



# Wprowadzenie do sztucznej inteligencji: Algorytmy genetyczne. Podstawy i charakterystyka algorytmów genetycznych. Metody selekcji, krzyżowanie, mutacja

dr hab. inż. Piotr Szczuko, prof. PG  
Katedra Systemów Multimedialnych



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego  
Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

# Algorytm genetyczny

*Algorytmy genetyczne* wykorzystywane do **optymalizacji**: poszukiwanie najlepszego lub zbliżonego do najbliższego rozwiązania spośród wszystkich możliwych rozwiązań danego problemu

Weryfikacja **wszystkich rozwiązań** byłaby **czasochłonna**

Poszukiwane w AG przyspieszane jest poprzez zastosowane operacji naśladujących **ewolucję i dobór naturalny**

# Problemy klasycznej optymalizacji

Co jest optymalne: konfiguracja sprzętu, rozwiązania techniczne, kompromisy między wagą a pojemnością baterii, liczbą manipulatorów a wagą (robotyka), kolejność etapów produkcji, itd.

Klasyczna metoda optymalizacji:

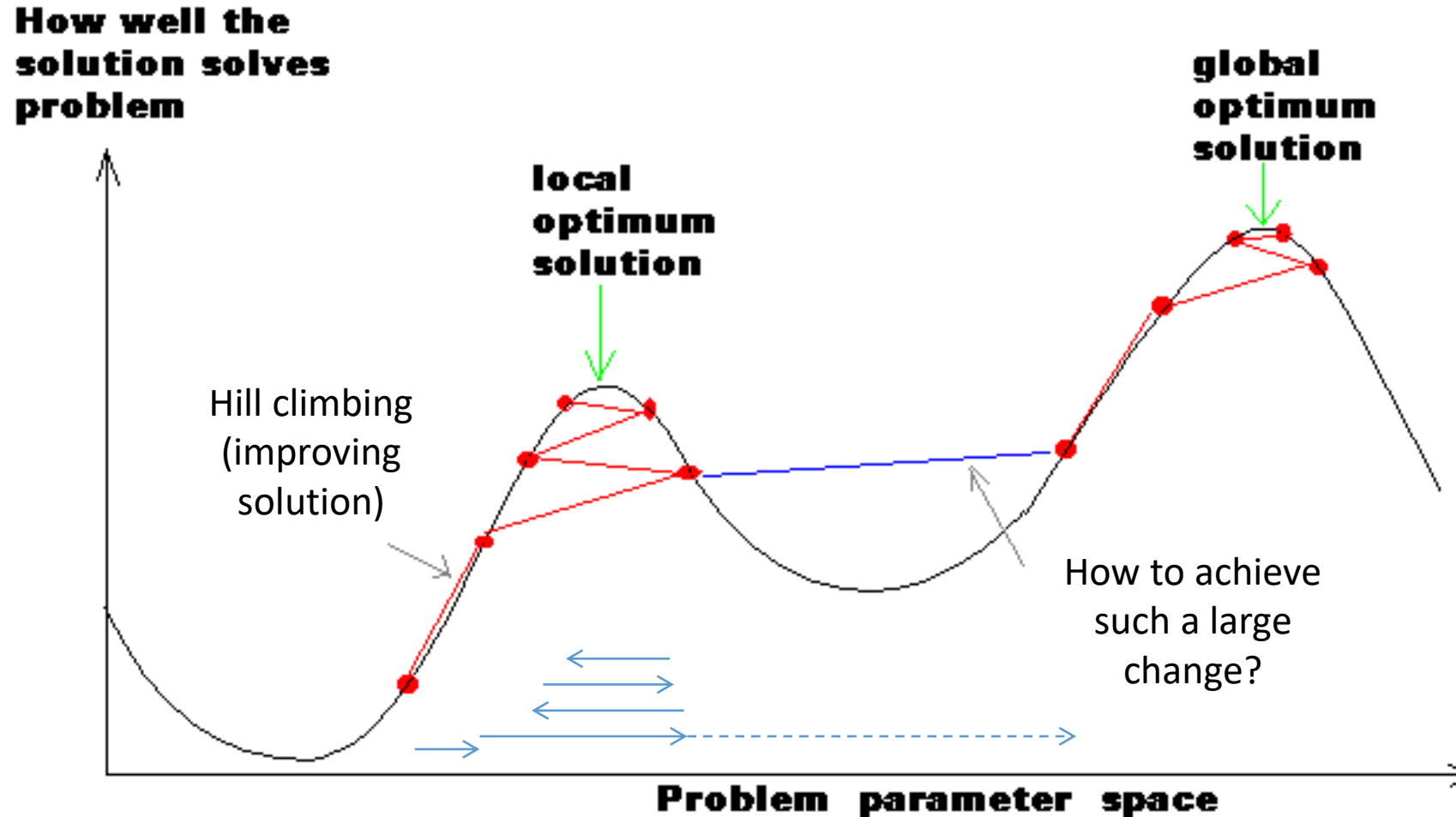
- Utworzenie funkcji celu  $n$  zmiennych:  $f(x_1, x_2, \dots, x_n)$
- Analiza funkcji: minima, maksyma, wartość optymalizująca wynik
  - Jeśli funkcję da się wyrazić algebraicznie – poszukiwanie zer pierwszej pochodnej.
  - Jeśli funkcji nie da się wyrazić – próbkowanie w przestrzeni zmiennych, określanie dla nich wyniku  $f(x)$ , np. poprzez symulację pojedynczego przypadku, praktyczną realizację, itd.

# Problemy klasycznej optymalizacji

Trudne do uniknięcia problemy:

- Zagadnienia są złożone, trudne do opisanie funkcją.
- Próbkowanie, testowanie, duża liczba wymaganych działań (np. symulacji)
- Rozwiązanie może być zaledwie lokalnym optimum (pozornie nie do poprawienia)

# Pułapka lokalnego optimum



# Algorytmy ewolucyjne

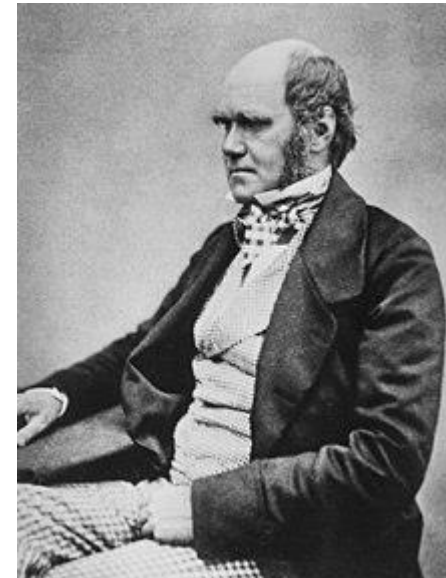
- Alternatywny sposób optymalizacji
- Szybszy, potencjalnie bez zatrzymania w lokalnych optimach
- **UWAGA:** rozwiązanie znalezione to **przypuszczalnie** najlepsze rozwiązanie (nie ma pewności)

# Przykład

- [http://rednuht.org/genetic\\_walkers/](http://rednuht.org/genetic_walkers/)
- Wartość nagrody za pozostawanie w pozycji wyprostowanej i wykonane kroki

# Karol Darwin – selekcja naturalna

- Karol Darwin, 1854, „O pochodzeniu gatunków”:
- „...*advancement of all organic beings, namely, multiply, vary, let the strongest live and the weakest die.*”
- Ulepszanie wszystkich istot poprzez: rozmnażanie, różnicowanie, niech przetrwa najsilniejszy...

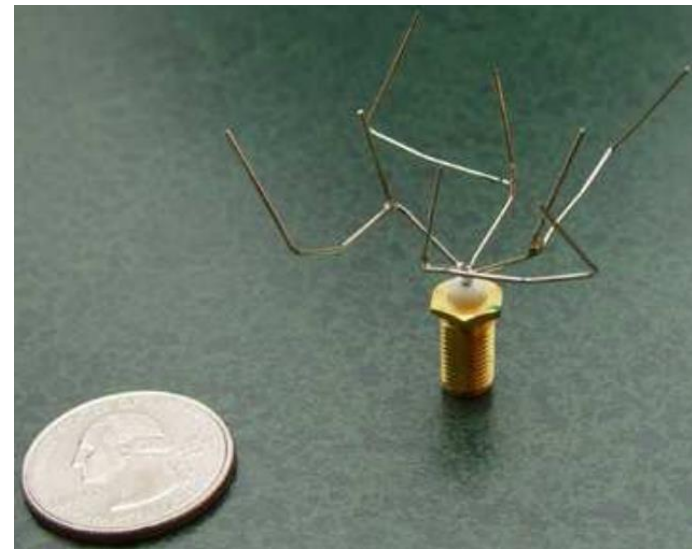




# Ewolucja „anteny”

- 2006 NASA Space Technology 5 mission
- Połączenie cech: szerokiej wiązki promieniowania dla kołowo spolaryzowanej fali i dużego zakresu impedancji

**Źródło:** Hornby, Gregory S.; Al Globus; Derek S. Linden; Jason D. Lohn (2006). "*Automated antenna design with evolutionary algorithms*". American Institute of Aeronautics and Astronautics.



# Założenia algorytmu genetycznego

Każde rozwiązanie (lepsze, gorsze, nie chodzi o najlepsze, tylko o testowane hipotetyczne sposoby realizacji celu) traktowane są jako osobniki populacji

Algorytm symuluje procesy:

- **Ocena** stopnia dopasowania osobnika do zadanych warunków (realizacja celu, adaptacja – analogia do przetrwania)
- **Eliminacja** najgorzej dopasowanych osobników
- **Rozmnażanie** pozostałych osobników

Oczekuje się, że nowa populacja ma więcej osobników dobrze zaadoptowanych do „życia” (działania i realizacji celu) w zadanym środowisku.

# Mini-Quiz

- Czy proces genetyczny gwarantuje osiągnięcie optymalnego rozwiązania? Tak/ Nie?

# Mini-Quiz

- Czy mutacja jest szkodliwa? Tak / nie / czasami nie

# Nazewnictwo

**Osobnik** – możliwe badane rozwiązanie

**Populacja** – zbiór osobników w danej iteracji

**Gen** – liczbowa reprezentacja rozwiązania (jego kluczowych aspektów poddawanych optymalizacji)

**Fenotyp** – fizyczna manifestacja zawartości genu, uzyskane rozwiązanie (wygląd, instynkt, cechy fizyczne istoty żywej)

**Dopasowanie (ang. Fitness)** – sposób oceny zachowania, czyli fenotypu

# Przykład – przybliżenie liczby Pi

- Cyfry: A, B, C, D, E, F, G, H, I, J

Gen

- Utworzyć ułamek:

$$\frac{ABCDE}{FGHIJ}$$

Fenotyp

- Wynik powinien być zbliżony do : 3.14159265359...

- Jak blisko jesteśmy celu:

- $\text{abs}\left(\frac{ABCDE}{FGHIJ} - \text{Pi}\right) = \text{error}$

Fitness

- $ABCDE = 10,000 \cdot A + 1,000 \cdot B + 100 \cdot C + 10 \cdot D + 1 \cdot E$

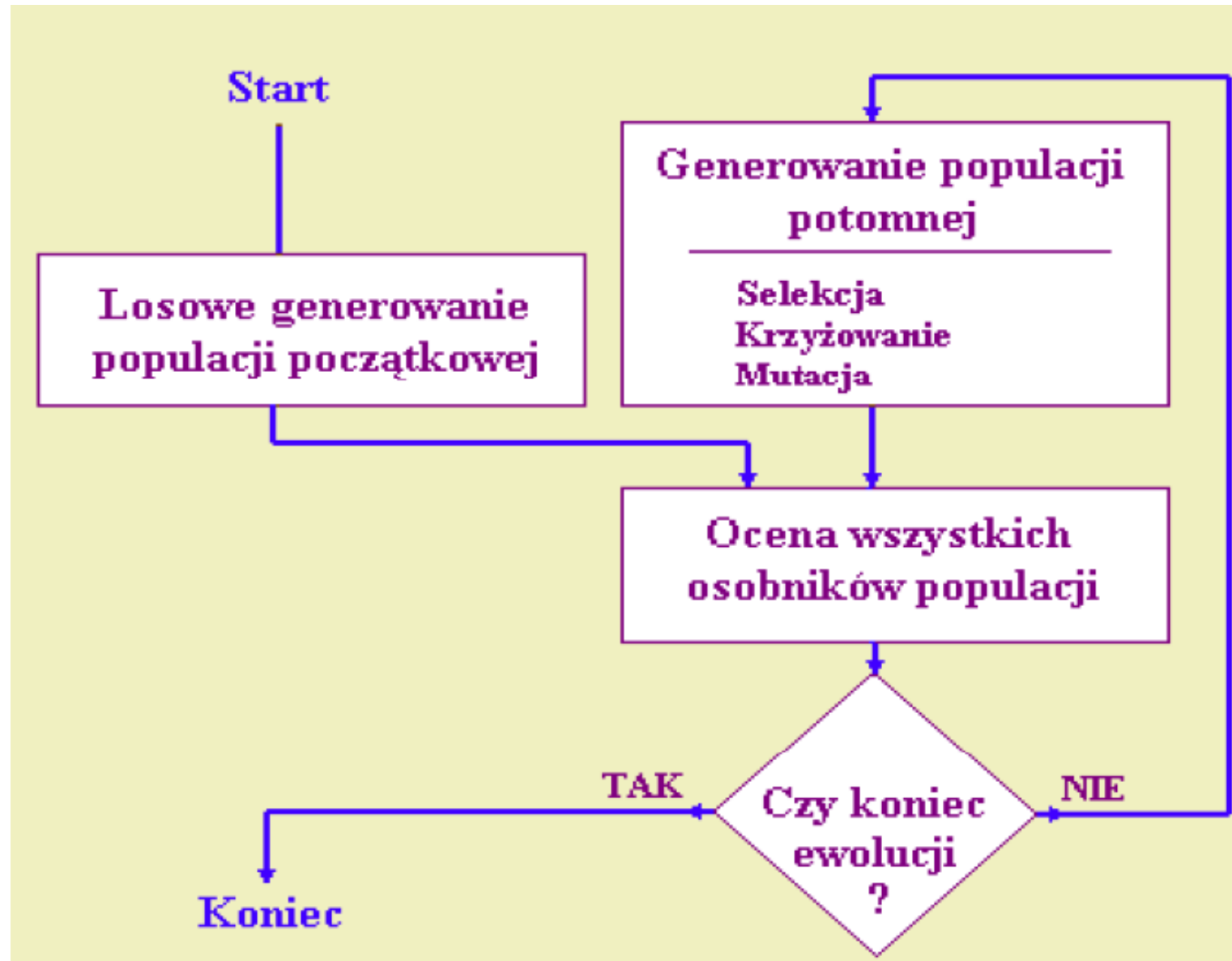
- Znany jest sposób pomiaru dopasowania (fitness)

- Nie jest znane wprost rozwiązanie!

# Proces genetyczny

1. Losowa inicjacja populacji
2. Ocena każdego osobnika
3. Selekcja na podstawie stopnia dopasowania
4. Użycie operacji genetycznych do generowania kolejnej populacji – potencjalnie lepszej
5. Powrót do kroku 2

# Proces genetyczny





# Rozmiar populacji

- Zbyt mały - 5 osobników dla problemu o 500,000 możliwych rozwiązaniach? (liczba rozwiązań określona kombinatorycznie, np. permutacja, kombinacja, itd.)
- Zbyt duży - 10,000 osobników dla problemu o 500,000 możliwych rozwiązaniach?

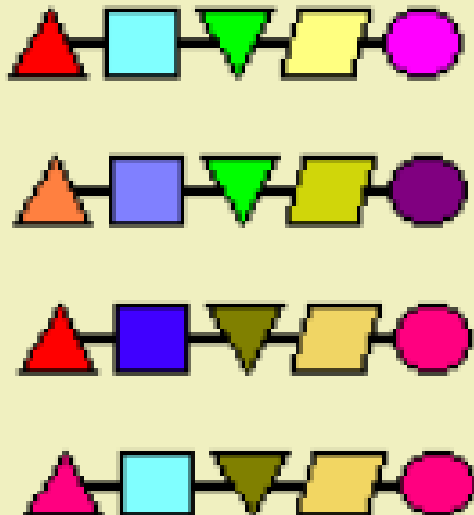
# Przykłady

- Przybliżenie liczby Pi (10 cyfr)
- Problem komiwojażera, odwiedzić K miast, bez powtórzeń.
- Problem plecakowy, pozycja i liczba pudełek w zadanej przestrzeni (kolejność i orientacja „pakunku”)
- Funkcja animacji robota (siły, momenty, długość ramion, sposób poruszania)

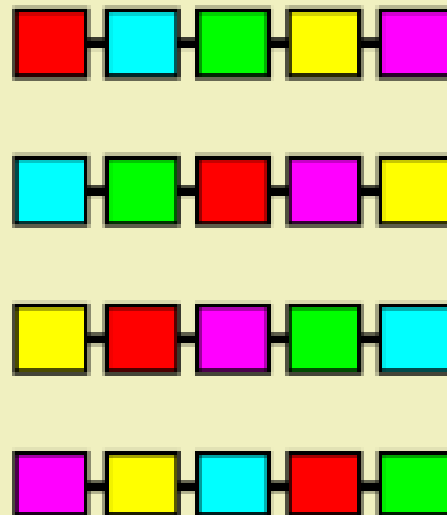
# Kodowanie genu

**Kodowanie** – zapisanie w formie liczbowej kluczowych cech fenotypu w genie. Decyzja na temat najlepszego sposobu wyrażenia cech w **formie liczbowej**

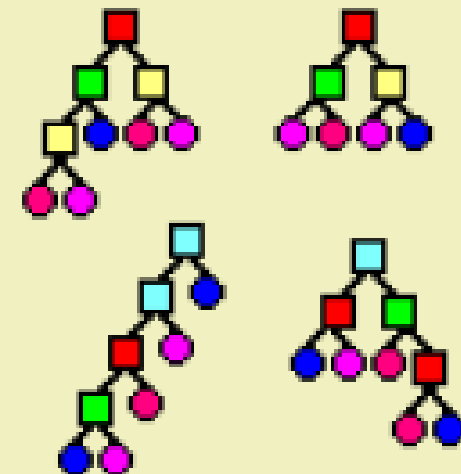
Klasyczne – wektor



Permutacyjne



Drzewiaste



# Strategie kodowania

**Klasyczne** – każdy gen przechowuje inny typ informacji. Znaczenie każdej pozycji się nie zmienia, ale zmieniają się wartości na pozycjach (w problemie liczby Pi: pierwszy to zawsze liczba tysięcy). *? Optymalizuje obiekt, proste cechy*

**Permutacyjny** – geny przechowują podobne informacje, wartości bez zmian, ale zmienia się kolejność, pozycja w ciągu. *? Optymalizuje kolejność w ramach procesu*

**Drzewiasty** – gen zorganizowany hierarchicznie, wyraża zależności między elementami, np. tworzenie wyrażeń matematycznych: dwie liczby – należy dodać, wynik należy podzielić, wynik tego należy... itd. *? Optymalizuje wiele współzależnych elementów procesu*

**Mini-Quiz: która strategia...**

# Funkcja dopasowania

**Funkcja dopasowania** wymaga obliczenia jednej oceny z całej treści genu. Wyraża stopień adaptacji do środowiska, stopień realizacji celu.

Wartość tę można minimalizować (np. błąd, zużycie zasobów) lub maksymalizować (wydajność)

# Jak mierzyć dopasowanie

- Przybliżenie Pi (error  $\rightarrow 0$ )
- Problem komiwojażera (czas, paliwo  $\rightarrow 0$ )
- Problem plecakowy (liczba pudełek  $\rightarrow \max$ , liczba pozostałego wolnego miejsca  $\rightarrow \max$ )
- Poruszanie robotem (zasoby  $\rightarrow 0$ , przebyty dystans  $\rightarrow \max$ )

# Przykład

- [Smart Rockets \(shivank1006.github.io\)](https://shivank1006.github.io)
- Losowe odpalanie silników i nawigacja, cel – dotarcie do celu
- Materiały: Shiffman D., The Nature of Code: ch9: The evolution of code: <https://natureofcode.com/book/chapter-9-the-evolution-of-code/>

# Selekcja genetyczna

- Gdy znana jest już wartość dopasowania każdego osobnika, podjąć należy decyzję, które z nich „przeżywają” i których cechy zostaną przekazane nowym osobnikom.
- Rozważyć należy:
  - Liczba osobników przeżywających
  - Metoda: ruletka, ranking, turniej



# Selekcja – liczba osobników

- Za mała (np. 2 z 1,000)?
- Za duża (500 z 1,000)?
  
- Rozważ tempo propagowania się korzystnych cech!
- Rozważ zapewnianie odpowiednio dużego zróżnicowania w każdej populacji!

# Selekcja: ruletka

Policzyć dopasowanie (fitness) osobnika  $i$ :  $F(i)$

Prawdopodobieństwo wyboru (przeżycia) jako względne dopasowanie na tle całej populacji:

$$p(i) = \frac{F(i)}{\sum_{i=1}^n F(i)}$$

Losuj osobniki biorąc pod uwagę ich prawdopodobieństwa.

$p(i)$  jest jak szerokie lub wąskie pole ruletki



# Selekcja: ranking i turniej

**Ranking** – posortuj wg stopnia dopasowania, wybierz N najlepszych, tzw. TOP50.

**Turniej** – losowo podziel na N grup, ustal wygrywającego w każdej grupie. W ten sposób uzyskana selekcja N osobników.

# Selekcja: do rozważenia

Jakie szanse ma słaby osobnik w każdej strategii selekcji: zerowe?  
niezerowe?

Najlepsze osobniki wcale nie muszą posiadać optymalnych wartości cech – po prostu są w danej populacji względnie najlepsi.

Optymalna wartość cechy może losowo pojawić się u osobnika gorszego – trzeba zapewnić „szansę” propagacji tej cechy.

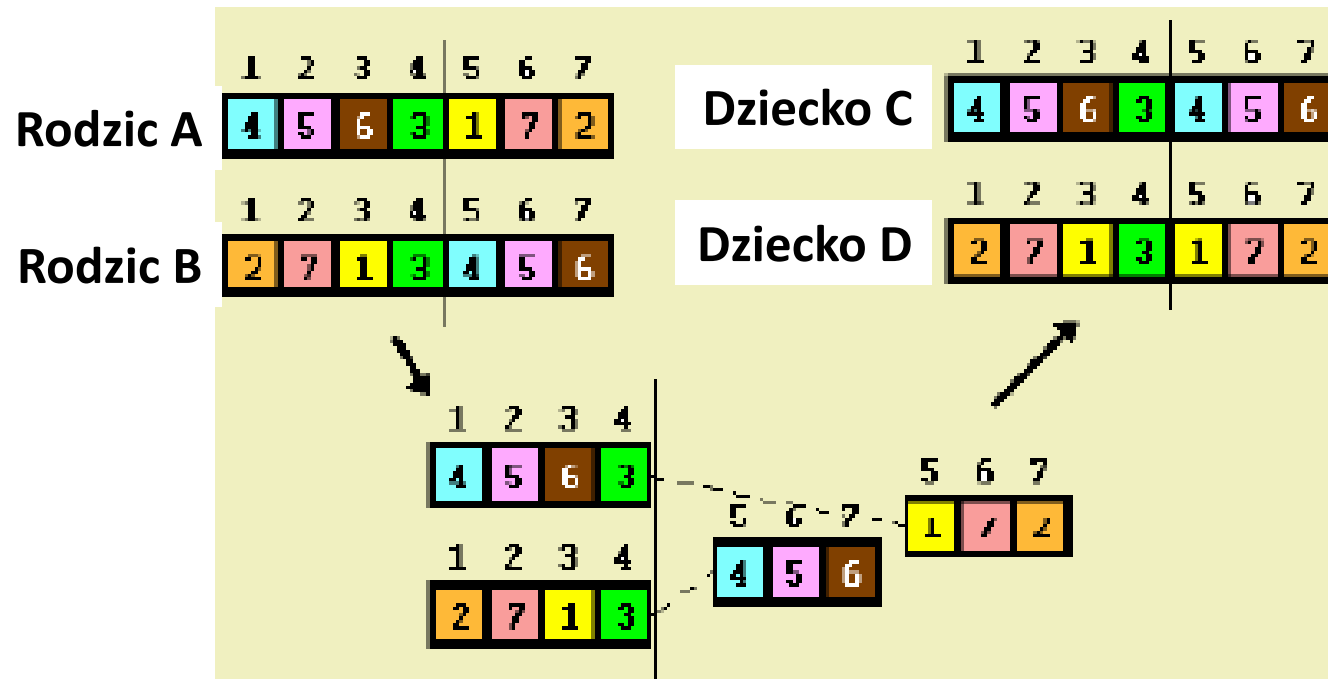
# Krzyżowanie genetyczne

Po selekcji osobniki, które „przetrwały” zostają rodzicami nowego pokolenia.

Nowe osobniki tworzone poprzez krzyżowanie genów rodziców.

Części genów losowo są wymieniane między parą rodziców A+B, nowe osobniki są tworzone w ten sposób: C, D. Wszystkie cztery osobniki A, B, C, D wchodzą do nowej populacji

Kompletnie nic nie da się powiedzieć o **przydatności konkretnych genów!**



# Znaczenie genów i krzyżowanie

- Dwa możliwe podejścia:
- Albo pojedynczy gen ma samodzielnie znaczenie, wtedy jego wymiana losowa poprawiać może osobniki,
- Albo grupy genów (współistniejące wartości) mają znaczenie, wtedy wymiana losowa całych grup może poprawić osobniki,
- Krzyżowanie pozostaje losowe
  - Wybierz losowo miejsce cięcia ciągu genów
  - Zamień części genów rodziców

# Mutacja

Losowa zmiana w zapisie genetycznym, niekontrolowana, może być szkodliwa lub korzystna.

Mutacja komórek jest skutkiem błędów podziału komórki: zmiana genu, usunięcie, wstawienie, zmiana całych ciągów.

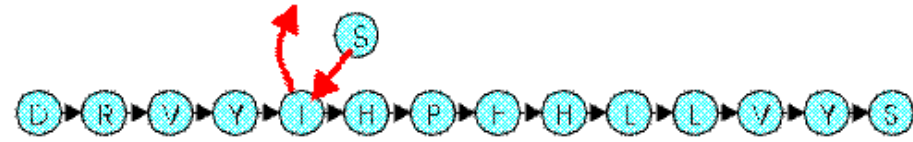
# Proces mutacji

Dla wybranego osobnika:

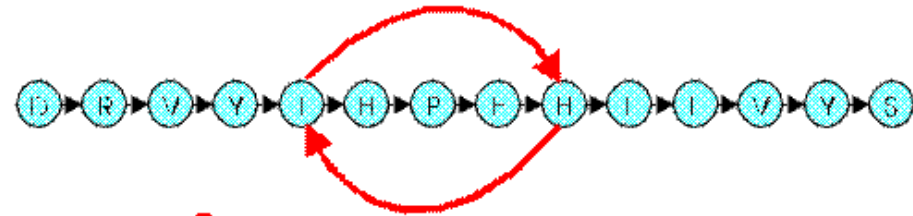
1. Losowo podejmij decyzję o wykonaniu mutacji
2. Losowo wybierz pozycję w genie
3. Losowo przypisz nową wartość



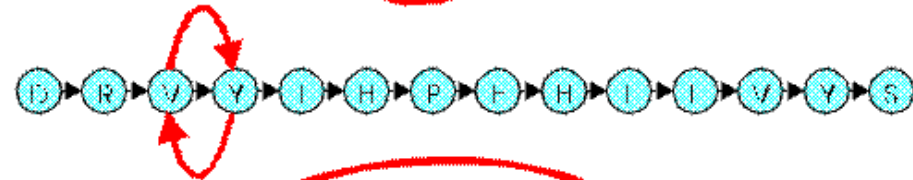
# Strategie mutacji



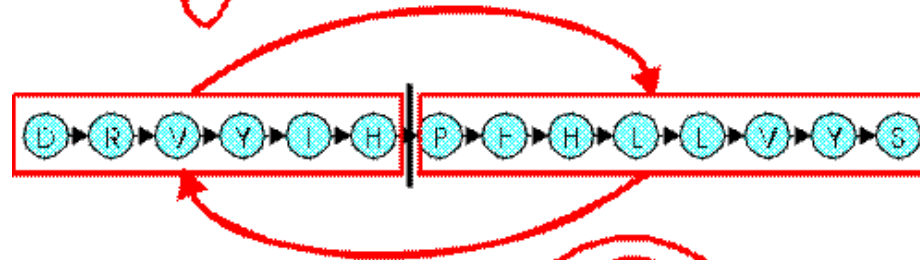
(a) Replacement



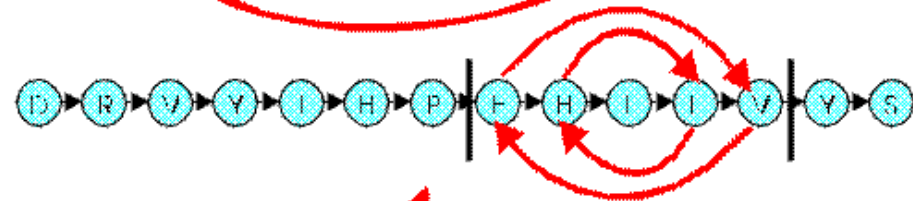
(b) Random swap



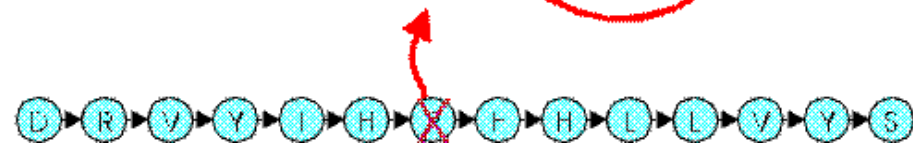
(c) Adjacent swap



(d) End-for-end swap

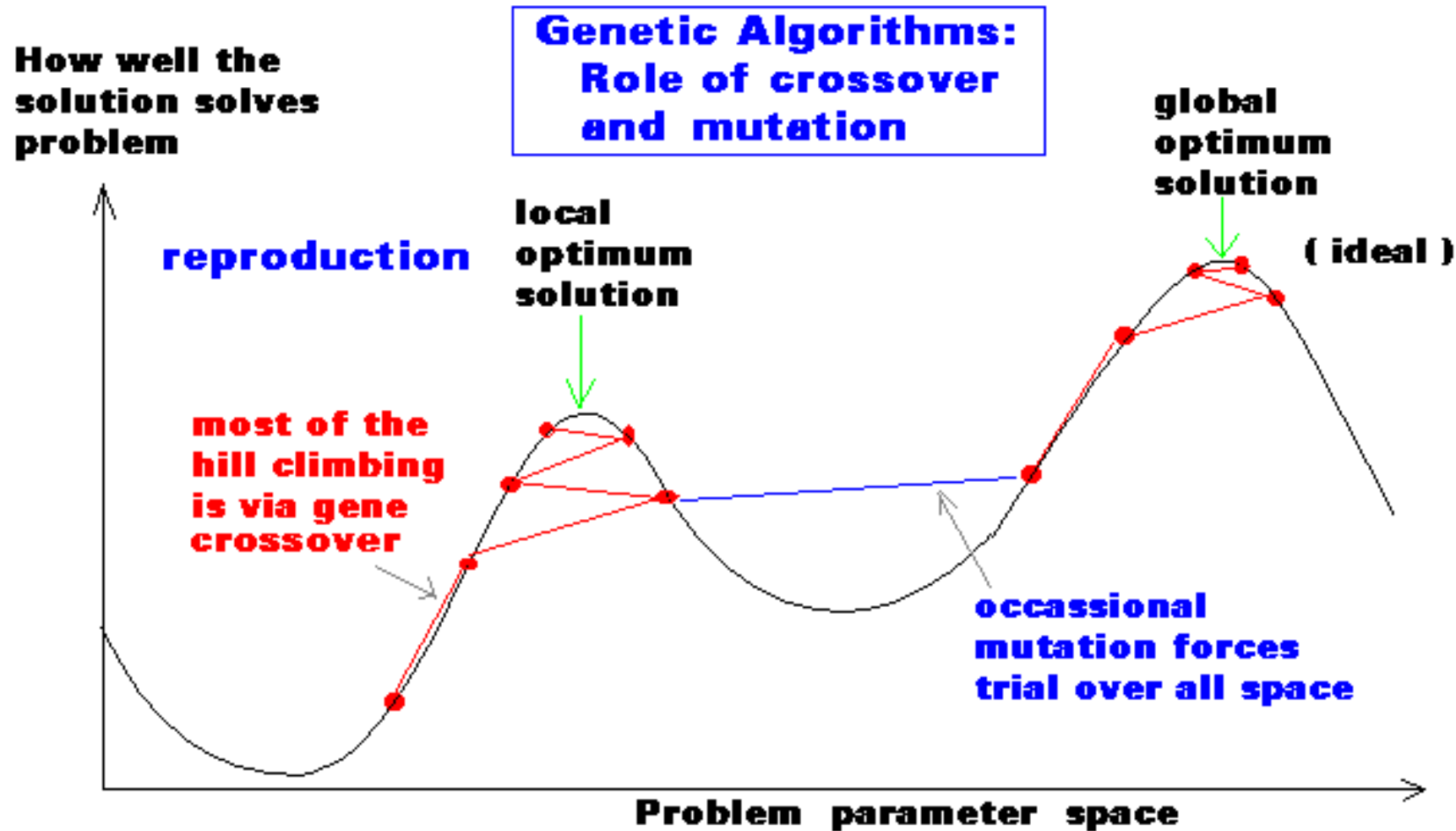


(e) Inversion



(f) Deletion

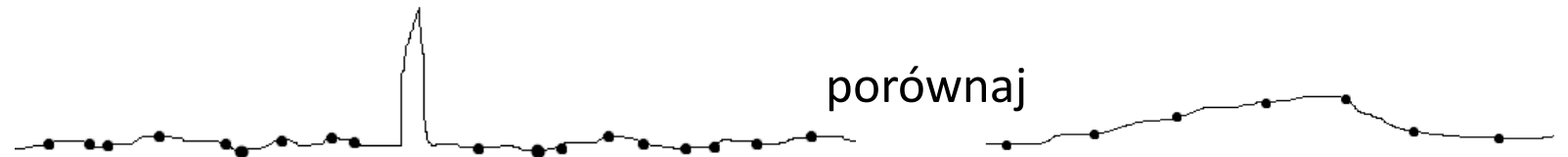
# Korzyści z krzyżowania i mutacji



# Ograniczenia algorytmu genetycznego

- Funkcja oceny może być trudna w obliczeniu (np. wymagana złożona symulacja fizyczna w celu określenia rezultatu, dla tysięcy osobników w iteracji)

- „Fitness landscape”:



- Przeważająca liczba rozwiązań z całego rozważanego zakresu może mieć równie niski fitness. Najlepsze może być bardzo specyficzne i występować wyłącznie w b. wąskim zakresie, pomijane i nigdy nie znalezione
- AG pomija nisze ewolucyjne i zwiększa podobieństwo między osobnikami
- Trudne w zastosowaniu w problemach klasyfikacyjnych, gdy fitness wyrażany jest tylko binarnie jako 0 lub 1 (niepoprawny, poprawny)
- Trudne w określeniu kryterium zatrzymania, co akceptujemy jako dostatecznie dobre rozwiązanie?

# Przykłady

- Połącz 50 przezroczystych wielokątów by uzyskać obraz:
- <https://rogerjohansson.blog/2008/12/07/genetic-programming-evolution-of-mona-lisa/>
- Modyfikowany kształt, kolor i przezroczystość każdego wielokąta

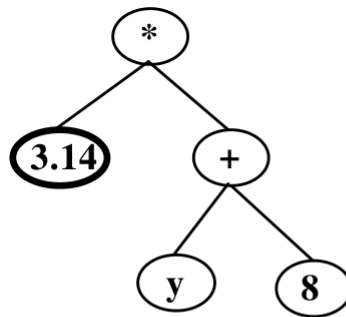
# Przykłady

- Symulacja dynamicznego zachowania:
  - Euphoria Physics Engine

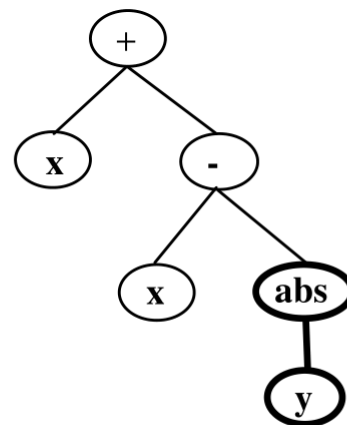


# Programowanie genetyczne

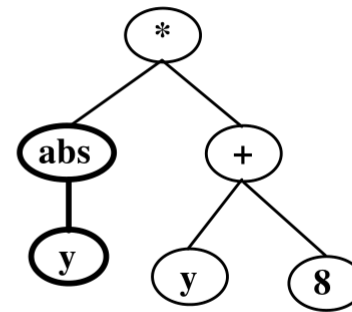
- Kodowanie drzewiaste – organizacja hierarchiczna elementów
- Elementy: liczby, zmienne, operacje
- Krzyżowanie: rozcinanie i wymiana gałęzi



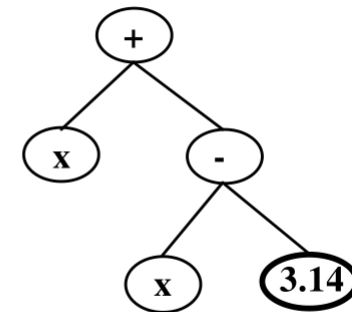
(\* 3.14 (+ y 8))



(+ x (- x (abs y)))



(\* (abs y) (+ y 8))



(+ x (- x 3.14))

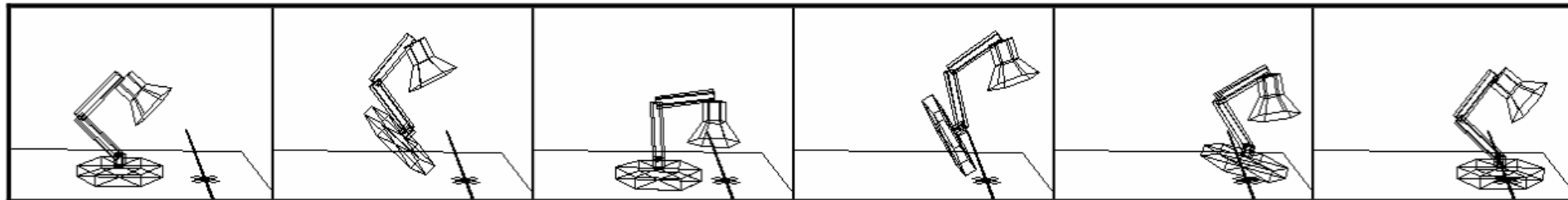
- Mutacja: losowe stałe, operatory, rozgałęzienia

# Przykład - Symulacja skoku

- Fitness: osiągnięcie celu w 2 skokach, punktacja za **styl**
- Cechy: położenie px,py,pz; kąty a0, a1, a2, itd.

## Styl – suma ważona:

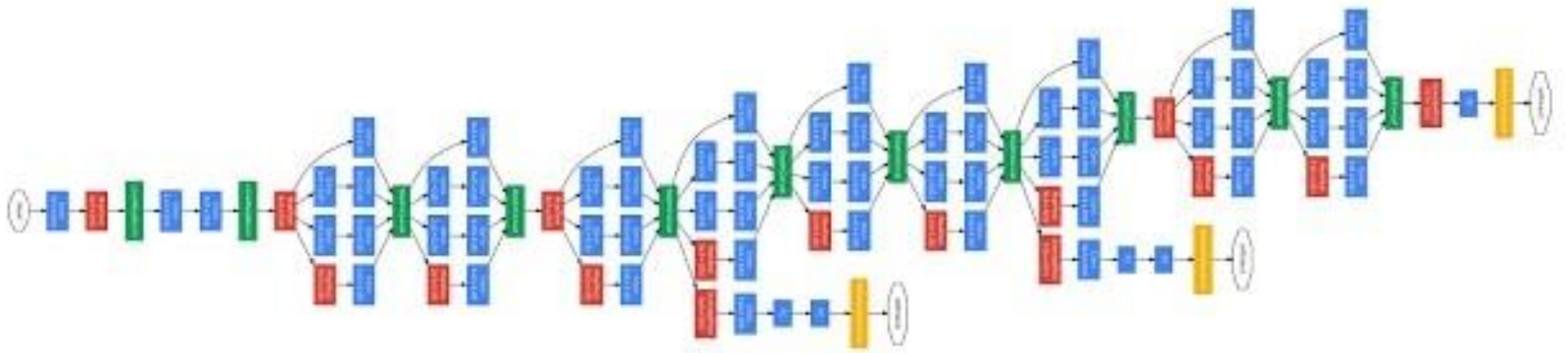
- a) Bonus za czas wykonania całego zadania
- b) Bonus za zakończenie ze stawami w pozycji neutralnej
- c) Kara za niestabilność po dotarciu do celu.
- d) Kara za uderzenie w „głowę” lub przewrócenie się.



```
(list
  (- (ifltz a0 pz a2) (% a1 t))
  (ifltz (- (- (ifltz a0 pz a2) (% a1 t)) (ifltz (+ a2 15.4963) (ifltz a0 pz a2) (% a1
    a2))) (ifltz (% (- s0 vx) (+ a2 s0)) (% (- s0 vx) (+ a2 s0)) (+ (- a1 a1) (- a1 t))) (%
    (+ a1 vz) (ifltz (* pz a1) (% a0 pz) (% px a0))))
  (- (ifltz s0 (ifltz (- (% vx a0) (% vx vx)) (- (% vx a0) (- (+ -28.4382 t) (% a1 t)))) (*
    16.5266 s0)) vz) (+ a1 vz)))
```

# Oprogramowanie ulepszające się

- Google AutoML (Machine Learning)
- Zdolne tworzyć ulepszoną wersję samego siebie



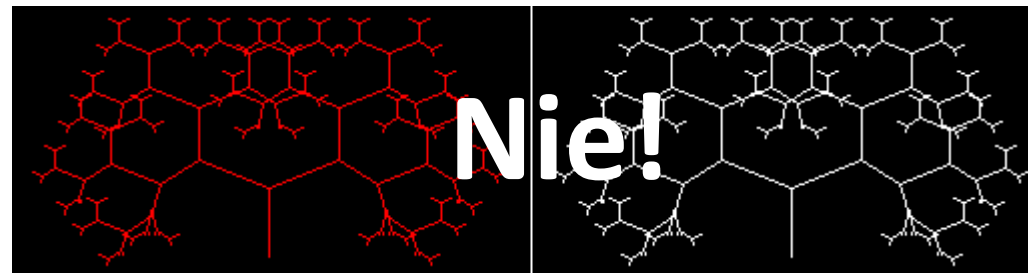


# Mini-Quiz

- Czy proces genetyczny tworzy rozwiązanie optymalne?
- Czy mutacja jest szkodliwa?

# Problemy „interesujące” dla AG

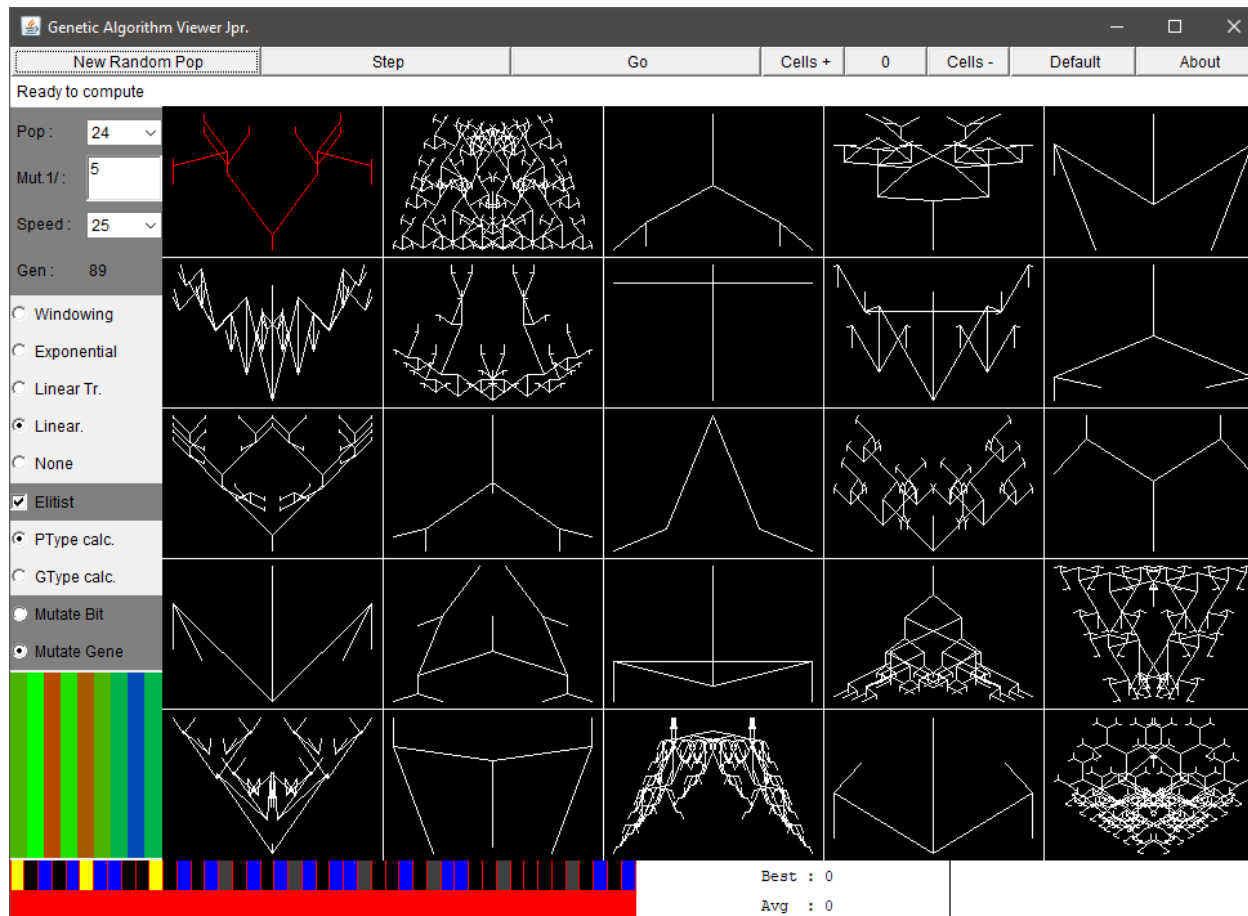
- Wiemy jak mierzyć dopasowanie
- Nie znamy z góry rozwiązania
- Algorytm musi w czymś nam pomagać!



<http://www.rennard.org/alife/english/gavgb.html>

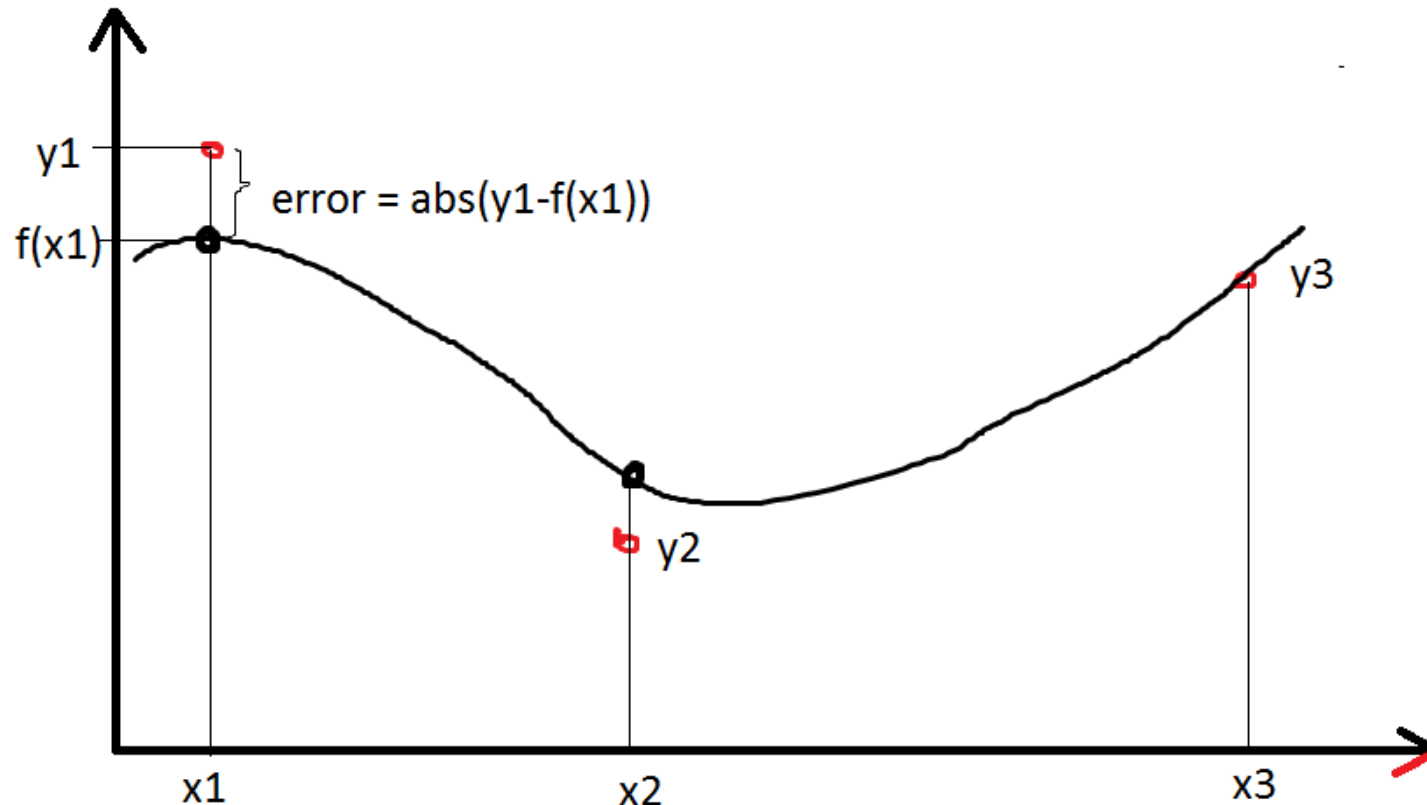
# Przykład – mniej „interesujący”

- Poszukiwane rozwiązanie jest znane od początku!



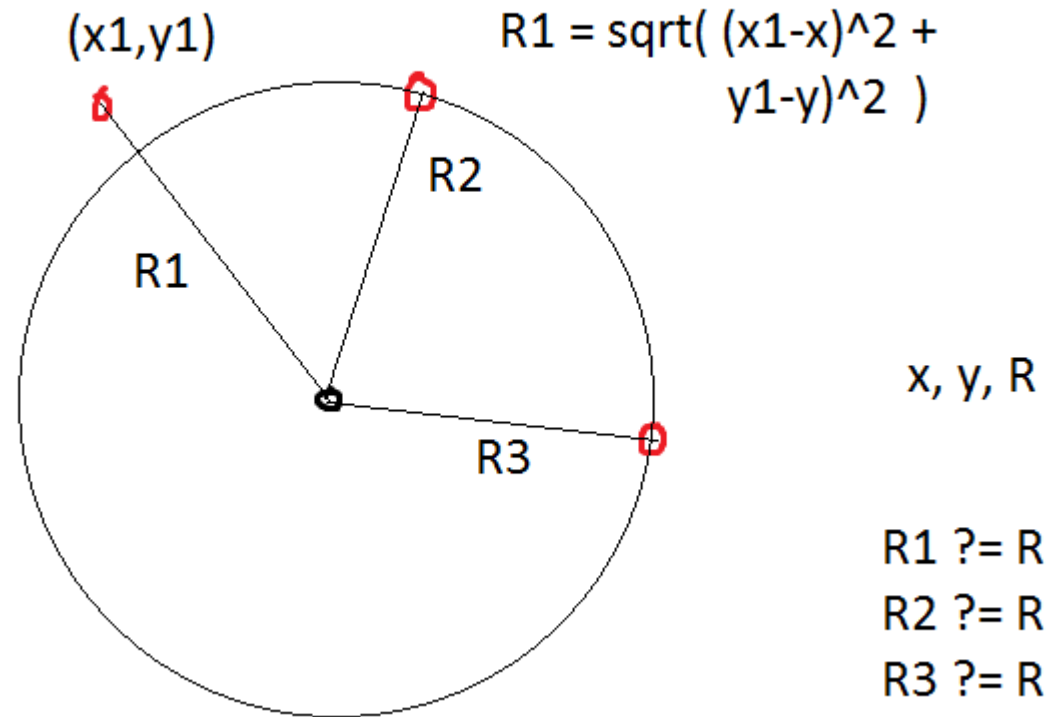
# Przykład

- Wielomian:  $y=f(x)=Ax^3+Bx^2+Cx+D$
- Jakie wartości współczynników  $A,B,C,D$  zapewnią, że  $f(x_i)=y_i$  dla danych par punktów  $(x_i,y_i)$ ?



# Przykład

- Znajdź współrzędne środka  $x, y$  okręgu, który przejdzie przez 3 punkty:



# Więcej zastosowań

- Slowik, A., Kwasnicka, H. Evolutionary algorithms and their applications to engineering problems. Neural Comput & Applic 32, 12363–12379 (2020). <https://doi.org/10.1007/s00521-020-04832-8>
  - **Genetic Algorithms** —energy fuels (EF), engineering electrical electronic (EEE), operations research management science (ORMS), engineering civil (EC),
  - **Genetic Programming** —engineering civil (EC), water resources (WR), energy fuels (EF), automation control systems (ACS),
  - **Differential evolution** —energy fuels (EF), automation control systems (ACS), engineering electrical electronics (EEE), engineering civil (EC),
  - **Evolution strategies** —construction building technology (CBT), energy fuels (EF), engineering civil (EC), engineering electrical electronic (EEE),
  - **Evolutionary programming**—construction building technology (CBT), engineering civil (EC), computer science software engineering (CSSE), transportation science technology (TST).



Dziękuję za uwagę

Piotr SZCZUKO

[szczuko@multimed.org](mailto:szczuko@multimed.org)



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego  
Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.