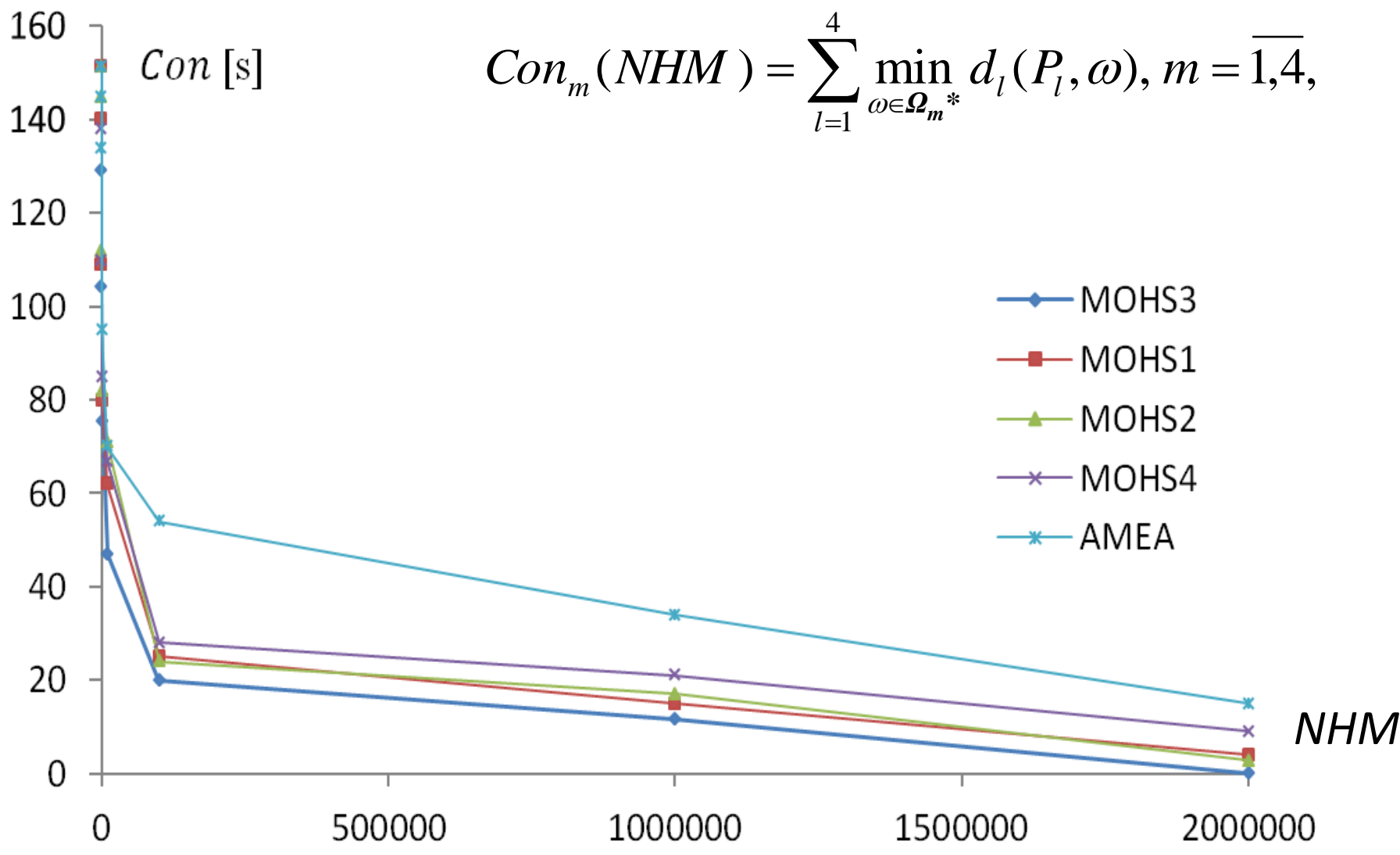
display concat. HM+HM2

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 $^{\wedge}$	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

Porównanie wielokryterialnych algorytmów harmonicznych MOHS i ewolucyjnych

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



Wnioski i uwagi

1. **Problemy optymalizacji strategii sieci agentów opierają się na siedemnastu kryteriach wybieranych przez interesariuszy.**
2. **Optymalizacja może dotyczyć minimalizacji obciążenia procesorów w newralgicznym komputerze, minimalizacji obciążenia komunikacyjnego w newralgicznym hoście, maksymalizacji wydajności gridu, minimalizacji kosztów zakupu komputerów bądź kosztów realizacji zadań, maksymalizacji stopnia rozproszenia agentów czy też maksymalizacji wielkości dostępnej pamięci RAM w newralgicznym komputerze.**

Wnioski i uwagi

- 3. Równoważenie obciążeń może odnosić się również do innych zasobów, takich jak pamięć dyskowa HDD czy pamięć półprzewodnikowa SSD. Brane pod uwagę są wymagania związane z maksymalizacją poziomu dostępności gridu czy minimalizacją zużycia energii elektrycznej. Można także rozważać minimalizację łącznego obciążenia procesorów w węzłach gridu, łącznego obciążenia komunikacyjnego lub łącznego czasu realizacji zadań. Na wartości każdego z kryteriów optymalizacji mogą być nakładane ograniczenia.**
- 4. Do wyznaczania rozwiązań dla problemów optymalizacji strategii sieci agentów opracowano cztery grupy algorytmów. Pierwszą grupę stanowią algorytmy do optymalizacji jednokryterialnej, w której interesariusze wybierają funkcję celu oraz zestaw ograniczeń. Druga grupa obejmuje algorytmy do wyznaczania rozwiązań Pareto-optymalnych. Trzecia grupa to algorytmy poszukujące strategii kompromisowych. Do czwartej grupy należą algorytmy programowania genetycznego do implementacji programów wyznaczających strategię zespołu agentów w dynamicznym środowisku w zależności od stanu systemu.**

Wnioski i uwagi

5. Algorytmy zaimplementowano w języku Java w ramach pakietu AAG'16, który pozwala na powtórzenie wykonanych eksperymentów i uzyskanych dla nich wyników badań. Opracowane metody mogą być zastosowane do wyznaczania strategii, które odzwierciedlają preferencje interesariusza odnośnie kryteriów i ograniczeń w sformułowanych zagadnieniach optymalizacji wielokryterialnej.
6. Wykonano szereg eksperymentów dla testowych instancji sformułowanych zagadnień optymalizacji z wykorzystaniem opracowanych algorytmów. Otrzymane rezultaty potwierdziły prawidłowość rozważań dotyczących modelu systemu, sformułowanych zadań optymalizacji i skonstruowanych algorytmów programowania genetycznego.
7. Opracowane metody wykorzystano do rozwiązania problemów optymalizacji w odniesieniu do laboratoryjnej instancji gridu Comcute, a także w odniesieniu do chmury obliczeniowej. Świadczy to o uniwersalności rozważanego modelu i możliwościach jego adaptacji do różnorodnych zagadnień optymalizacji.

Kierunki dalszych badań

- 1. Interesujące kierunki dalszych badań obejmują rozwój metod programowania genetycznego w celu rozwiązywania innych zagadnień polioptymalizacji.**
- 2. Ciekawym obszarem zastosowań jest także tzw. przetwarzanie we mgle (ang. *fog computing*) w miejskich infrastrukturach wspierających inteligentne procesy obliczeniowe. Umożliwia ono realizację obliczeń, które wspomagają funkcjonowanie inteligentnego miasta czy regionu.**

Najważniejsze publikacje

1. Balicki, J., Korłub, W., Paluszak, J., Tyszka, M.: Harmony Search for Self-configuration of Fault-Tolerant and Intelligent Grids. In IFIP International Conference on Computer Information Systems and Industrial Management (pp. 566-576). Springer International Publishing, 2016.
2. Balicki J., Beringer M., Korłub W., Przybyłek P., Tyszka M., Zadroga M.: Collective citizens' behavior modelling with support of the Internet of Things and Big Data, W: 8th International Conference on Human System Interactions (HSI), 2015.
3. Balicki J., Korłub W., Szymański J., Zakidalski M.: Big Data Paradigm Developed in Volunteer Grid System with Genetic Programming Scheduler, In: Artificial Intelligence and Soft Computing. Part I, 2014, Springer-Verlag, Berlin.
4. Balicki J., Korłub W., Krawczyk H., Paluszak J.: Genetic Programming with Negative Selection for Volunteer Computing System Optimization, W: Proceedings of the 6th International Conference on Human System Interaction, 2013, IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA.

Dziękuję za uwagę

