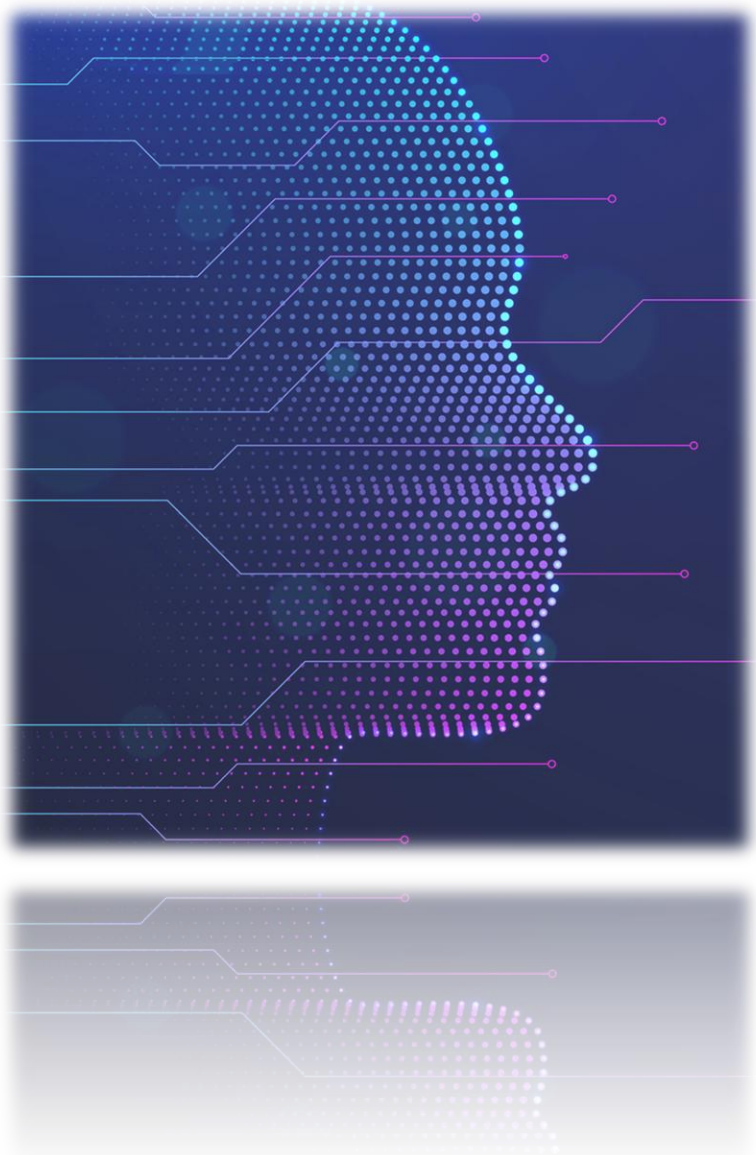
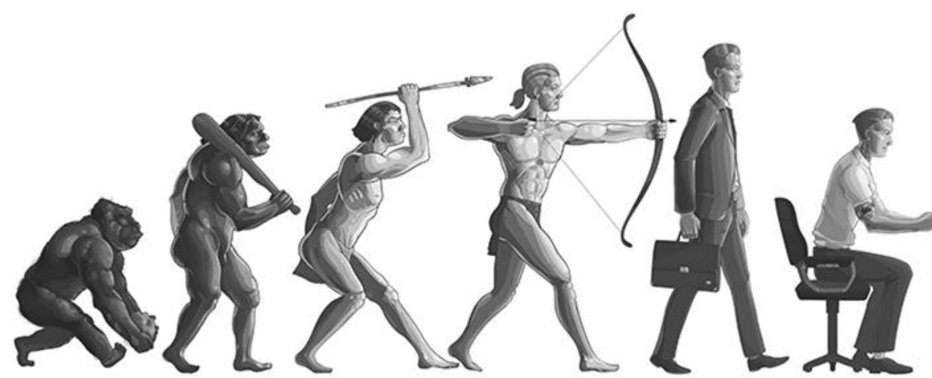
Inteligencja obliczeniowa w analizie danych



Metody ewolucyjne / Algorytmy genetyczne



Prof. dr hab. inż. Norbert Skoczylas

Gdzie jesteśmy

Algorytmy heurystyczne

Algorytmy probabilistyczne

Algorytmy genetyczne

Strategie ewolucyjne

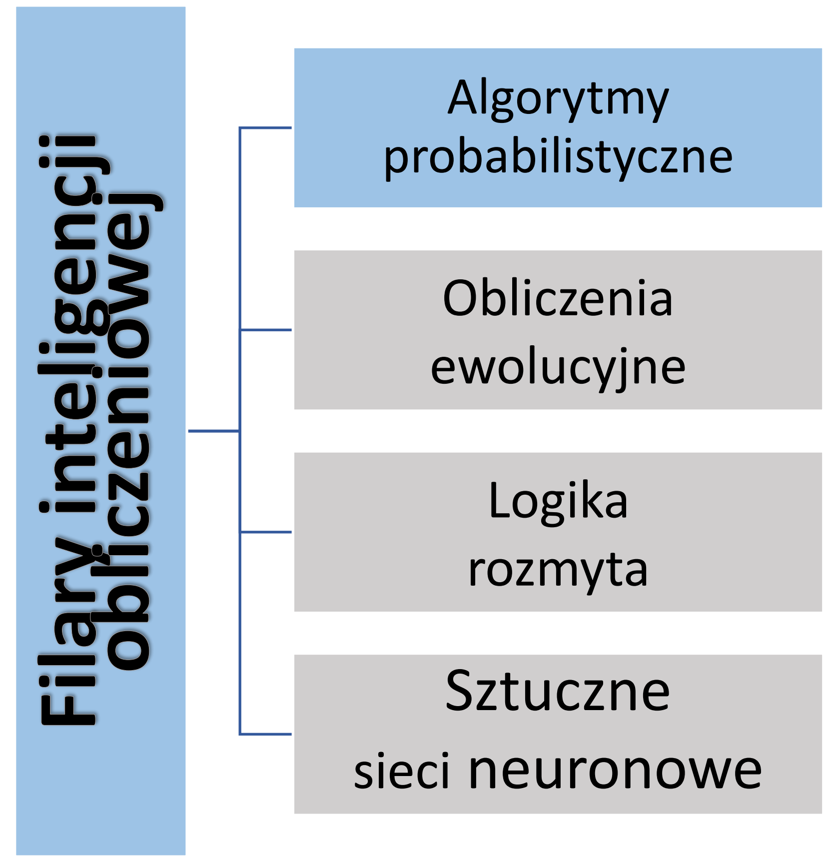
Metody roju cząstek

Logika rozmyta

Rozmyte systemy wnioskujące

Sterowanie rozmyte

Sztuczne sieci neuronowe



Algorytmy genetyczne

to kolejne narzędzie zaliczane
do metod sztucznej inteligencji –
idea jest inspirowana
przez naturę
(podobnie jak większość
metod inteligencji obliczeniowej)

Algorytmy heurystyczne

Heurystyka – metoda znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego, a często nawet prawidłowego.

Rozwiązań tych używa się np. wtedy, gdy pełny algorytm jest z przyczyn technicznych zbyt kosztowny lub gdy jest nieznany.

Algorytm dokładny – (algorytm siłowy) rozwiązania jakiegoś zadania,
 polega na wyczerpaniu wszystkich możliwości e celu znalezienia optymalnego rozwiązania problemu

Zadania optymalizacyjne

Problem optymalizacyjny (wikipedia)

– problem obliczeniowy, którego rozwiązanie polega na znalezieniu największej bądź najmniejszej wartości pewnego parametru problemu, która spełnia określoną własność…

Optymalizacja –
poszukiwanie najlepszego rozwiązania z punktu widzenia przyjętego kryterium
Zadanie optymalizacji (minimalizacji):

$$\boldsymbol{a}^{opt} = \arg \underset{a \in D}{min}\ fitness\,(\boldsymbol{a})$$

gdzie:

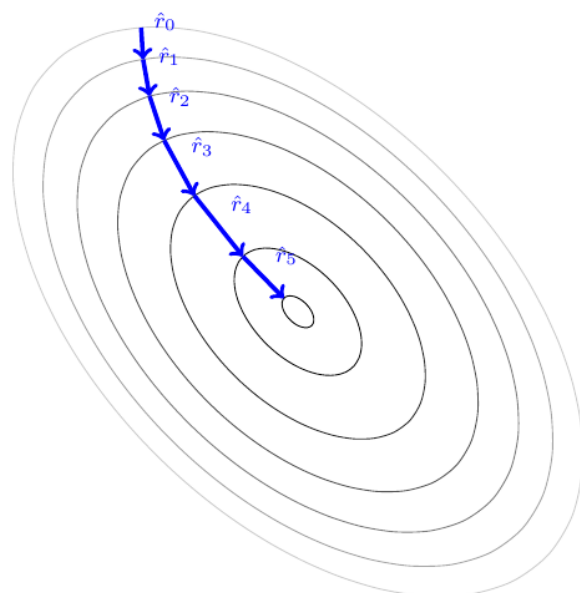
$\boldsymbol{a}$ - wektor reprezentujący rozwiązanie
$\boldsymbol{a}^{opt}$ - wektor rozwiązania optymalnego,
$fitness\,(\boldsymbol{a})$ - funkcja celu,
D - dziedzina rozwiązań.

Rozwiązywanie zadań optymalizacyjnych

Metody analityczne

- Bezpośrednie (skakanie po wykresie funkcji celu w kierunku gradientu)
- Pośrednie (porównanie gradientu funkcji celu do zera – układ równań)

metody gradientowe –
kierunek poszukiwań określa gradient funkcji celu



$$\nabla \boldsymbol{f}itness(a) = \left[\frac{\partial f}{\partial a_1}, \frac{\partial f}{\partial a_2}, \ldots, \frac{\partial f}{\partial a_n}\right]$$

Rozwiązywanie zadań optymalizacyjnych

Metody analityczne

- Bezpośrednie (skakanie po wykresie funkcji celu w kierunku gradientu)
- Pośrednie (porównanie gradientu funkcji celu do zera – układ równań)

Metody przeglądowe

- Przeglądamy wszystkie rozwiązania i analizujemy wartość funkcji celu

(złożoność obliczeniowa !)

Heurystyki

- Losowe przeszukiwanie przestrzeni rozwiązań
- Ukierunkowane pseudolosowe przeszukiwanie przestrzeni rozwiązań

Konsekwencje złożoności obliczeniowej

Większość problemów optymalizacyjnych jest obarczona trudnościami wynikającymi ze złożoności obliczeniowej…

Problemy o charakterze optymalizacyjnym sprowadzają się do zagadnienia minimalizacji, bądź maksymalizacji (optymalizacji) pewnej funkcji celu – funkcji opisującej jakość optymalizacji …

Optymalizacja rozwiązania wymaga najczęściej analizy kombinacji rozwiązań.

Rozwiązania często podlegają ograniczeniom.

Rozwiązania kodowane są często za pomocą wektorów lub macierzy symboli.

W wielu przypadkach prowadzi to do znacznej czaso-chłonności i pamięcio-chłonności rozwiązań.

Pytanie – jak zależy ilość czasu i pamięci potrzebna do rozwiązania problemu od wielkości problemu ? – złożoność obliczeniowa...

Przykład 1:

Mamy N elementów zbioru i musimy skonfrontować każdy z każdym – jaka złożoność ?
(np. turniej każdy z każdym) – ile jest par elementów ?

Ile byłoby par dla:	2 osób ?	- 1
	3 osób ?	- 3
	4 osób ?	
	5 osób ?	

	N osób?	

Przykład 1:

Mamy N elementów zbioru i musimy skonfrontować każdy z każdym – jaka złożoność ?
(np. turniej każdy z każdym) – ile jest par elementów ?

Ile byłoby par dla: 2 osób ? - 1
3 osób ? - 3
4 osób ?
5 osób ?

N osób?

	1	2	3	4	5	6
1	-	x	x			
2		-	x			
3			-			
4						
5						
6						

Przykład 1:

Mamy N elementów zbioru i musimy skonfrontować każdy z każdym – jaka złożoność ?
(np. turniej każdy z każdym) – ile jest par elementów ?

Ile byłoby par dla:	2 osób ?	- 1
	3 osób ?	- 3
	4 osób ?	- 6
	5 osób ?	

	N osób?	

	1	2	3	4	5	6
1	-	x	x	x		
2		-	x	x		
3			-	x		
4				-		
5						
6						

Przykład 1:

Mamy N elementów zbioru i musimy skonfrontować każdy z każdym – jaka złożoność ?
(np. turniej każdy z każdym) – ile jest par elementów ?

Ile byłoby par dla: 2 osób ? - 1
3 osób ? - 3
4 osób ? - 6
5 osób ? - 10

N osób?

	1	2	3	4	5	6
1	-	x	x	x	x	
2		-	x	x	x	
3			-	x	x	
4				-	x	
5					-	
6						

Przykład 1:

Mamy N elementów zbioru i musimy skonfrontować każdy z każdym – jaka złożoność ?
(np. turniej każdy z każdym) – ile jest par elementów ?

Ile byłoby par dla:	2 osób ?	- 1
	3 osób ?	- 3
	4 osób ?	- 6
	5 osób ?	- 10

	N osób?	- $(n^2-n)/2$

	1	2	3	4	5	6
1	-	X	X	X	X	X
2		-	X	X	X	X
3			-	X	X	X
4				-	X	X
5					-	X
6						-

Przykład 1:

Mamy N elementów zbioru i musimy skonfrontować każdy z każdym – jaka złożoność ?
(np. turniej każdy z każdym) – ile jest par elementów ?

Ile byłoby par dla: 2 osób ? - 1
3 osób ? - 3
4 osób ? - 6
5 osób ? - 10

N osób? - $(n^2-n)/2$

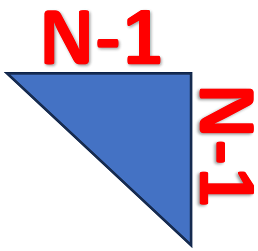
Problemy o złożoności wielomianowej

N

N-1

N-1

	1	2	3	4	5	6
1	-	x	x	x	x	x
2		-	x	x	x	x
3			-	x	x	x
4				-	x	x
5					-	x
6						-



Przykład 2:

Mamy N elementów zbioru i musimy wybrać z niego n podzbiorów
Na ile sposobów możemy to zrobić – jaka złożoność ?

N=1: Ø , 1
N=2: Ø , 1 , 2 , 1 2
N=3:
N=4:

N=1: n=2
N=2: n=4

Przykład 2:

Mamy N elementów zbioru i musimy wybrać z niego n podzbiorów
Na ile sposobów możemy to zrobić – jaka złożoność ?

N=1: Ø , 1
N=2: Ø , 1 , 2 , 1 2
N=3: Ø , 1 , 2 , 3 , 1 2 , 1 3 , 2 3 , 1 2 3
N=4:

N=1: n=2
N=2: n=4
N=3: n=8

Przykład 2:

Mamy N elementów zbioru i musimy wybrać z niego n podzbiorów
Na ile sposobów możemy to zrobić – jaka złożoność ?

N=1: Ø , 1
N=2: Ø , 1 , 2 , 1 2
N=3: Ø , 1 , 2 , 3 , 1 2 , 1 3 , 2 3 , 1 2 3
N=4: Ø , 1 , 2 , 3 , 4, 1 2 , 1 3 , 1 4 , 2 3 , 2 4 , 3 4 , 1 2 3, 1 2 4 , 1 3 4 , 2 3 4 , 1 2 3 4

N=1: n=2
N=2: n=4
N=3: n=8
N=4: n=16

Przykład 2:

Mamy N elementów zbioru i musimy wybrać z niego n podzbiorów
Na ile sposobów możemy to zrobić – jaka złożoność ?

N=1: Ø , 1
N=2: Ø , 1 , 2 , 1 2
N=3: Ø , 1 , 2 , 3 , 1 2 , 1 3 , 2 3 , 1 2 3
N=4: Ø , 1 , 2 , 3 , 4, 1 2 , 1 3 , 1 4 , 2 3 , 2 4 , 3 4 , 1 2 3, 1 2 4 , 1 3 4 , 2 3 4 , 1 2 3 4

N=1: n=2
N=2: n=4
N=3: n=8
N=4: n=16

$n=2^n$

Analogia – ile stanów logicznych możemy zakodować ***n*** bitami ?

Przykład 2:

Mamy N elementów zbioru i musimy wybrać z niego n podzbiorów
Na ile sposobów możemy to zrobić – jaka złożoność ?

N=1: Ø , 1
N=2: Ø , 1 , 2 , 1 2
N=3: Ø , 1 , 2 , 3 , 1 2 , 1 3 , 2 3 , 1 2 3
N=4: Ø , 1 , 2 , 3 , 4, 1 2 , 1 3 , 1 4 , 2 3 , 2 4 , 3 4 , 1 2 3, 1 2 4 , 1 3 4 , 2 3 4 , 1 2 3 4

N=1: n=2
N=2: n=4
N=3: n=8
N=4: n=16

$$\mathbf{n=2^n}$$

Problemy o złożoności wykładniczej

Analogia – ile stanów logicznych możemy zakodować ***n*** bitami ?

Przykład 3:

Problem komiwojażera

Na ile sposobów możemy pojechać mięrzy N miastami, żeby w każdym być dokładnie jeden raz?
Jaka jest złożoność obliczeniowa ?

1: Losujemy jedno z N miast – N możliwości

Przykład 3:

Problem komiwojażera

Na ile sposobów możemy pojechać mięrzy N miastami, żeby w każdym być dokładnie jeden raz?
Jaka jest złożoność obliczeniowa ?

1: Losujemy jedno z N miast – N możliwości

2: Z każdego miasta pozostaje nam N-1 możliwości

$N \cdot (N-1)$

Przykład 3:

Problem komiwojażera

Na ile sposobów możemy pojechać mięrzy N miastami, żeby w każdym być dokładnie jeden raz?
Jaka jest złożoność obliczeniowa ?

1: Losujemy jedno z N miast – N możliwości

2: Z każdego miasta pozostaje nam N-1 możliwości

3: Z każdego miasta pozostaje nam N-2 możliwości

$N \cdot (N-1) \cdot (N-2) \cdot$

Przykład 3:

Problem komiwojażera

Na ile sposobów możemy pojechać mięrzy N miastami, żeby w każdym być dokładnie jeden raz?
Jaka jest złożoność obliczeniowa ?

1: Losujemy jedno z N miast – N możliwości

2: Z każdego miasta pozostaje nam N-1 możliwości

3: Z każdego miasta pozostaje nam N-2 możliwości

.....

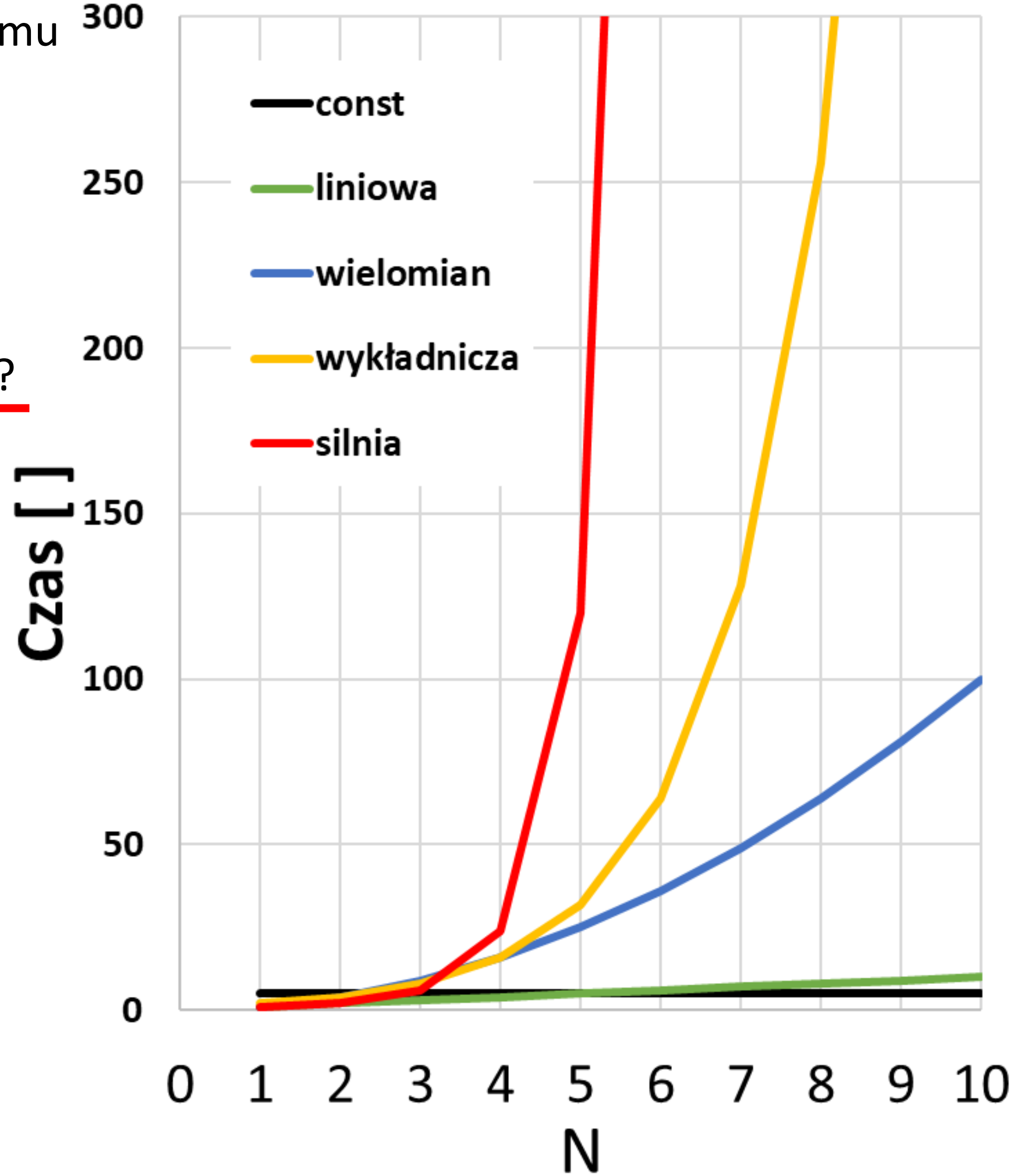
N-1 Pozostało nam ostatnie miast do odwiedzania

$N \cdot (N-1) \cdot (N-2) \cdot \ldots \cdot 2 \cdot 1 = N!$

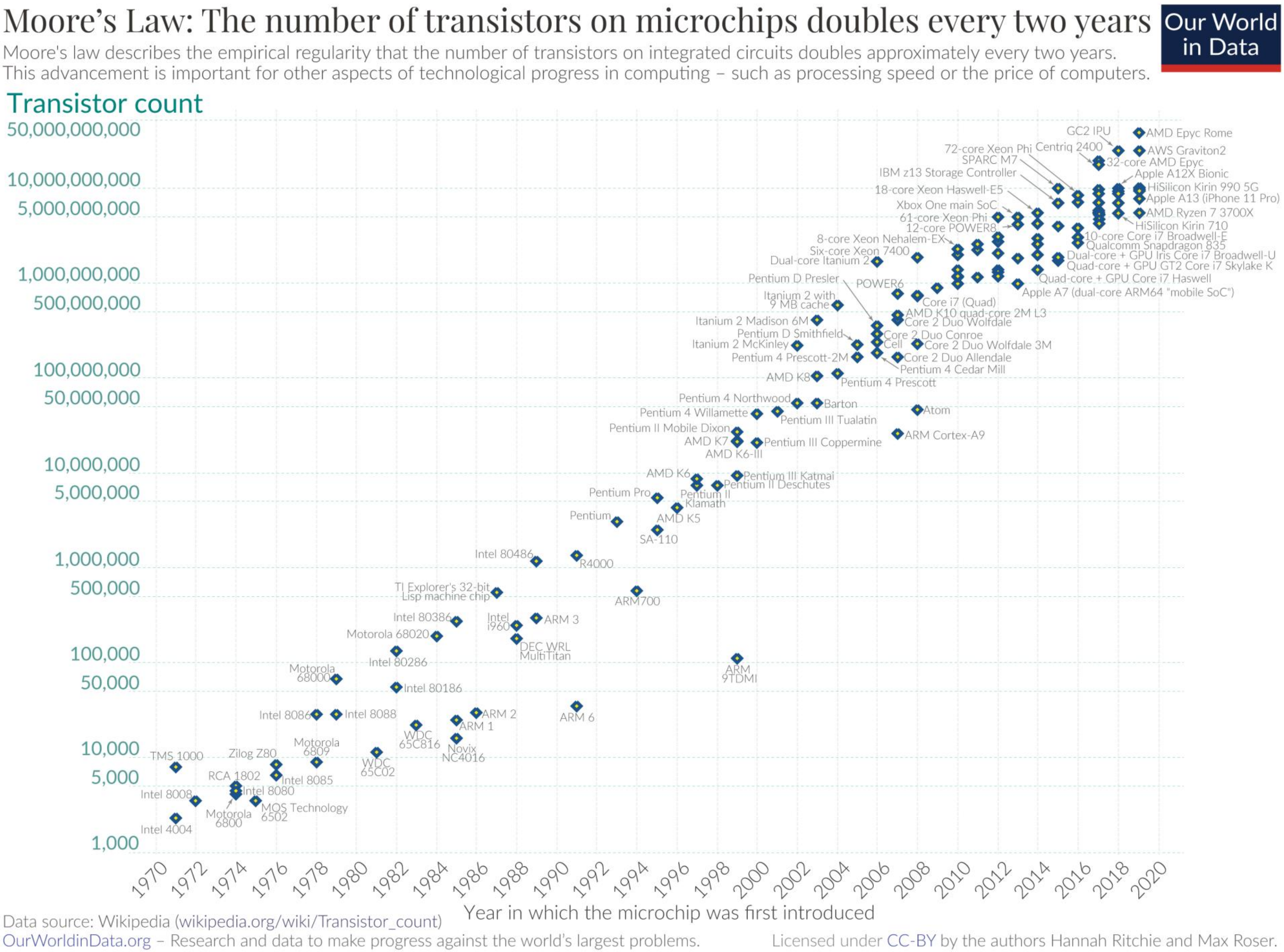
Pytanie – jak zależy ilość czasu i pamięci potrzebna do rozwiązania problemu od wielkości problemu ? – złożoność obliczeniowa...

jak czas potrzebny na rozwiązanie problemu wzrasta wraz z wielkością problemu ?

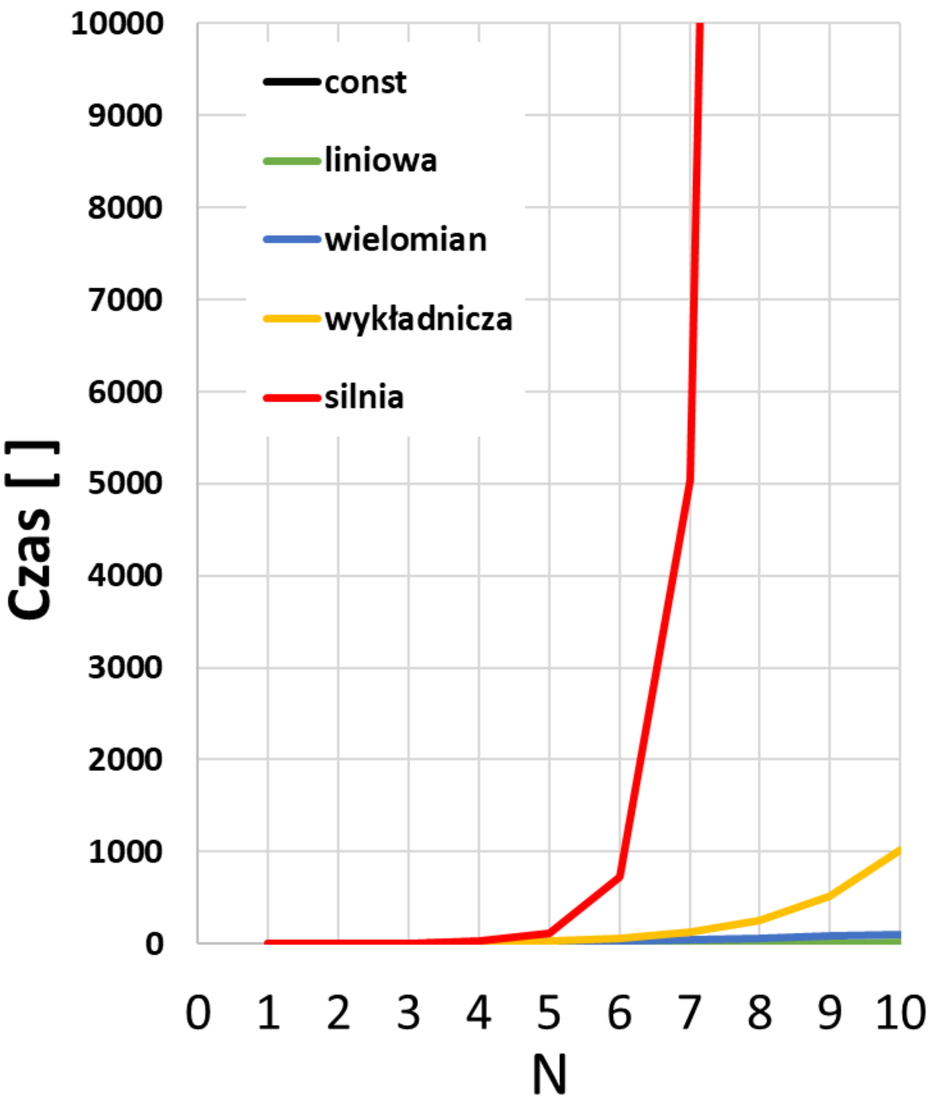
- nie wzrasta ?
- rośnie liniowo?
- rośnie wielomianowo?
- rośnie wykładniczo?
- czy rośnie w tempie zależnym od N! ??

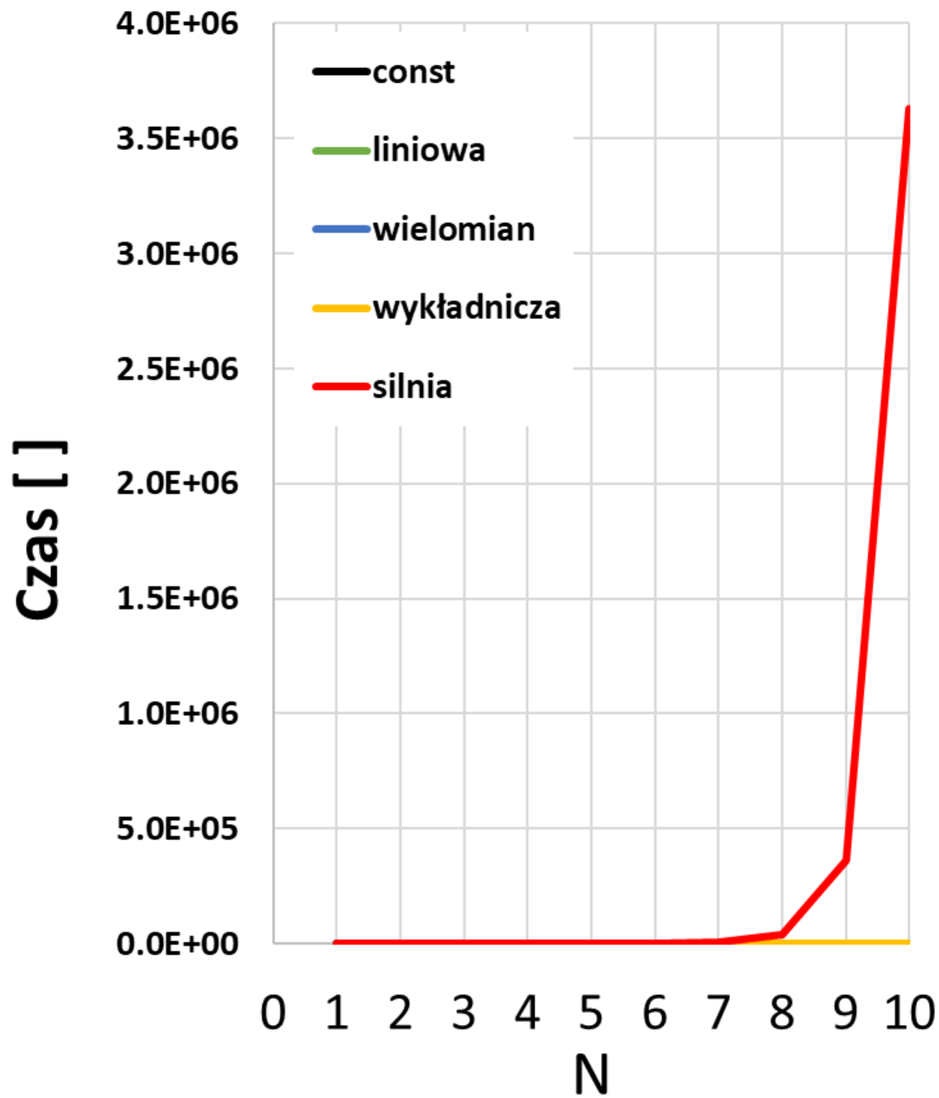


Przełożenie złożoności obliczeniowej na zapotrzebowanie na pamięć już nas tak bardzo nie boli ..



Ale w przypadku wydajności, odbiliśmy się od sufitu częstotliwości taktowania …
Postęp dokonuje się głównie w dziedzinie zrównoleglania obliczeń – nie każdy problem łatwo daje się zrównoleglić …





Algorytmy genetyczne

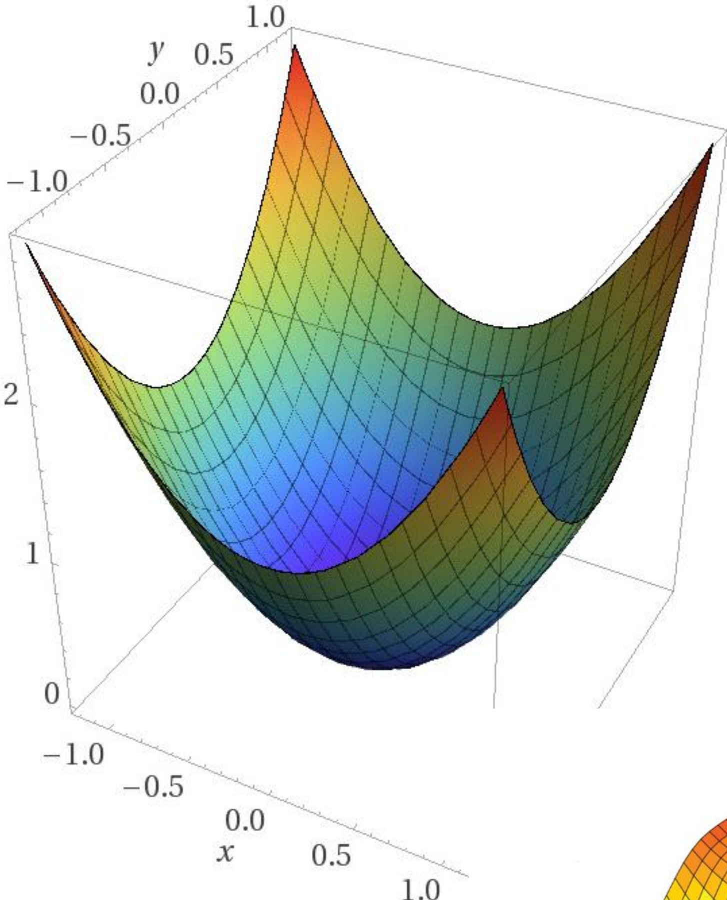
Algorytmy genetyczne są dedykowane
do rozwiązywania <u>zadań optymalizacyjnych</u> !
(z pośród dopuszczalnych rozwiązań problemu wybieramy najlepsze
ze względu na przyjęte kryterium (koszt, czas ..)

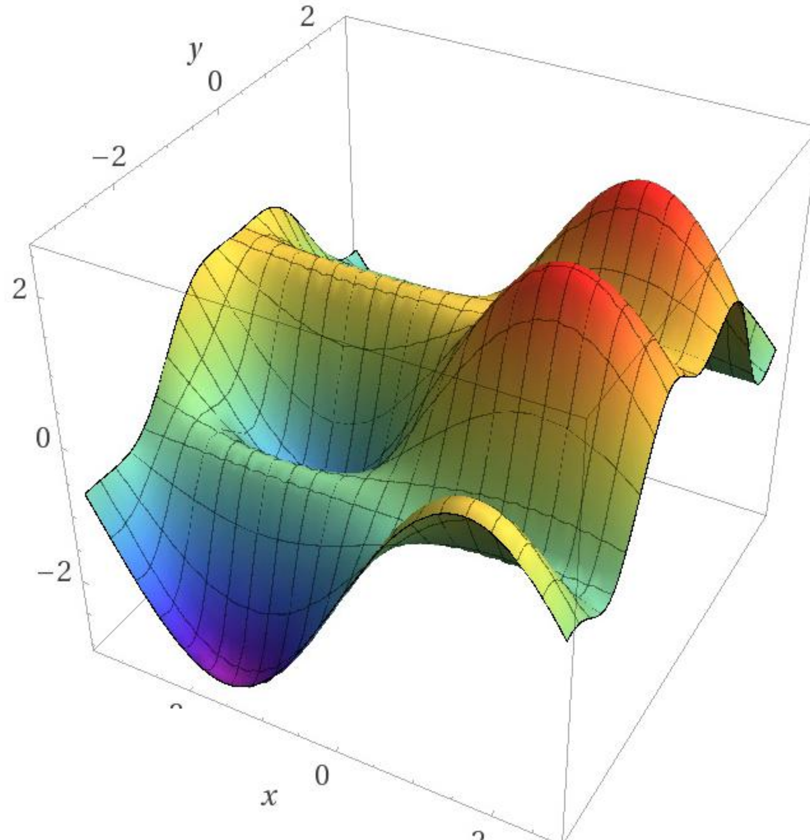
Uzyskanie pożądanego rozwiązania
na drodze ewolucji

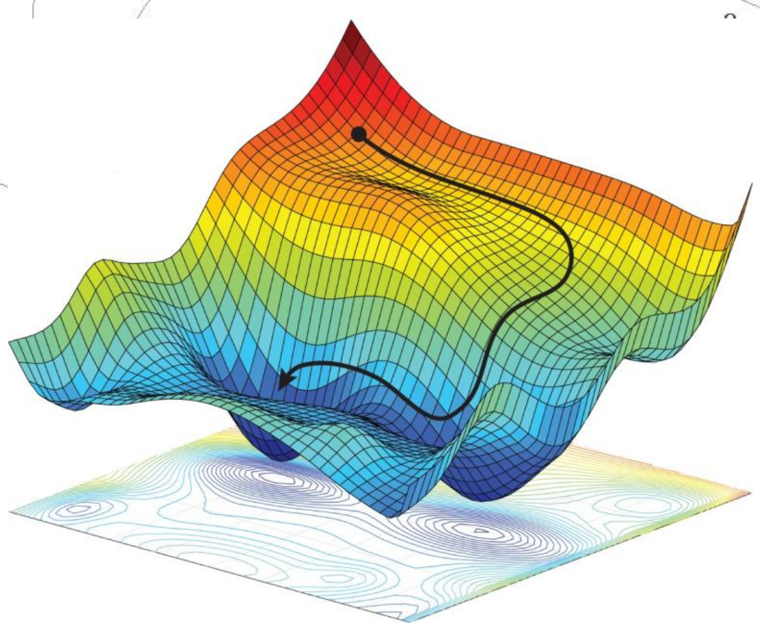
Rozwiązaniem problemu optymalizacji jest minimalizacja funkcji jakości zdefiniowanej jako koszt, bądź maksymalizacja funkcji jakości określonej jako zysk.

Algorytmy genetyczne są najważniejszą grupą metod klasyfikowanych jako metody ewolucyjne

Funkcja jakości, którą minimalizujemy, może mieć różne kształty



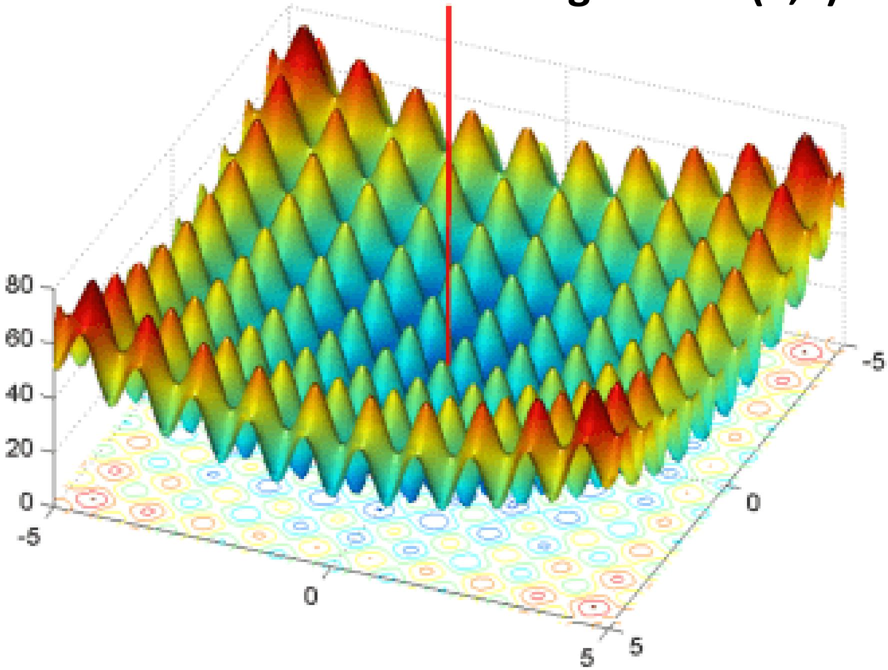


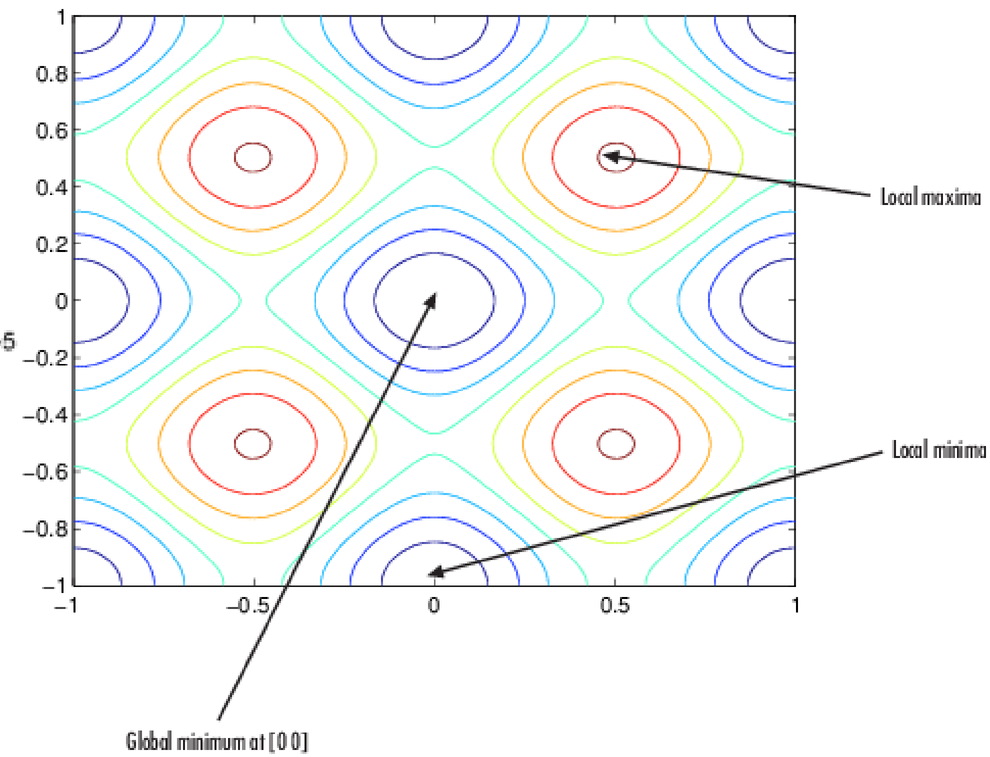


Funkcja jakości, którą minimalizujemy, może mieć różne kształty

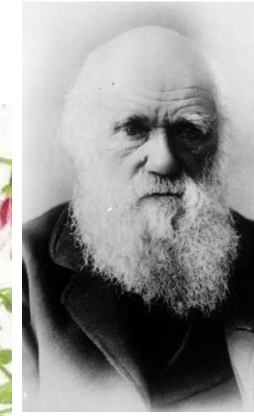
Rastrigin's Function: $Ras(x)=20+x^2_1+x^2_2-10(\cos 2\pi x1+\cos 2\pi x2)$.

Minimum globalne (0,0)

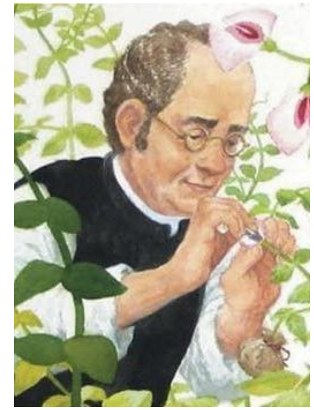




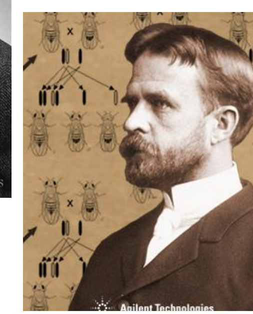
Krótka historia genetyki



1859 **Karol Darwin** – ..o pochodzeniu gatunków.. - twórca teorii ewolucji, zgodnie z którą wszystkie gatunki pochodzą od wcześniejszych form, Darwin uważał, że rozgałęziony schemat ewolucji wynika z procesu, który nazwał doborem naturalnym.



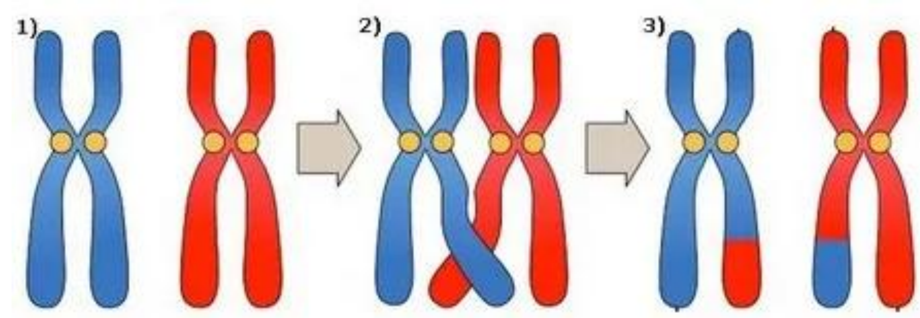
1866 **Gregor Mendel** - ..badania nad mieszańcami roślin.. - prekursor genetyki, określił prawa dziedziczenia na podstawie badań nad krzyżowaniem grochu jadalnego.



Thomas Hunt Morgan - ogłosił **chromosomową teorię dziedziczności** - teoria, według której czynniki dziedziczności – geny – są jednostkami materialnymi i znajdują się na chromosomach, są ułożone liniowo i zajmują ściśle określone miejsca.

1910 Hermann **Joseph Muller** - wykazał, że promieniowanie rentgenowskie wywołuje **mutacje.**

1913 Alfred Sturtevant - zasugerował możliwość zachodzenia Crossing-over ->



Najważniejsze prawa dziedziczenia, z których czerpią algorytmy genetyczne

Ewolucja przez dobór naturalny **oznacza, że przeżywają i rozmnażają się osobniki najlepiej przystosowane** do warunków środowiska.

Na świat przychodzi dużo więcej potomstwa, niż może pomieścić środowisko.

Przeżywają nieliczni, ale za to najlepsi (selekcja naturalna).

W procesie ewolucji istotne jest zachowywanie różnorodności cech.

Siła ewolucji to nie zaawansowany proces doskonalenia jednostki, lecz utrzymywanie dużej liczby różnorodnych osobników (tzw. populacji), która ewoluuje jako całość.

Algorytmy genetyczne

John Henry Holland (1929 - 2015) – amerykański naukowiec, profesor psychologii, elektrotechniki i informatyki na University of Michigan. **Pionier dziedziny zwanej obecnie algorytmami genetycznymi.**

- AG to algorytmiczna analogia naturalnych procesów ewolucji zachodzących w przyrodzie.
- Siłą napędową ewolucji jest maksymalne dopasowanie osobników do wymagań stawianych przez środowisko.
- Rolę środowiska w przypadku implementacji algorytmicznej spełnia tu funkcja jakości (funkcja celu, funkcja przystosowania).
- Pomimo elementu losowości AG nie błądzą przypadkowo, lecz wykorzystują efektywnie przeszłe doświadczenia.

Algorytmy genetyczne - pojęcia

chromosom - uporządkowany ciąg genów – pojedynczy osobnik;

gen – fragment chromosomu, który decyduje o dziedziczności jednej lub kilku cech

1	0	0	1	1	1	1	0	0	1
1	1	0	0	1	1	0	0	1	1
1	2	3	4	5	6	7	8	9	10

locus - miejsce genu w chromosomie;

allele - warianty (stany) jednego genu warunkujące daną cechę;

0	R1	?1
1	R2	?2

populacja (określona liczba chromosomów, osobników)

1	1	0	1	0	0	1	1	1	0
1	1	0	0	0	1	1	1	1	0
…	…	…	…	…	…	…	…	…	…
0	1	0	1	0	1	0	1	1	0

Algorytmy genetyczne - pojęcia

selekcja - wybór osobników, które zostaną poddane operacjom genetycznym dokonany na podstawie funkcji jakości.

(funkcji jakości)

1	1	1	0	1	0	0	1	1	1	0
2	1	1	0	0	0	1	1	1	1	0
3	1	1	1	0	0	1	1	0	0	0
…	…	…	…	…	…	…	…	…	…	…
i	0	1	0	1	0	1	0	1	1	0

krzyżowanie - operacja mająca na celu wymianę materiału genetycznego między osobnikami.

Losowy locus (zakres genów w chromosomie)

Losowy osobnik *n*	1	0	0	0	1	1	1	0	0	1
Losowy osobnik *m*	1	1	1	0	1	0	1	0	1	0

mutacja - zmiana jednego lub kilku genów w chromosomie (przeciwnu allel)

Losowy osobnik *n* przed mutacją	1	0	0	1	1	1	1	0	0	1
Losowy osobnik *n* po mutacji	1	0	0	0	1	1	1	0	0	1

Losowy locus

Algorytmy genetyczne - pojęcia

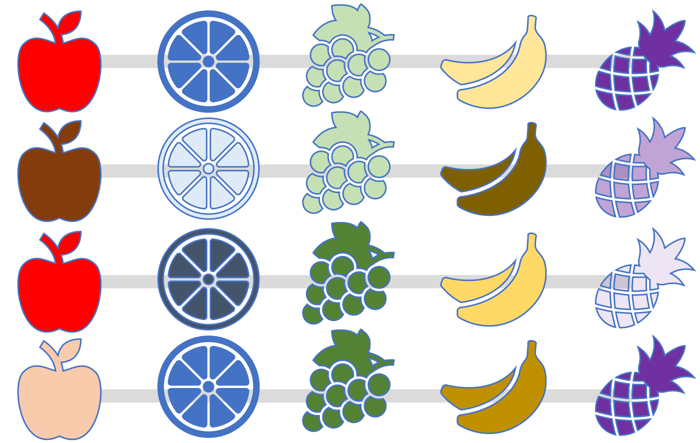
Reasumując – **osobnikiem**, bądź **chromosomem** nazywamy konkretną (jedną z wielu) koncepcję rozwiązania jakiegoś problemy ..
Czy to jest dobra koncepcja ?? – to oceni **funkcja jakości** ..

Jeśli jest niezła, być może pewien jej fragment jest doskonały – podczas **krzyżowania** liczymy na to, że złożenie fragmentów koncepcji rozwiązania problemu będzie skutkowało powstaniem jeszcze lepszego ..

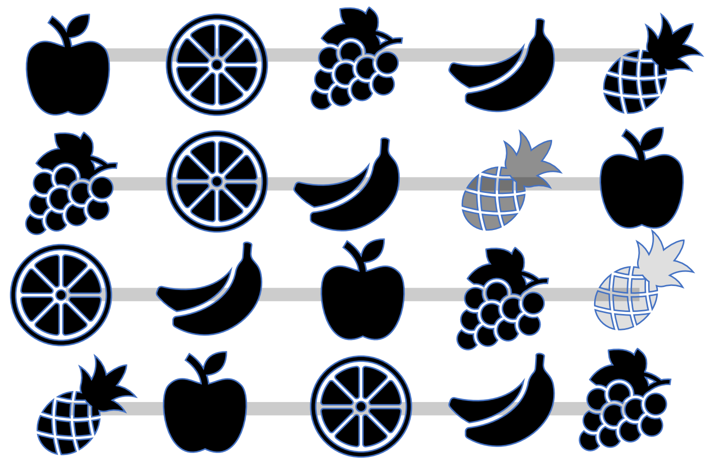
Mutacja wprowadza element losowy do potencjalnie dobrego rozwiązania – jest szansa że taki zabieg wybije nas z minimum lokalnego i pozwoli efektywniej kontynuować ewolucję ..

Algorytmy genetyczne - pojęcia

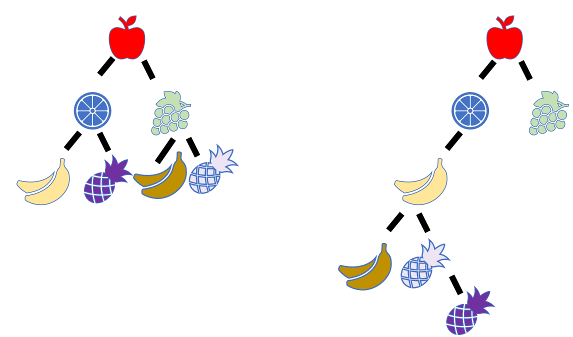
Geny w chromosomie mogą kodować rozwiązanie w sposób



Klasyczny - geny na różnych pozycjach przechowują różne informacje. W wyniku krzyżowania geny nie zmieniają pozycji, lecz wartości. Wykorzystywany w problemach, gdzie chcemy dobrać optymalne cechy osobnika.



Permutacyjny - geny przechowują podobne informacje. W wyniku krzyżowania nie zmieniają wartości, lecz miejsce w chromosomie. Wykorzystywany w problemach kombinatorycznych, np. problemie komiwojażera.



Drzewiasty - chromosom tworzy złożoną strukturę drzewiastą. W czasie krzyżowania przesunięciom ulegają całe gałęzie genów. Często geny mogą zmieniać także wartości. Wykorzystywany w programowaniu genetycznym oraz tam, gdzie ewolucji podlegają reguły matematyczne.

Algorytmy heurystyczne

- Algorytmy probabilistyczne
- **Algorytmy genetyczne**
- **Strategie ewolucyjne**
- Metody roju cząstek

Logika rozmyta

- Rozmyte systemy wnioskujące
- Sterowniki rozmyte

Sztuczne sieci neuronowe

Zagadnienia optymalizacji i aproksymacji

Systemy neuronowo-rozmyte

Filary inteligencji obliczeniowej

- Algorytmy probabilistyczne
- Obliczenia ewolucyjne
- Logika rozmyta
- Sztuczne sieci neuronowe

Klasyczny algorytm genetyczny pracuje na binarnych chromosomach (alg. ewolucyjny dopuszcza liczby zmiennoprzecinkowe), pełna reprezentacja populacji w trakcie reprodukcji (nawet najgorsze osobniki mają szansę), najpierw selekcja, potem rekombinacja)

Np. Analiza wrażliwości cech sieci neuronowej (binarna reprezentacja)

Kodowanie jest bardzo istotnym etapem projektowania algorytmu. Sposób zakodowania w chromosomie informacji o proponowanym rozwiązaniu wydatnie wpływa na szybkość i jakość znajdowanych wyników. Przyczyną takiego zjawiska jest wpływ kodowania na sposób w jaki przeszukiwana jest przestrzeń rozwiązań. Złe kodowanie może spowodować, że nigdy nie zostanie przeszukany fragment przestrzeni, w którym znajdują się najlepsze rozwiązania!

Algorytmy genetyczne – schemat działania

begin

t:=0

losujemy populację początkową P(t)

oceniamy P(t)

while (not *warunek_zakończenia*) do

begin

t:=t+1

wybieramy P(t) z P(t-1) (selekcja)

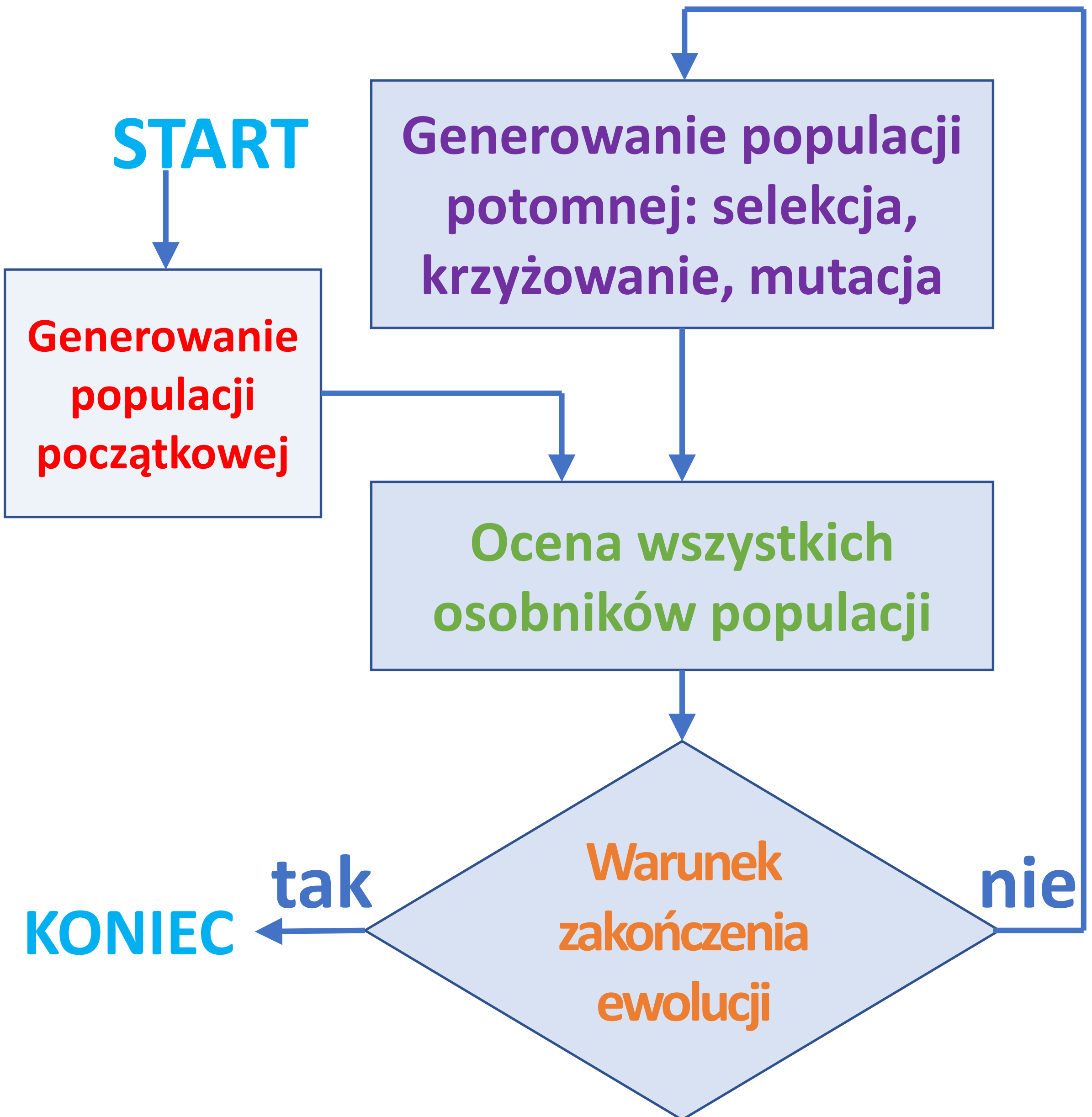
ewolucja P(t)

(operatory genetyczne)

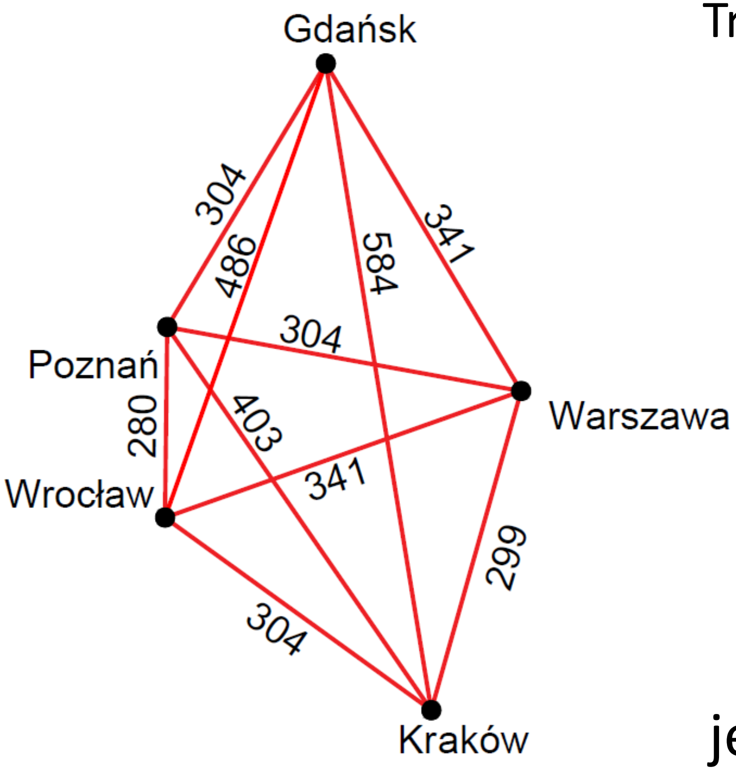
oceniamy P(t)

end

end



Czy dla niektórych problemów heurystyki to jedyny wybór ? – problem KOMIWOJAŻERA



Trasa komiwojażera jest cyklem przechodzącym przez każdy wierzchołek grafu dokładnie jeden raz – „cykl Hamiltona”.
Nie jest znany działający w czasie co najwyżej wielomianowym algorytm rozwiązujący problem. Problem jest NP-trudny. Złożoność czasowa *O(n!)*.

Najszybszym superkomputerem w Polsce jest Prometheus na AGH - niemal 2.4 PFLOPS (10^{15} floating point operations per second).

n	5	10	15	20	30	50	100
n!	120	3.6E+06	1.3E+12	2.4E+18	2.7E+32	3.0E+64	9.3E+157
n!/2.4E15	5.0E-14	1.5E-09	5.4E-04	1.0E+03	1.1E+17	1.3E+49	3.9E+142

Wiek Wszechświata określa się na 10E+18 sekund – tyle czasu upłynęło od Wielkiego Wybuchu.

Klasyczny przykład – problem KOMIWOJAŻERA

chromosom – wektor zawierający n liczb naturalnych – liczby to numery miast które chcemy odwiedzić

Kodowanie chromosomu

7	**9**	**12**	**4**	**10**	**1**	**8**	**6**	**…**	**3**
1	2	3	4	5	6	7	8	…	*n*

locus - miejsce genu w chromosomie – odpowiada także kolejności w jakiej odwiedzimy miasta
(miast nr 7 jako 1wsze,
miast nr 9 jako 2gie,
ostatnie miasto nr 3 jako *n*-te)

Klasyczny przykład – problem KOMIWOJAŻERA

Funkcja jakości - ∑x n → m

Odległość x (kilometr)	m / n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Warszawa	1		294	172	295	105	176	318	336	278	110	437	465	215	289	516
Kraków	2	295		309	80	201	121	169	127	632	366	336	543	379	85	691
Lublin	3	173	310		351	114	191	164	403	448	281	612	639	389	356	691
Katowice	4	294	79	350		242	161	248	51	571	305	260	467	317	7	615
Radom	5	105	197	113	239		78	192	291	382	183	489	516	267	244	568
Kielce	6	175	120	189	162	81		158	214	452	254	423	513	263	167	565
Rzeszów	7	318	168	170	248	192	157		294	594	396	503	710	546	253	858
Rybnik	8	335	126	407	51	299	219	295		612	346	254	462	358	49	610
Elbląg	9	281	577	450	571	387	458	600	612		247	687	342	272	565	247
Płock	10	108	366	277	305	182	253	396	345	244		421	381	50	298	368
Wałbrzych	11	437	335	605	260	488	427	504	254	688	422		275	434	260	530
Gorzów Wielk.	12	464	542	632	466	516	515	710	461	378	381	274		350	467	257
Włocławek	13	208	372	377	310	260	259	541	351	272	51	426	351		304	319
Chorzów	14	287	85	356	8	248	168	254	50	564	298	260	467	310		615
Koszalin	15	518	798	686	723	569	568	967	718	246	370	531	259	320	723	

Klasyczny przykład – problem KOMIWOJAŻERA

Warunek zakończenia ewolucji

Osiągnięcie konkretnej wartość funkcji jakości – *wymaga wiedzy o spodziewanym optimum …*

Poprawa funkcji jakości względem wartości startowej o zadaną wartość % - *możemy przerwać dobrze zapowiadającą się ewolucję …*

Określona ilość pokoleń bez poprawy funkcji jakości

Klasyczny przykład – problem KOMIWOJAŻERA

begin

    t:=0
    losujemy populację początkową P(t)
    oceniamy P(t)
    while (not *warunek_zakończenia*) do
    begin
        t:=t+1
        wybieramy P(t) z P(t-1) (selekcja)
        ewolucja P(t)
        (operatory genetyczne)
        oceniamy P(t)
    end

end

Klasyczny przykład – problem KOMIWOJAŻERA

Losujemy populację początkową

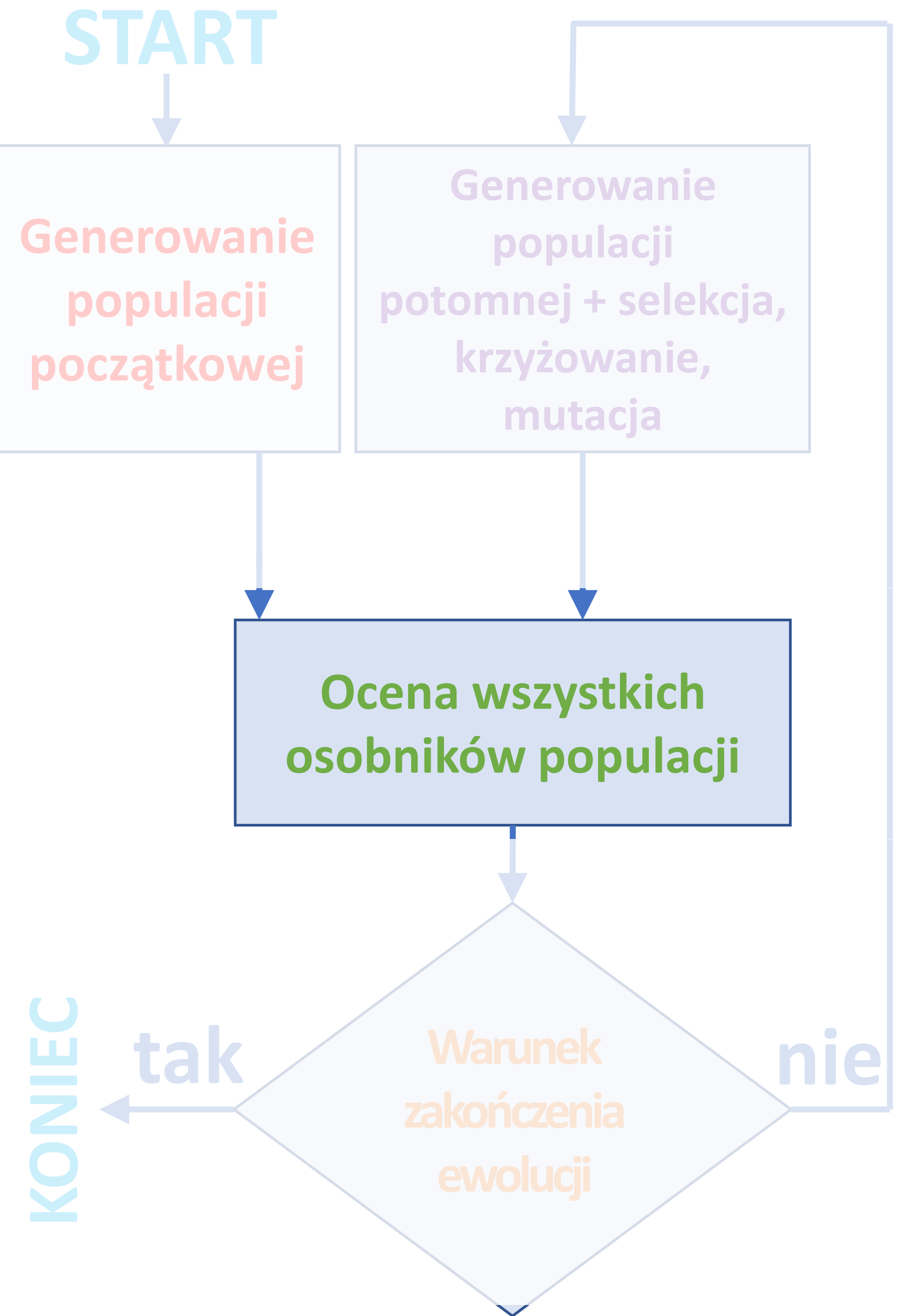
1	7	9	12	4	10	1	8	6	…	3
2	2	10	3	6	5	9	7	6	…	4
3	7	4	6	7	5	3	2	1	…	8
…	…	…	…	…	…	…	…	…	…	…
m	10	8	9	5	4	6	2	3	…	1
	1	2	3	4	5	6	7	8	…	*n*

Liczebność populacji

Długość chromosomu – ilość miast do odwiedzenia

Klasyczny przykład – problem KOMIWOJAŻERA

```
begin
    t:=0
    losujemy populację początkową P(t)
    oceniamy P(t)
    while (not warunek_zakończenia) do
    begin
        t:=t+1
        wybieramy P(t) z P(t-1) (selekcja)
        ewolucja P(t)
        (operatory genetyczne)
        oceniamy P(t)
    end
end
```



Klasyczny przykład – problem KOMIWOJAŻERA

kolejność przystosowania osobników

oceniamy populację P(n) – funkcja jakości [km]

(minimalizujemy funkcję jakości)

1	7	9	12	4	10	1	8	6	…	3	33 548	4
2	2	10	3	6	5	9	7	6	…	4	31 445	1
3	7	4	6	7	5	3	2	1	…	8	38 449	m
…	…	…	…	…	…	…	…	…	…	…	…	…
m	10	8	9	5	4	6	2	3	…	1	32 459	3
	1	2	3	4	5	6	7	8	…	*N*		

Klasyczny przykład – problem KOMIWOJAŻERA

begin

t:=0

losujemy populację początkową P(t)

oceniamy P(t)

while (not *warunek_zakończenia*) do

begin

t:=t+1

wybieramy P(t) z P(t-1) (selekcja)

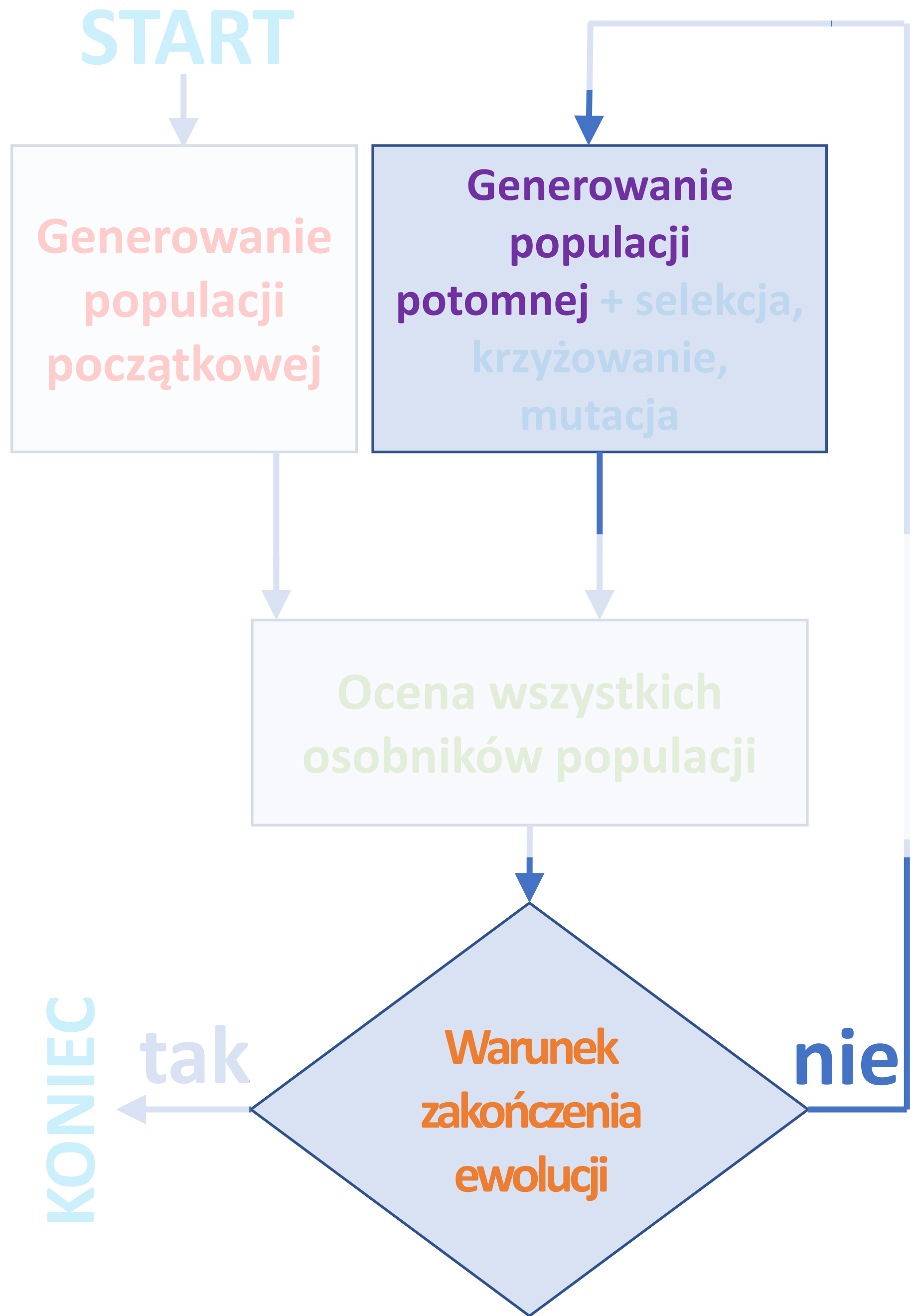
ewolucja P(t)

(operatory genetyczne)

oceniamy P(t)

end

end



Klasyczny przykład – problem KOMIWOJAŻERA

Lista rankingowa – odcięcie – „twarda ewolucja”

32	7	9	12	4	10	1	8	…	3	30 548	1
4	2	10	3	6	5	9	7	…	4	31 445	2
11	7	4	6	7	5	3	2	…	8	32 449	3
…	…	…	…	…	…	…	…	…	…	…	…
35	2	4	6	7	5	3	9	…	8	35 332	25
4	3	4	6	7	5	7	2	…	8	35 871	26
…	…	…	…	…	…	…	…	…	…	…	…
N	10	8	9	5	4	6	2	…	1	52 459	m

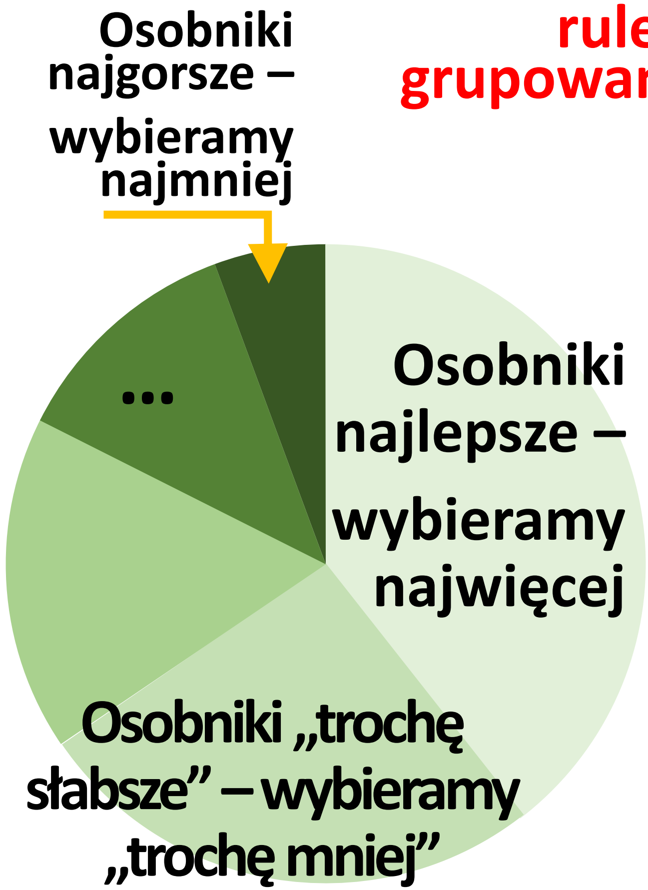
…wybieramy pewną określoną liczbę najlepszych osobników, którzy wezmą udział w rozmnażaniu…

Osobniki, które nie zostały wybrane, są usuwane z populacji…

Metoda prosta – lista rankingowa
Metoda turniejowa – losujemy określoną ilość par chromosomów, pomiędzy chromosomami realizujemy pojedynki

Klasyczny przykład – problem KOMIWOJAŻERA

Metoda ruletki z grupowaniem



Metoda ruletki bezpośrednia

Przykładowy sposób wyznaczenia prawdopodobieństwa *P* reprodukcji osobnika *k* reprezentującego rozwiązanie o jakości *r(k)* przy bieżącym optymalnym rozwiązaniu r_{opt}

$$P(k) = a + b \cdot \left(\frac{r(k)}{r_{opt}}\right)$$

a, b: współczynniki kontrolne,

$$\sum_{i=0}^{N} P(k_i) = 1 \;;\; 0 \le P(k) \le 1 \;;\; r(k) \ge r(l) \Rightarrow p(k) \ge r(l)$$

Klasyczny przykład – problem KOMIWOJAŻERA

Po wykonaniu operacji krzyżowania oraz mutacji należy przeprowadzić proces sukcesji.

sukcesja z całkowitym zastępowaniem

nową populacją bazową staje się populacja potomna.

żaden osobnik z poprzedniej populacji nie zostaje przeniesiony do nowej.

najwolniej prowadzi do optymalnego rozwiązania,

jest najbardziej odporna na tendencję osiągania ekstremów lokalnych

sukcesja z częściowym zastępowaniem

w nowej populacji znajdują się osobniki z poprzedniej i potomnej populacji

metoda ta prowadzi zwykle do stabilniejszej pracy algorytmu ewolucyjnego,

może spowodować tendencję do osiągania ekstremów lokalnych

sukcesja elitarna

w nowej populacji znajduje się co najmniej jeden najlepszy osobnik z poprzedniej populacji

może przyspieszyć znalezienie optymalnego rozwiązania

zwiększa prawdopodobieństwo osiągania ekstremów lokalnych

*odporność na utykanie w minimach lokalnych

Klasyczny przykład – problem KOMIWOJAŻERA

begin

t:=0

losujemy populację początkową P(t)

oceniamy P(t)

while (not *warunek_zakończenia*) do

begin

t:=t+1

wybieramy P(t) z P(t-1) (selekcja)

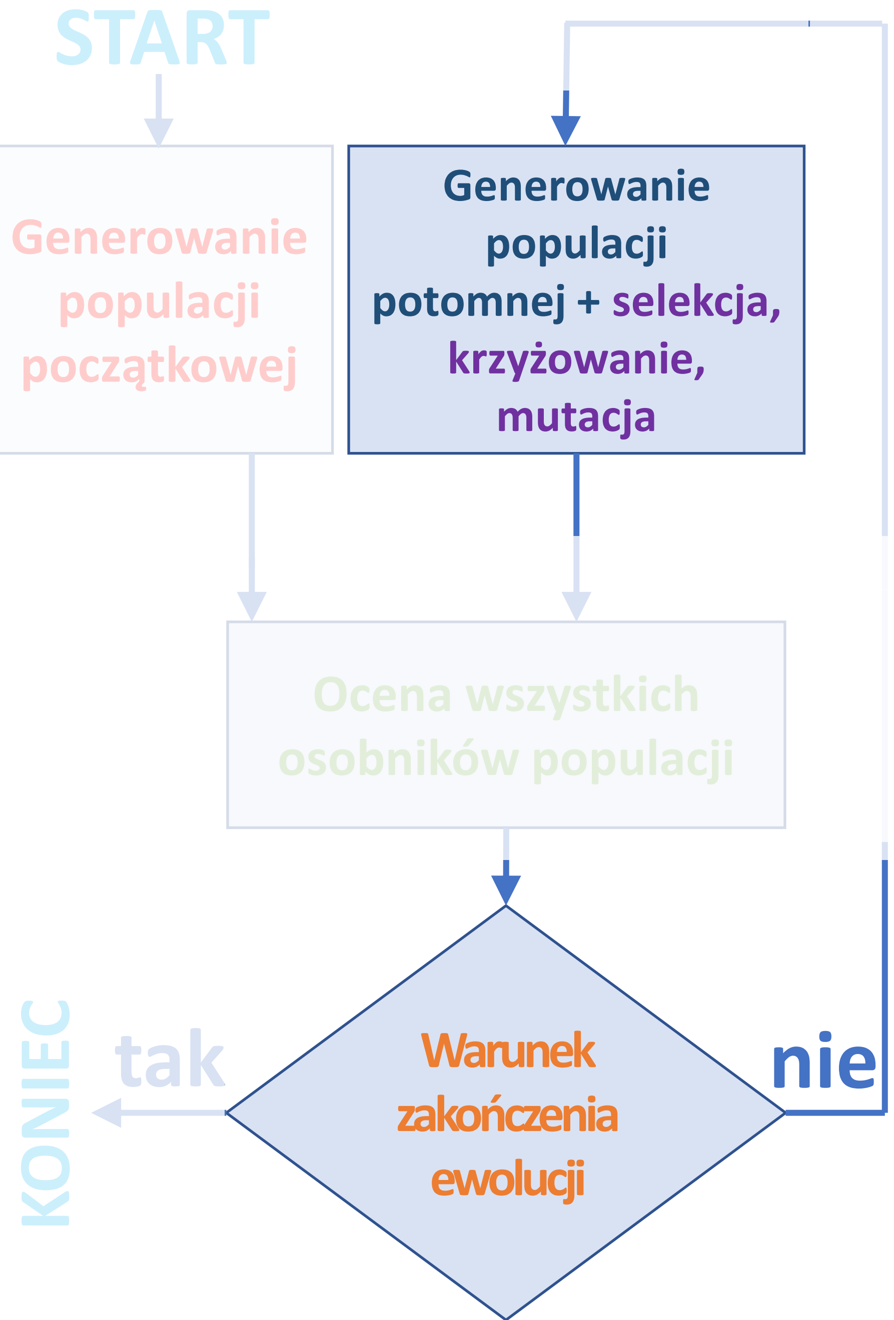
ewolucja P(t)

(operatory genetyczne)

oceniamy P(t)

end

end



Klasyczny przykład – problem KOMIWOJAŻERA

KRZYŻOWANIE

losowy Locus początku

losowy Ilość genów

losowy Locus końca

losowy osobnik rodzicielski *n*	10	7	4	2	1	9	8	6	5	3
losowy osobnik rodzicielski *m*	6	5	3	2	4	7	10	8	1	9

osobnik potomny *n*	10	7	3	2	4	9	8	6	5	3
osobnik potomny *m*	6	5	4	2	1	7	10	8	1	9

Problem – każde z miast możemy odwiedzić tylko 1 raz !

legalizacja

2 → 2 nieistotne

3 → 4

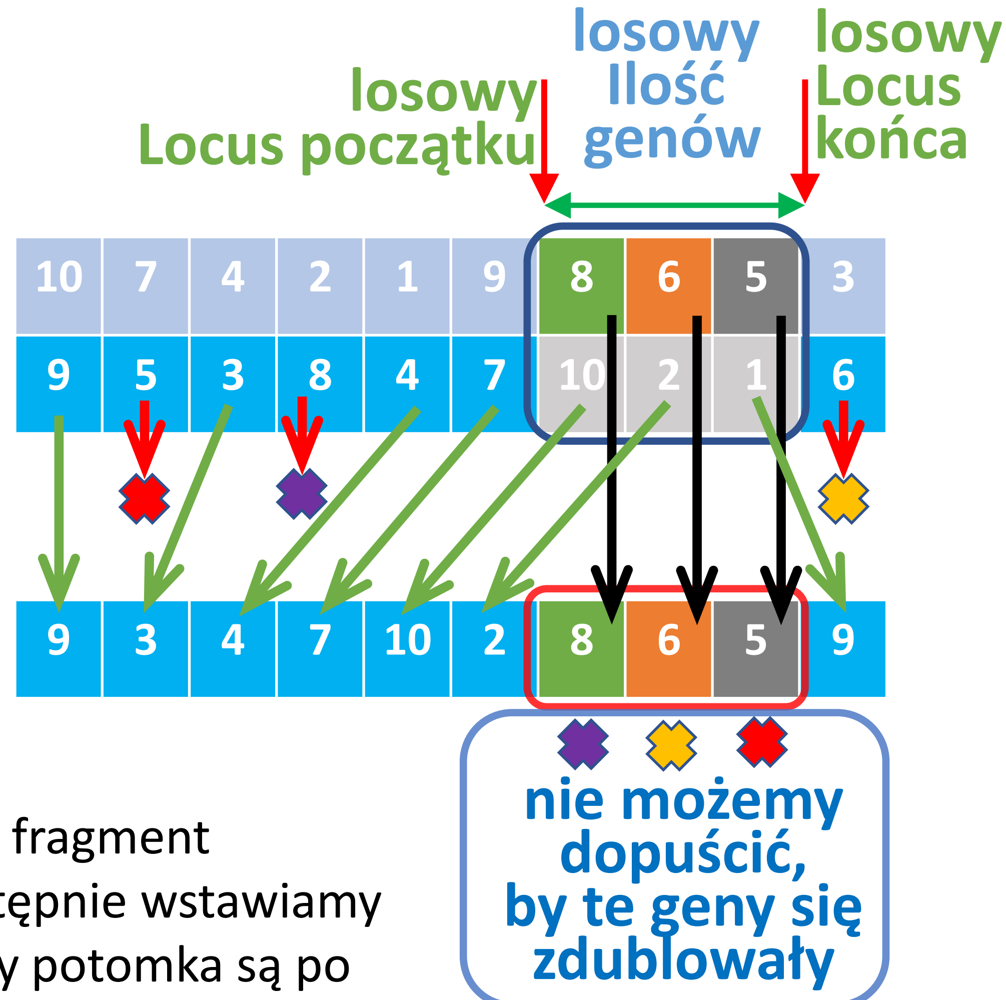
4 → 1

3 → 4 i 4 → 1 wymiana „przechodnia” – 4ki są ok

osobnik potomny *n+1*	10	7	3	2	4	9	8	6	5	1
osobnik potomny *m+1*	6	5	4	2	1	7	10	8	3	9

Klasyczny przykład – problem KOMIWOJAŻERA

KRZYŻOWANIE –
prostsza koncepcja
(tzw. uporządkowane crossover)



W tej metodzie krzyżowania wybieramy fragment chromosomu pierwszego rodzica, a następnie wstawiamy go do potomka. Wszelkie brakujące geny potomka są po kolei dodawane od drugiego rodzica (w kolejności, w jakiej występowały), za wyjątkiem genów które dublują się z pochodzącymi od pierwszego rodzica.

Klasyczny przykład – problem KOMIWOJAŻERA

MUTACJA

losujemy 2 Locusy w obrębie jednego chromosomu

Ver.1

losowy osobnik *n*

10	7	4	2	1	9	8	6	5	3

losowy osobnik *n* po mutacji

10	7	9	2	1	4	8	6	5	3

zamieniamy allele na losowych locusach

allel - wariant (stan) jednego genu

Przy tych operacjach nie ma możliwości wygenerowania zdublowanego genu w chromosomie

Ver.2

losowy osobnik *n*

10	7	4	2	1	9	8	6	5	3

losowy osobnik *n* po mutacji

10	7	2	1	9	4	8	6	5	3

przesuwamy grupę alleli

Duży udział mutacji jest pożądany we wczesnych fazach pracy algorytmu (poszukiwanie minimów lokalnych).

Mniejszy udział mutacji jest korzystny w późnych fazach pracy algorytmu (lokalizacja rozwiązania wewnątrz obszaru przyciągania minimum).

..na podstawie wskazówek z literatury, prawdopodobieństwo mutacji $Pm \approx 1/L$, gdzie L to długość chromosomu).

Klasyczny przykład – problem KOMIWOJAŻERA

begin

    t:=0

    losujemy populację początkową P(t)

    oceniamy P(t)

    while (not *warunek_zakończenia*) do

    begin

        t:=t+1

        wybieramy P(t) z P(t-1) (selekcja)

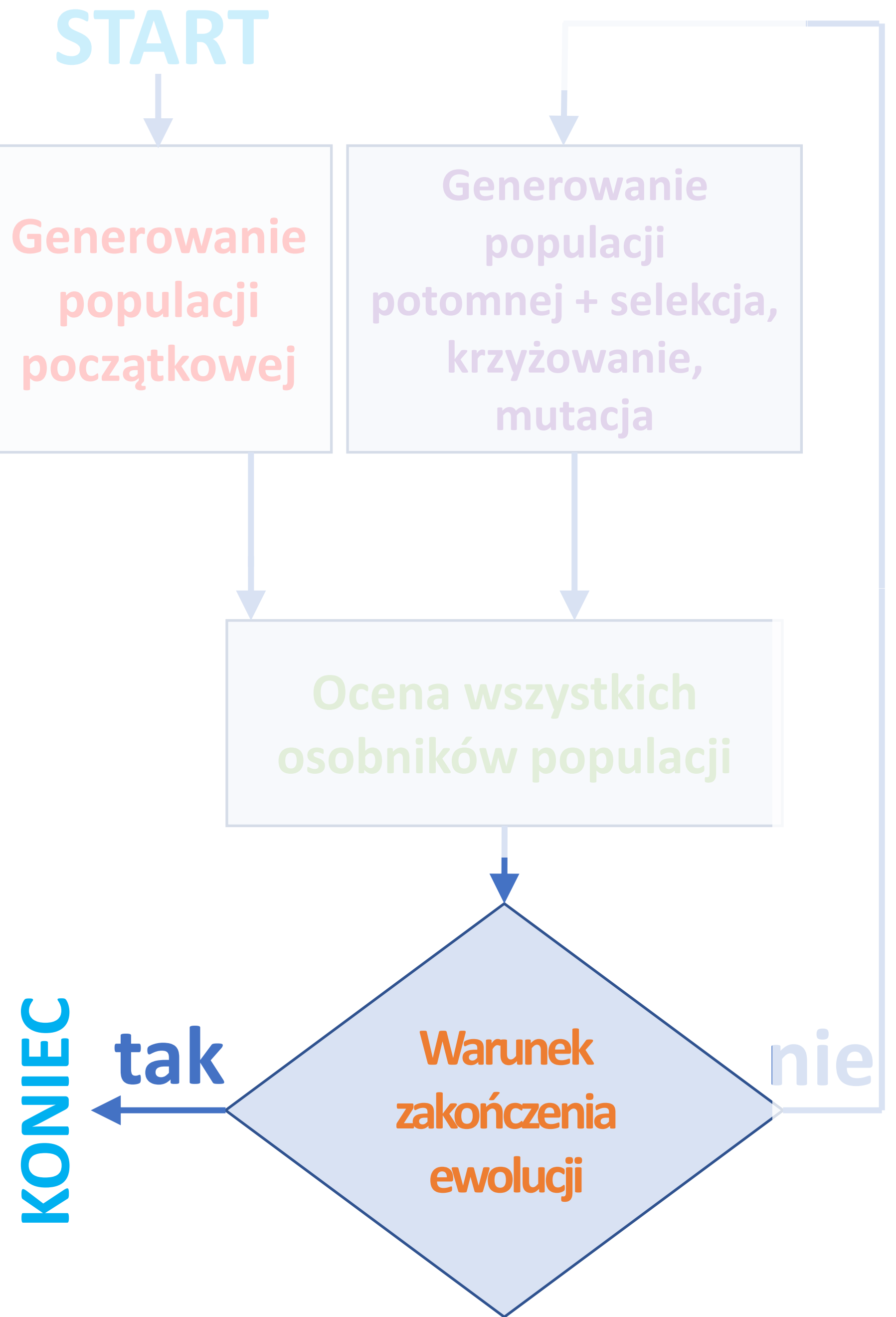
        ewolucja P(t)

        (operatory genetyczne)

        oceniamy P(t)

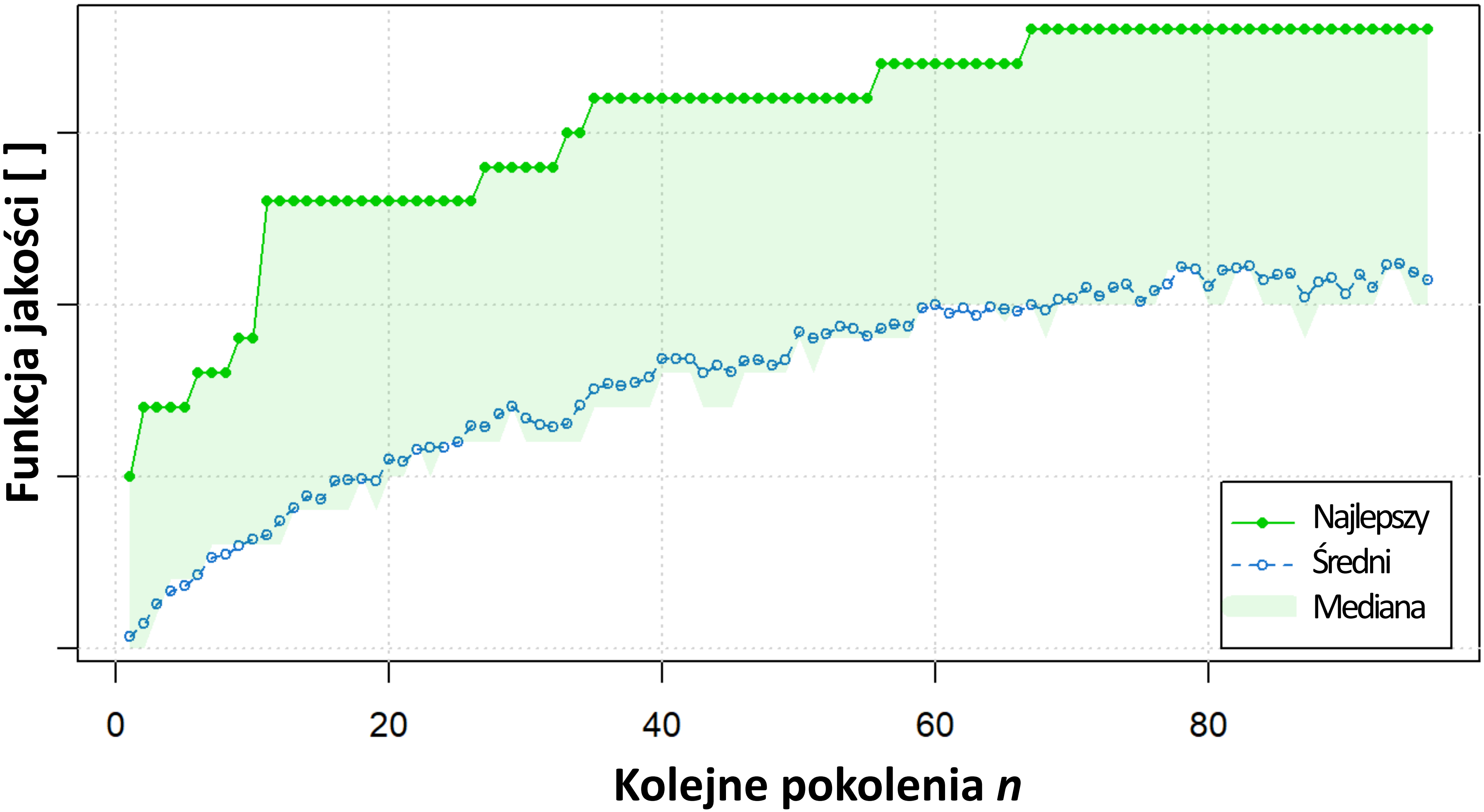
    end

end

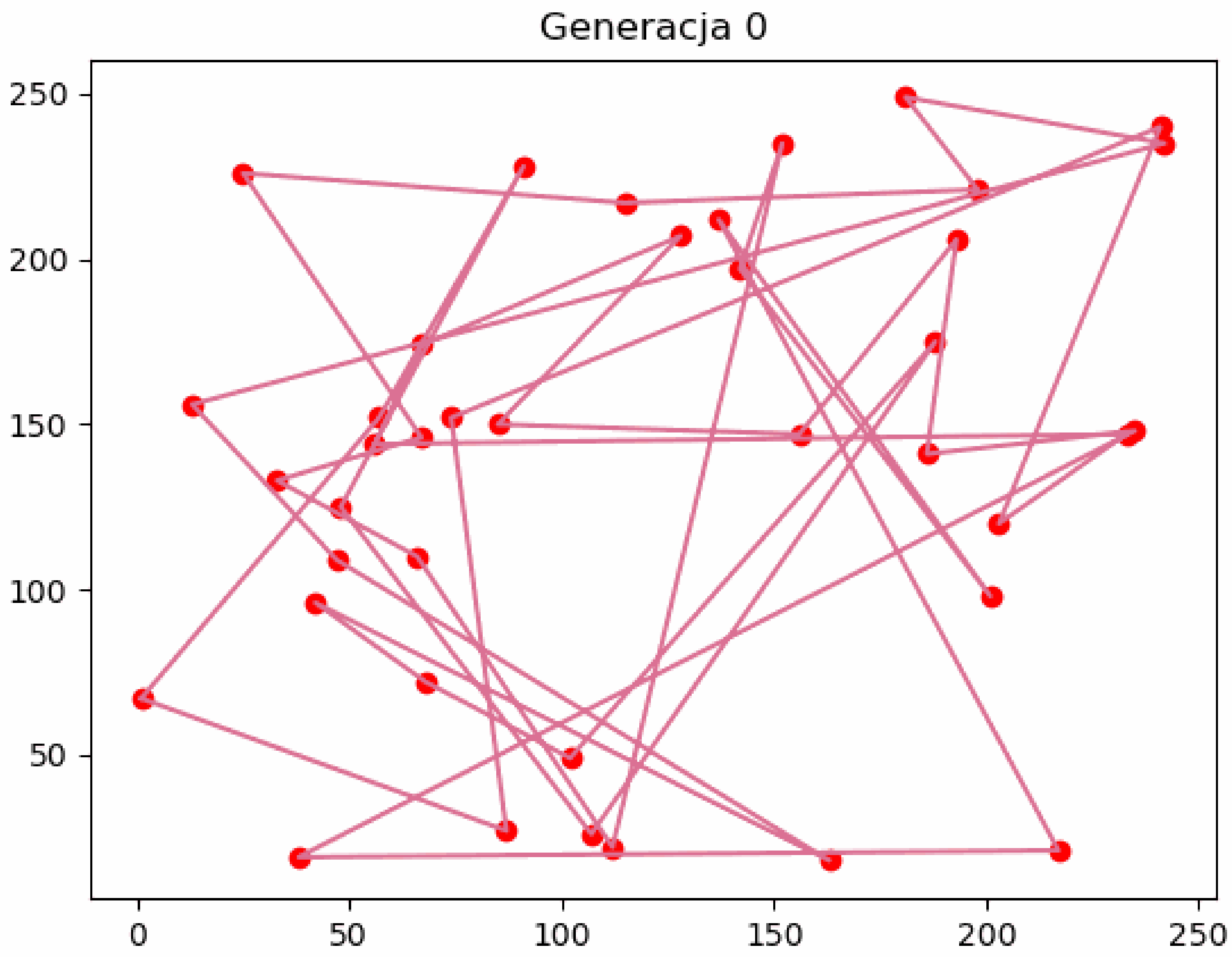


Klasyczny przykład – problem KOMIWOJAŻERA

efekty ewolucji



Klasyczny przykład – problem KOMIWOJAŻERA



Algorytmy genetyczne - najważniejsze

- **Zastanówmy się czy:**

  - **Czy jest to problem natury optymalizacyjnej – czy poszukujemy optymalnego rozwiązania spośród wielu dostępnych ?**

  - **Czy nie ma prostych rozwiązań problemu ? Algorytm zachłanny? Jaka jest złożoność obliczeniowa zagadnienia?**

  - **Istnieje funkcja opisująca jakość rozważanego rozwiązania (osobnika), która efektownie nada kierunek ewolucji**

  - **Istnieje możliwość zakodowania potencjalnego rozwiązania problemu (osobnika) w sensownej postaci (chromosom)**

Algorytmy genetyczne - zastosowanie

- wyznaczanie topologii układów elektronicznych;
- harmonogramowanie – szeregowanie zadań;
- sterowanie adaptacyjne; sterowanie optymalne;
- rozgrywanie gier;
- zadanie komiwojażera; zadanie plecakowe;
- **PRZYKŁAD**
- nieliniowe systemy dynamiczne – analiza danych;
- przewidywanie;
- projektowanie sieci neuronowych: architektury i wagi;
- poruszanie robotem;
- tworzenie programów;
- planowanie;
- znajdowanie kształtu molekuł białek;
- tworzenie grafik i muzyki;

Marzec 2025

Na kiedy ??

Pn	Wt	Śr	Cz	Pt	So	N
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26 !	27 !	28 !	29	30
31						

Kolejne obligatoryjne początkiem maja..

TERMINY ODDANIA PROJEKTÓW

6 kwietnia/ montecarlo

11 maja/ genetyczny

Wszelkie problemy z genetycznym – piszcie albo przyjdźcie na nieobligatoryjne, albo do mnie (p.317)

Dzisiejszy wykład

Najbliższe obligatoryjne ćwiczenia – juro i pojutrze:
2024-03-27 i 2024-03-28
Na tych opiszemy
drugi projekt ...

!

Kwiecień 2025

Pn	Wt	Śr	Cz	Pt	So	N
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Program staży letnich w Pega:

PEGA Platform Solutions Customers Learn Services & Partners Events About

Software Engineer Summer Intern - Microservices

Job Category: Internships

Location: Poland - Krakow

SHARE THIS PAGE

Apply Now

Already have an

Enter first name

Enter last name

Enter phone (re

Enter email (req

Enter city (requi

Select country o

Enter postal coc

How did you lea

About The Summer Internship Program at Pega:

Join an award-winning Internship Program at Pega Poland! Our program starts on July 1st and runs until September 26th. As an intern, you'll actively participate in your team's daily work while also enjoying additional benefits such as:

- Coffee Break Series with Pega's leadership team — ask questions and learn from their experience!
- Diversity & Inclusion (D&I) and Employee Resource Group workshops.
- Volunteer activities.

...and much more!

Meet Our Team:

We are building the next generation of Voice AI that transforms words into actions, driving exceptional customer experiences. At Pega, we're bringing cutting-edge customer service solutions to market, equipped with capabilities that make life easier for both customers and customer service agents. Our team is made up of highly technical software engineers and machine learning scientists.

Program staży letnich w Pega:

Dołącz do nagradzanego programu stażowego w Pega!

Nasz program rozpoczyna się 1 lipca i potrwa do 26 września.

https://www.pega.com/about/careers/21089/software-engineer-summer-intern-microservices

Pod linkiem na dole slajdu znajduje się zestawienie z datami oddania kolejnych projektów i .. ogólnie bieżącą sytuacją – proszę sprawdzajcie, czy się nie pomyliłem – jeśli się pomyliłem, piszcie do mnie …

Nazwisko	Imię	Monte	Genetycz	Fuzzy	Neuro	Ile oddanych	Ocena z projektów	Ocena z egzaminu	Ocena Końcowa
A	Łucja Weronika	3/21/2025				1	2		
B	Jakub Marcel	3/24/2025				1	2		
B	Maja	3/21/2025				1	2		
Ch	Jakub Hubert					0	2		
Ch	Wiktor Jan					0	2		
D	Konrad Adam					0	2		
D	Kamil Stanisław	3/19/2025				1	2		
D	Adam					0	2		
D	Tomasz Piotr					0	2		
D	Gabriela					0	2		
D	Jan Bartosz	3/21/2025				1	2		
E	Maurycy					0	2		
F	Julia					0	2		
F	Aleksandra Maria					0	2		
G	Zuzanna Ewa					0	2		
G	Patrycja	3/19/2025				1	2		
G	Mikołaj					0	2		
G	Konrad Wojciech					0	2		
G	Jakub	3/19/2025				1	2		
H	Karol					0	2		
H	Aleksander Jakub	3/21/2025				1	2		
I	Bartosz					0	2		
J	Filip Andrzej					0	2		
J	Aleksandra					0	2		
J	Julia Weronika					0	2		
J	Roksana Kamila	3/24/2025				1	2		
J	Weronika					0	2		
Ki	Maria					0	2		
Kn	Maria	3/17/2025				1	2		
K	Karolina					0	2		
K	Oliwier Piotr	3/15/2025				1	2		
K	Julia Anita					0	2		
K	Bartłomiej Mariusz	3/23/2025				1	2		
K	Dawid Tomasz	3/20/2025				1	2		
K	Natalia Katarzyna	3/21/2025				1	2		
K	Kacper Bartłomiej					0	2		
L	Patryk					0	2		
M	Patrycja Anna	3/21/2025				1	2		
M	Gerard	3/20/2025				1	2		
M	Jakub Franciszek	3/16/2025				1	2		
M	Eliza Klaudia					0	2		
M	Karolina					0	2		
O	Joanna Julia					0	2		
P	Miłosz	3/25/2025				1	2		
P	Szymon Mateusz					0	2		
Pie	Bartosz					0	2		
Piw	Bartłomiej Jakub	3/24/2025				1	2		
P	Magdalena Maria					0	2		
P	Dominika					0	2		
R	Gabriel					0	2		
S	Dominik					0	2		
S	Marcin Jan	3/17/2025				1	2		
S	Julia Krystyna	3/25/2025				1	2		
S	Julita					0	2		
S	Szymon Marcin	3/16/2025				1	2		
S	Weronika					0	2		
S	Martyna Joanna					0	2		
S	Wiktoria Danuta					0	2		
S	Aleksandra					0	2		
S	Anna Sara					0	2		
Szcz	Magdalena Anna					0	2		
Szt	Magdalena	3/19/2025				1	2		
Sz	Mikołaj					0	2		
T	Katarzyna					0	2		
T	Witold	3/17/2025				1	2		
W	Julia Barbara					0	2		
W	Katarzyna					0	2		
W	Oliwia Klaudia	3/25/2025				1	2		
Z	Michał	3/21/2025				1	2		
Z	Karolina					0	2		
	ile % ? / średnio	37.7	0.0	0.0	0.0	0.371	2.0		

https://docs.google.com/spreadsheets/d/1LYZjd4Fg9ql7hI4Qhglm06P6IkBDF_mb/edit?usp=sharing&ouid=102885865370465498299&rtpof=true&sd=true

Tematy inżynierek!!
do końca marca ..

Do zobaczenia
jutro i pojutrze !