

Programowanie Genetyczne

dr Dariusz Pałka
dpalka@agh.edu.pl

Programowanie genetyczne

Q: „**What is Genetic Programming?**”

A: „Genetic programming (GP) is an automated method for creating a working computer program from a high-level problem statement of a problem. Genetic programming starts from a high-level statement of “what needs to be done” and automatically creates a computer program to solve the problem.”



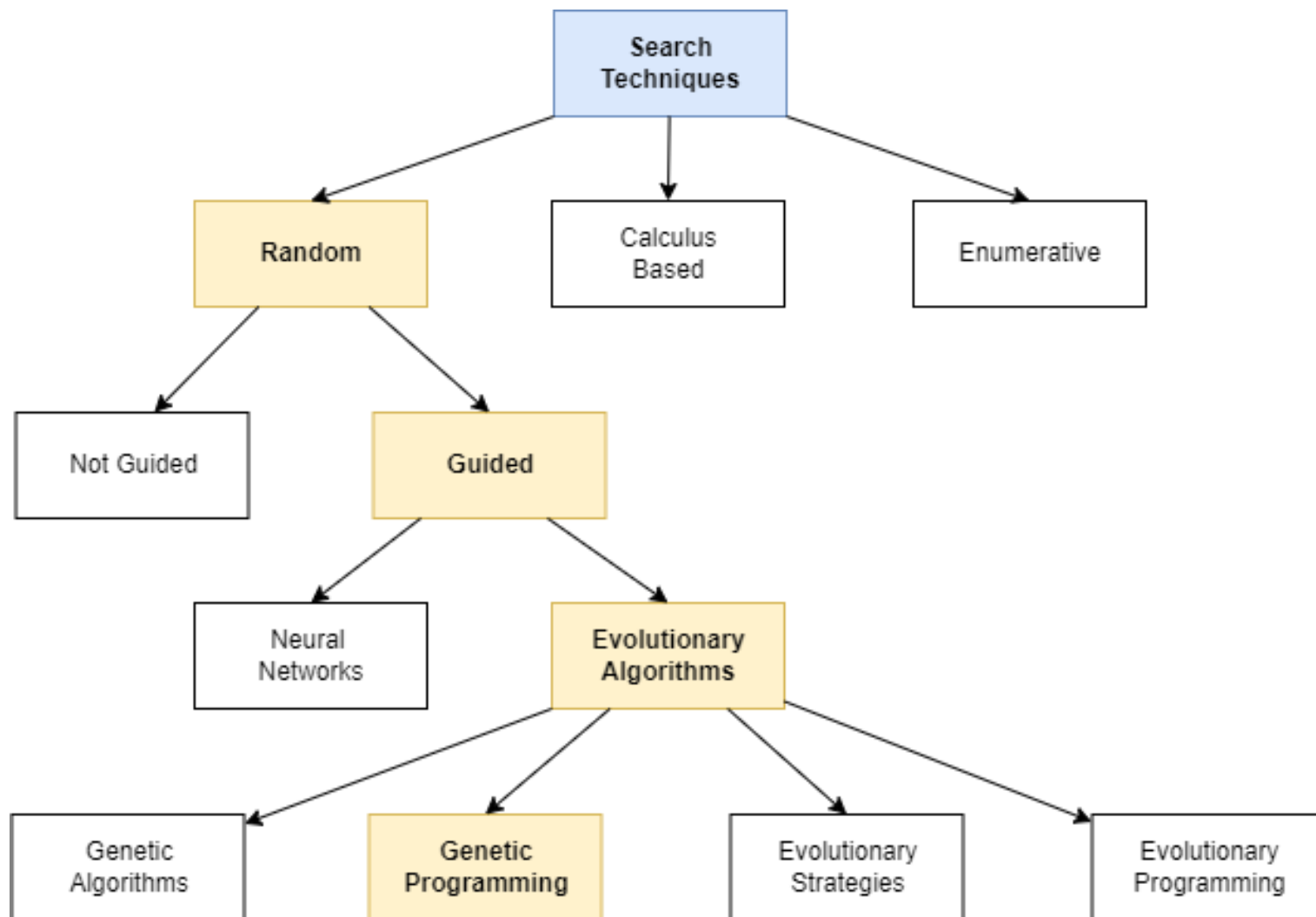
GP

- Kierowane wyszukiwanie losowe (guided random search)
- Wykorzystujące populację osobników (rozwiązań)
- Metoda inspirowana ewolucją naturalną
 - Osobniki lepiej przystosowane do środowiska mają większe szanse na reprodukcję i przekazanie swoich genów do następnego pokolenia
 - Funkcja dopasowania – ocena osobników
 - Operatory pozwalające na zmianę rozwiązania

Kiedy GP jest szczególnie użyteczne

- Olbrzymia przestrzeń poszukiwań (search space)
- Dokładna postać rozwiązania nie jest znana
- Można zaakceptować rozwiązanie przybliżone

Techniki wyszukiwania



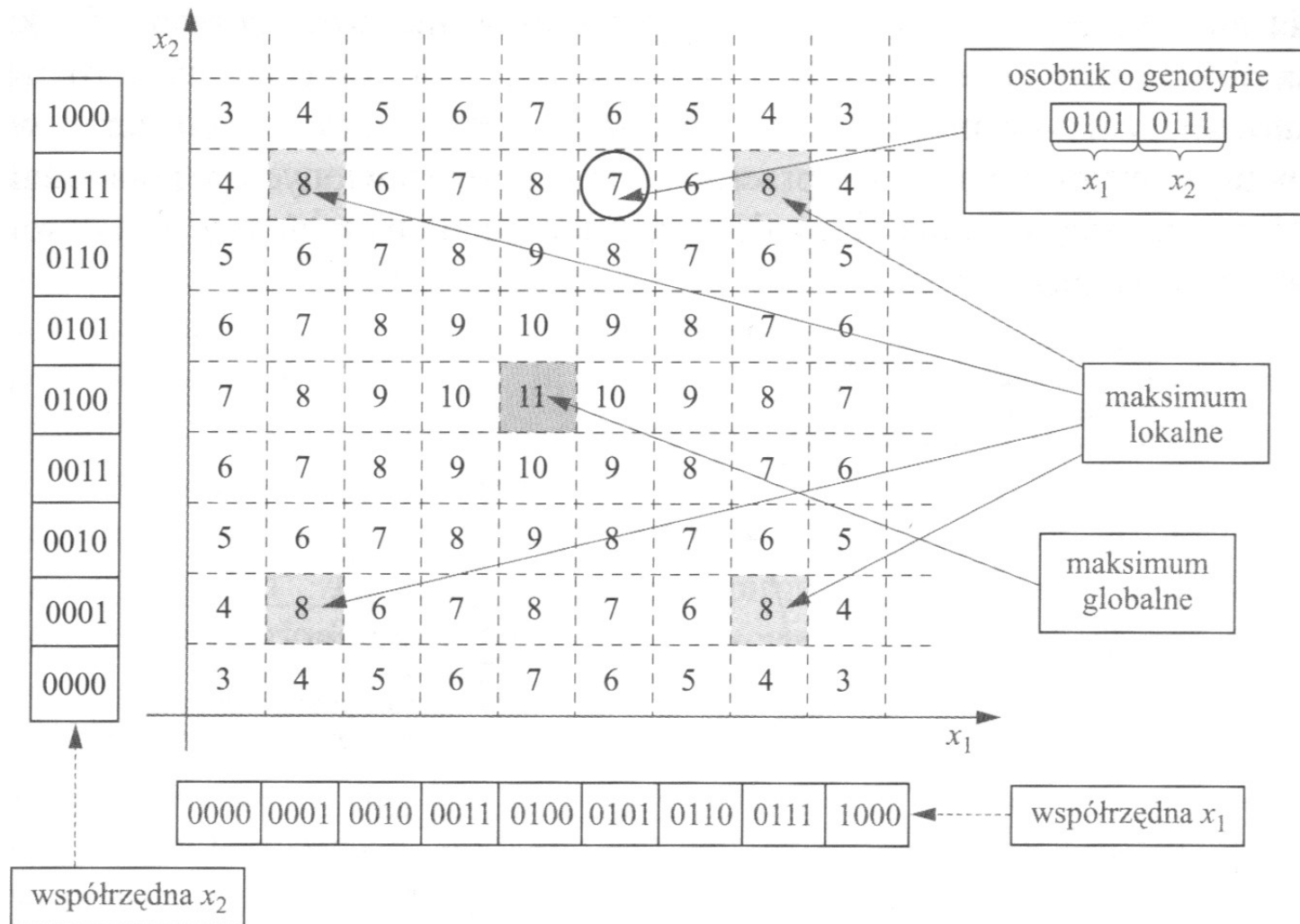
Geneza GP

- Pierwsze znane(?) koncepcje ewolucji programów pochodzą od Alana Turinga „Computing Machinery and Intelligence” (1950)
- Algorytmy genetyczne
 - prace Alexa Fräsera (1957)
 - Upowszechnienie GA przez Johna Hollada 1975 „Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence”
- 1966 praca Fogel, Owens i Walsh – idea programowania ewolucyjnego w którym do rozwiązania postawionego problemu używane są CA podlegające ewolucji

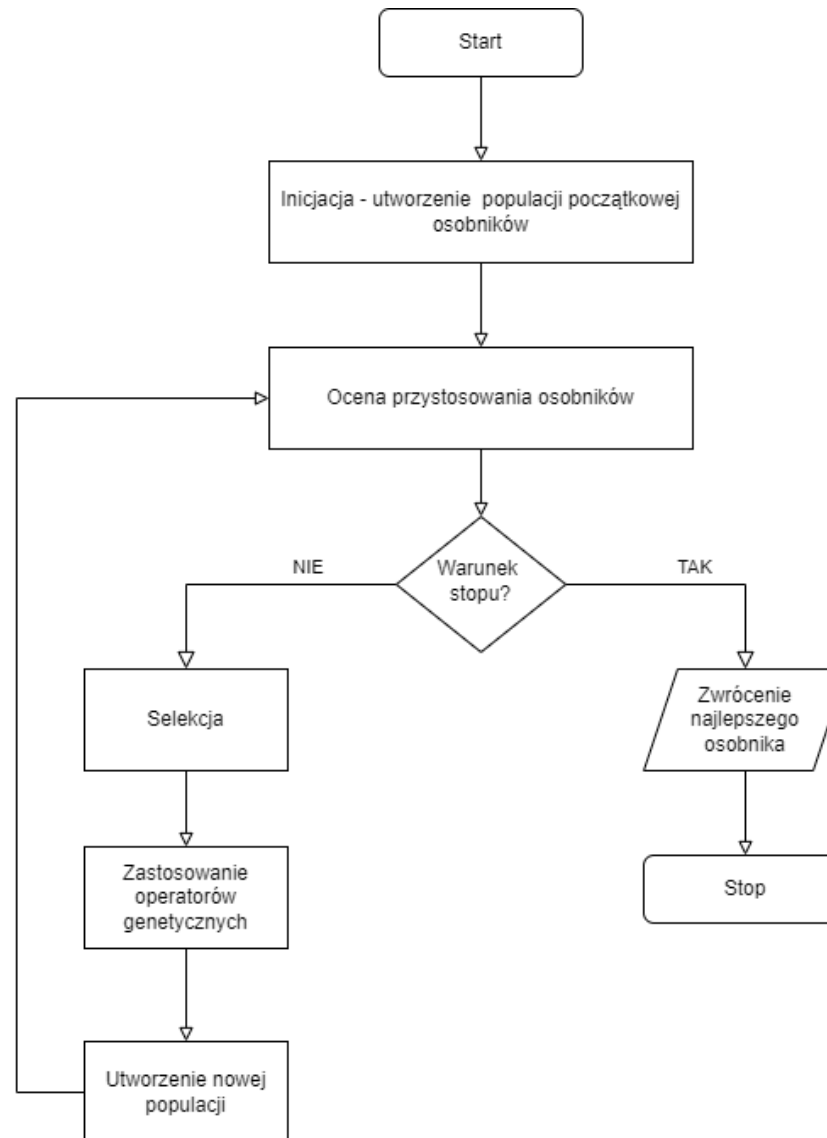
Geneza GP

- 1985 Cramer jako pierwszy zastosował operację krzyżowania poddrzew dla potrzeb ewolucji w języku TB (tree based)
- Programowanie genetyczne - John Koza (doktorant Hollanda) prace od 1988
 - 1990 „Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems” <http://www.genetic-programming.com/jkpdf/tr1314.pdf>
 - 1992 „Genetic Programming: On the Programming of Computers by Means of Natural Selection”, MIT Press
 - 1994 Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press

GA



GA - schemat



Algorytmy ewolucyjne terminologia

- **Populacja** – zbiór osobników o określonej liczebności
- **Osobniki** – zakodowane w postaci chromosomu zbiory parametrów rozwiązania (punkty w przestrzeni poszukiwań)
- **Gen** (cecha, znak) – pojedynczy element genotypu
- **Genotyp** – zespół (struktura) chromosomów danego osobnika. Osobnikami populacji mogą być genotypy lub pojedyncze chromosomy (jeśli genotyp składa się z jednego chromosomu)
- **Fenotyp** – zestaw wartości odpowiadający danemu genotypowi (zdekodowana struktura), zbiór parametrów rozwiązania
- **Allel** – wartość danego genu (wartość cechy)

GP

- Nie nakłada ograniczeń na formę rozwiązania – struktura rozwiązania jest dowolna
- Rozwiązania mogą reprezentować cokolwiek co da się „zakodować” - np.
 - Programy (w dowolnych językach)
 - Funkcje, formuły, równania
 - Układy elektroniczne
 - Projekty

GP

- „+” - Szeroki zakres zastosowań
- „-” - Rozwiązania mogą szybko się rozrastać i być niepotrzebnie złożone

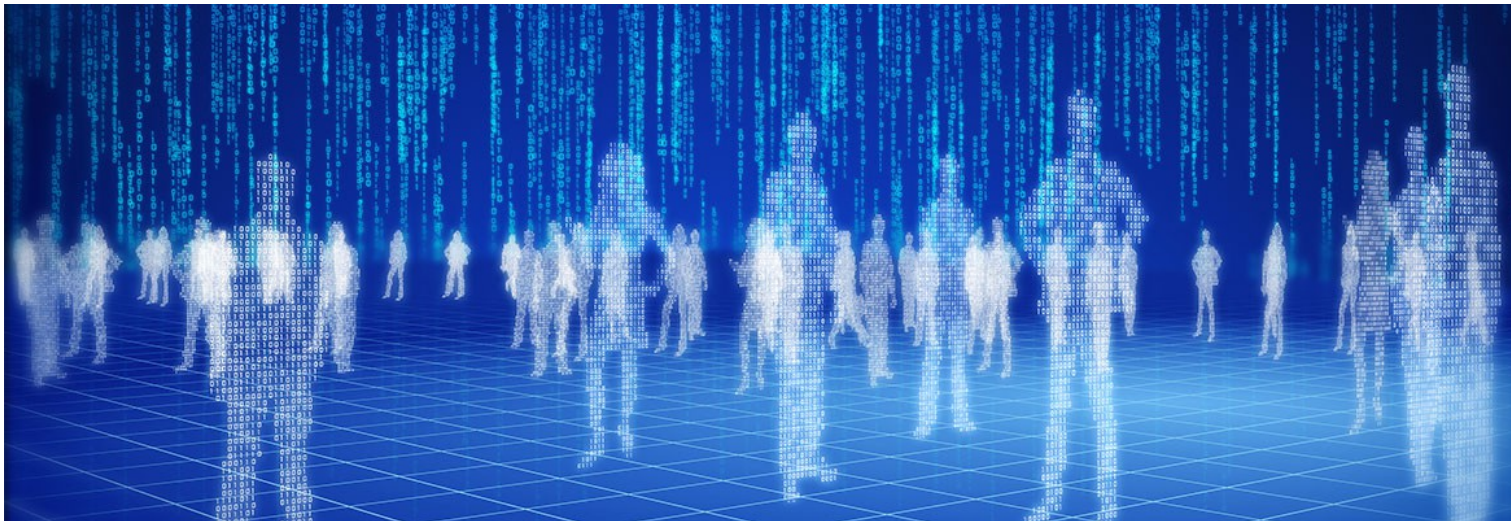
GP Przykłady

- Projekt
 - 2006 NASA Space Technology 5 aircraft antenna (<https://www.jpl.nasa.gov/nmp/st5/TECHNOLOGY/antenna.html>)



Wykorzystanie populacji

- Możliwość zastosowania obliczeń równoległych
- Możliwość przeszukiwania wielowymiarowych przestrzeni rozwiązań



Źródło: <https://continuingstudies.uvic.ca/>

GP - podstawy

- Rozszerzenie klasycznych algorytmów genetycznych
- Kodowanie binarne zastąpione (na ogół) kodowaniem drzewiastym (LISP)
- Zmiana sposobu działania operacji genetycznych (adaptacja do reprezentacji drzewiastej)

Lisp przykład

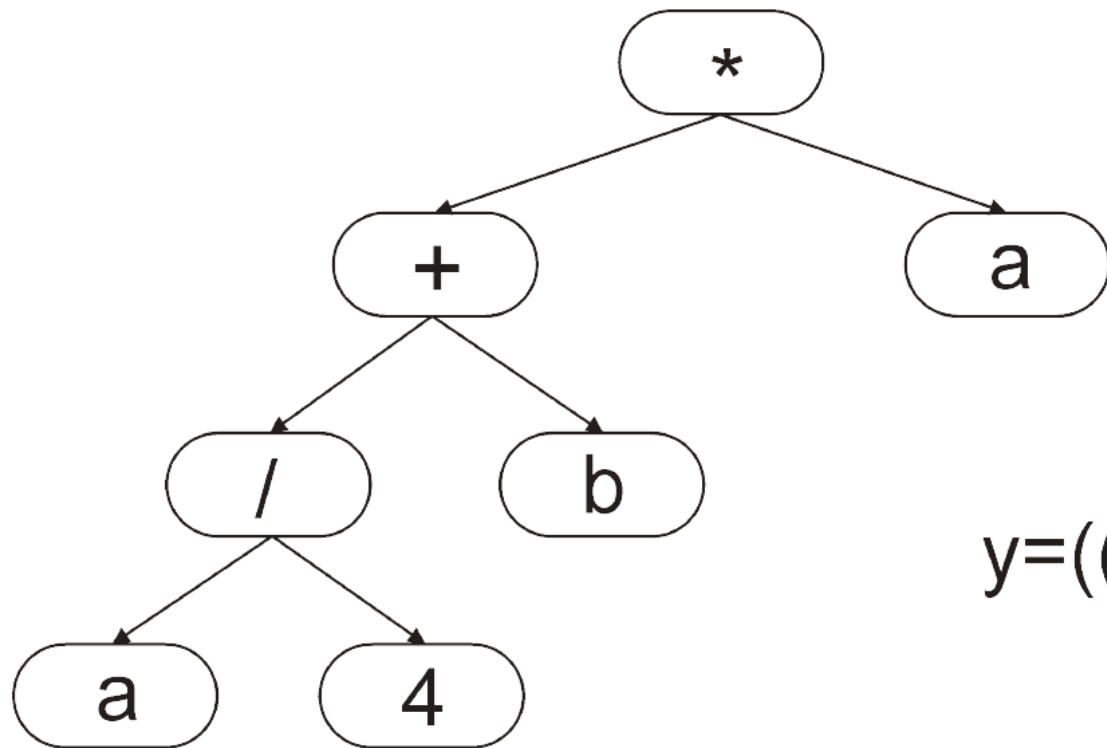
```
(defun factorial (n)
  (if (zerop n) 1
      (* n (factorial (1- n)))))
```

```
(defun factorial (n)
  (loop for i from 1 to n
        for fac = 1 then (* fac i)
        finally (return fac)))
```


GP - reprezentacja

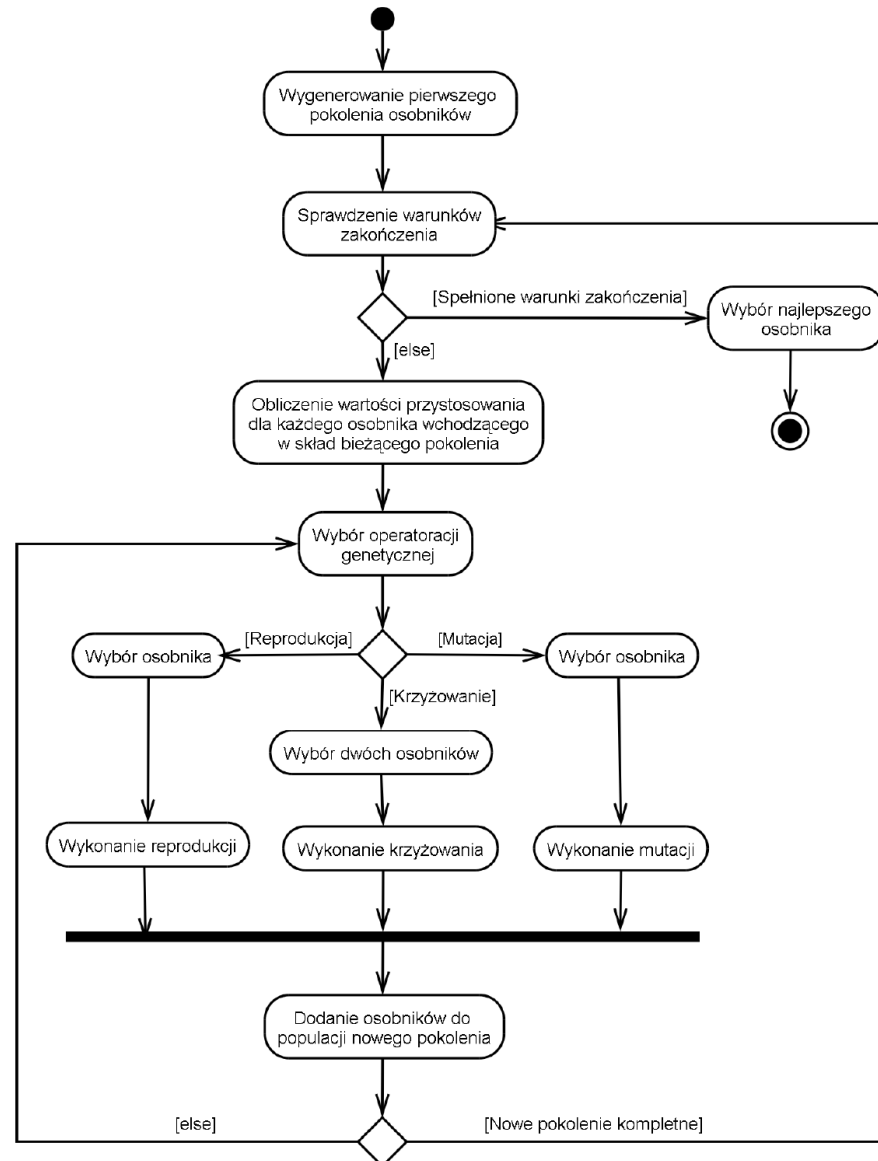
- Program reprezentowany jest za pomocą drzewa, które składa się z jednego lub większej ilości wierzchołków
- Wierzchołki pochodzą z 2-ch rozłącznych zbiorów:
 - Zbioru funkcji F
 - Zbioru terminali T
- Wyznaczenie zbiorów F i T to etap wstępny użycia GP

GP - reprezentacja



$$y = ((a/4) + b) * a$$

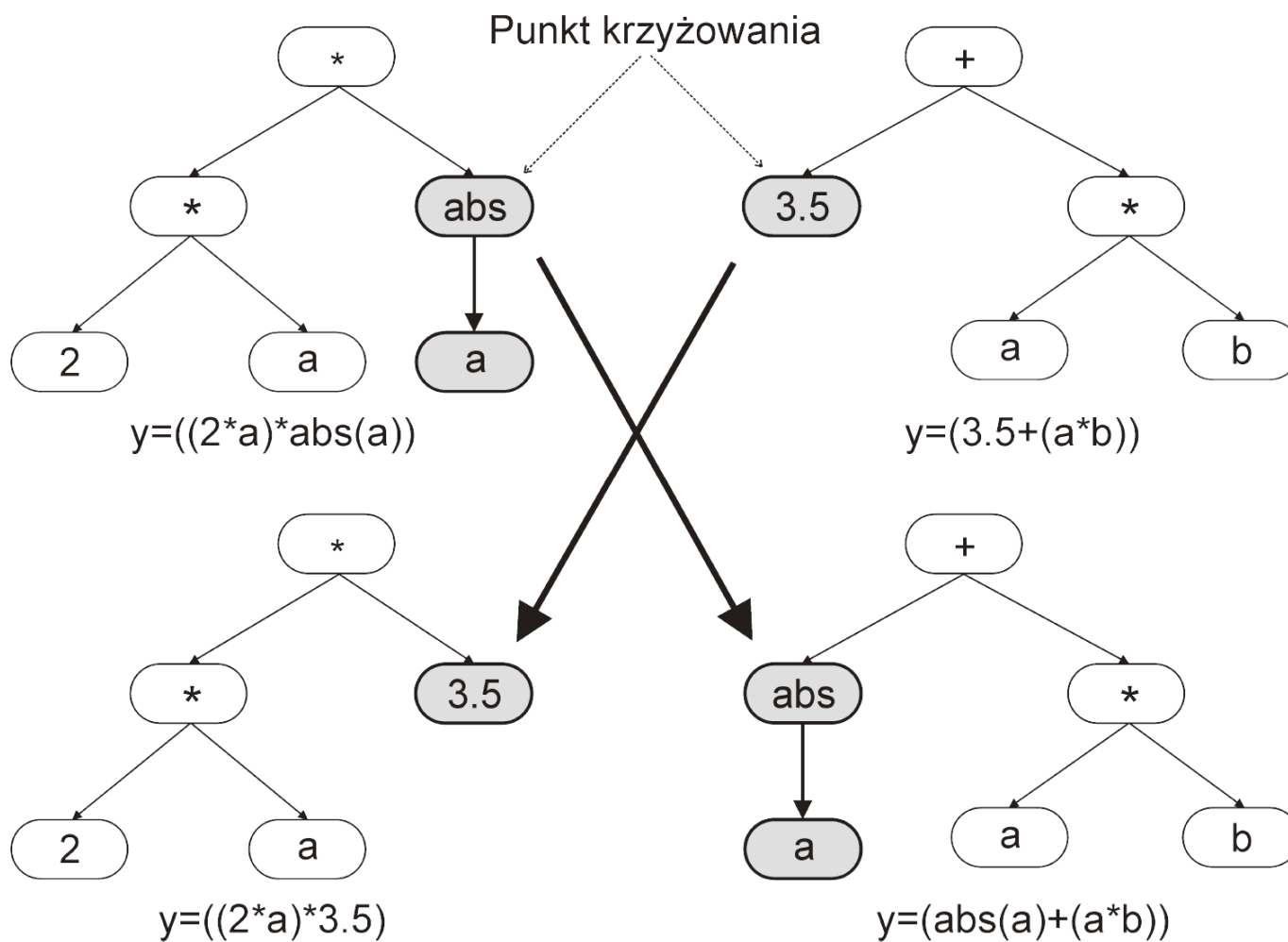
GP - schemat



Etapy GP

- Generowanie pierwszego pokolenia
 - Metoda „full”
 - Metoda „grow”
 - Metoda „ramped half and half”
- Ocena osobników
- Selekcja osobników
- Operacje genetyczne...

GP crossover



GP przykłady

- Regresja symboliczna
- ...
- GP in Deep Learning
- Annual "Humies" Awards For Human-Competitive Results (<https://www.human-competitive.org/>)

GP Regresja symboliczna

- Funkcja $f(x) = x_1 * x_1 + 2$
- (21 pokolenie) Best Individual :
$$\begin{aligned} & (((((X_1 * X_1) + -3.0233394996582597) - \\ & -4.801968845897364) - (-0.8545071961179138 / ((2.0442178544747467 + \\ & 1.8153729320993577) + (0.030506589503922044 / ((((-2.9385330367581917 - \\ & (2.0049232666401293 - ((0.01134529576916421 - 1.4868206109448732) + (- \\ & 0.841732337286504 * 2.0442178544747467)))) / ((((-0.7646475814000189 * ((- \\ & 0.6671452250416889 - -2.0007740520907316) * (0.6446568470410696 + \\ & (3.2917380291428504 + 1.3213029289151779)))) - 0.01134529576916421) / (((- \\ & 0.4997098444746708 - -2.0007740520907316) * (((3.383413944946442 + (- \\ & 0.4997098444746708 * (2.003699960875358 - -2.7308195191268503))) - (- \\ & 3.4126448179984092 + -0.21299446463918148)) / 2.0442178544747467) * (((- \\ & 2.8595363135507146 * 4.199601223568122) / ((-2.9385330367581917 - \\ & (2.0049232666401293 - 2.0049232666401293)) + (-0.841732337286504 * \\ & 1.4868206109448732))) - 1.4868206109448732))) + (0.9419984161536101 - \\ & (3.835093090071073 / 0.01134529576916421)))) - -0.21299446463918148)) - \\ & ((2.0421496256592704 + -3.0638365159785175) / ((1.3213029289151779 / \\ & 0.01134529576916421) + -3.546239445571934))) * (1.21911428092318 / \\ & 2.0049232666401293)) * -2.9176395411614733)))) \end{aligned}$$

Czy GP ma sens?

- Analogie do metodyk zwinnych
 - W TDD kryterium poprawności rozwiązania są testy unitarne (automatyczne)
 - W metodykach zwinnych np. w XP to czy dana funkcjonalność została poprawnie zaimplementowana jest oceniane przez testy akceptacyjne (automatyczne)

Czy GP ma sens?

- Niezrozumiały (wygenerowany) kod?
 - Co z funkcjami systemowymi i wstawkami np. w assemblerze?
 - A co np. z parserami LR (tablice Action i GOTO)?
 - Wprowadzanie zmian w nieznanym języku programowania
 - Co z programowaniem Maszyny Turinga albo językami typu brainf**k
(<https://en.wikipedia.org/wiki/Brainfuck>)?

Zasoby

- John Koza Genetic Programming - The Movie - Part 1 <https://www.youtube.com/watch?v=tTMpKrKkYXo>
- Regresja symboliczna 2:47, 9:47
- John Koza Genetic Programming - The Movie - Part 2 <https://www.youtube.com/watch?v=pRk6cth7Bpg>