



ALGORYTMY

GENETYCZNE

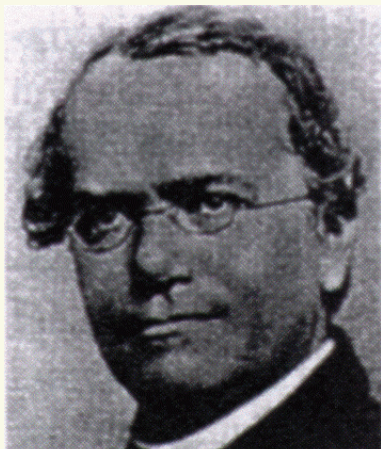


Algorytmy Genetyczne

- I. Co to są algorytmy genetyczne?
- II. Podstawowe pojęcia algorytmów genetycznych
- III. Proste algorytmy genetyczne
- IV. Kodowanie osobników i operacje genetyczne.

- V. Implementacja binarna
- VI. Kiedy zatrzymać algorytm genetyczny?
- VII. Symulacja działania algorytmu genetycznego
- VIII. Warunki eksperymentu
- IX. Zastosowanie

Syntetyczna teoria ewolucji



Gregor Mendel (1822-1884)

GENETYKA

Zasady
dziedziczenia



Charles Darwin (1809-1882)

EWOLUCJA

Zasady doboru
naturalnego

NEODARWINIZM

DNA

**SYNTETYCZNA
TEORIA EWOLUCJI**

Historia



Pojęcie „**genu**” wprowadzono w **1910** roku na określenie abstrakcyjnej jednostki dziedziczenia, odpowiedzialnej za jakąś cechę danego gatunku.

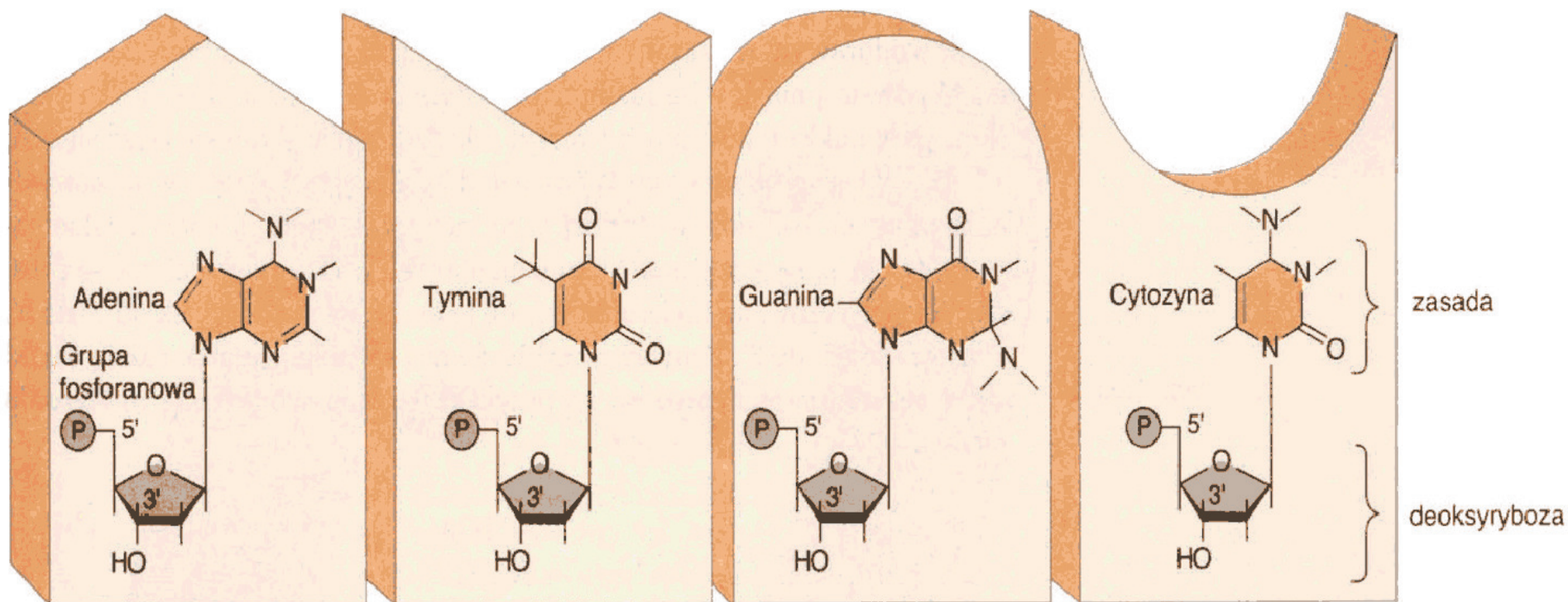
W **1869r.** J. F. **Miescher** wyizolował z jąder komórkowych kwas deoksyrybonuleinowy - **DNA**.

W **1953** roku na podstawie dostępnych fizycznych i chemicznych informacji J. **Watson** i F. **Crick** ustalili, że **DNA** ma strukturę podwójnej **helisy**.

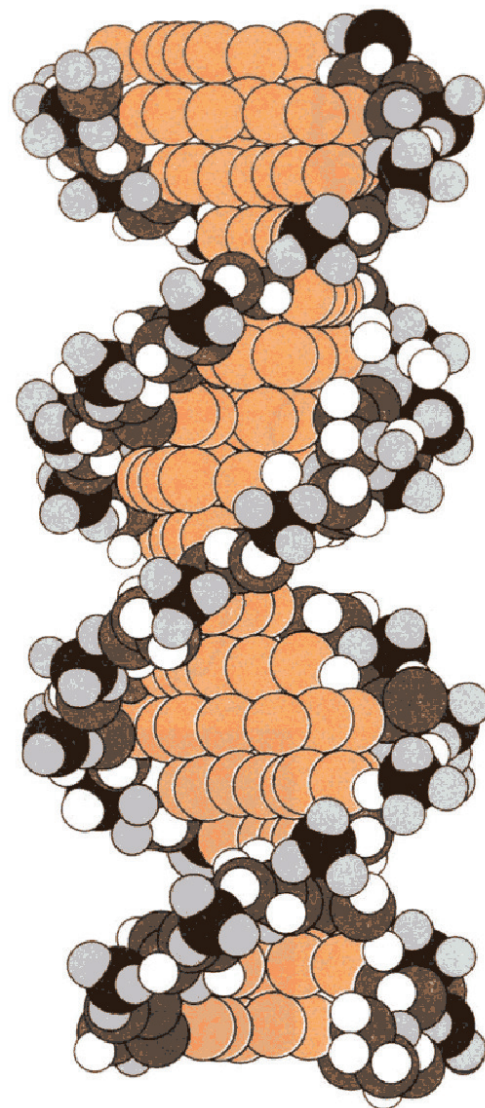
DNA jest **polimerem** składającym się z czterech cząsteczek zwanych nukleotydami: **adenina** (A), **tymina** (T), **guanina** (G), **cytozyna** (C).

Informacja genetyczna zapisywana jest w **DNA** za pomocą kolejności ułożenia cząstek.

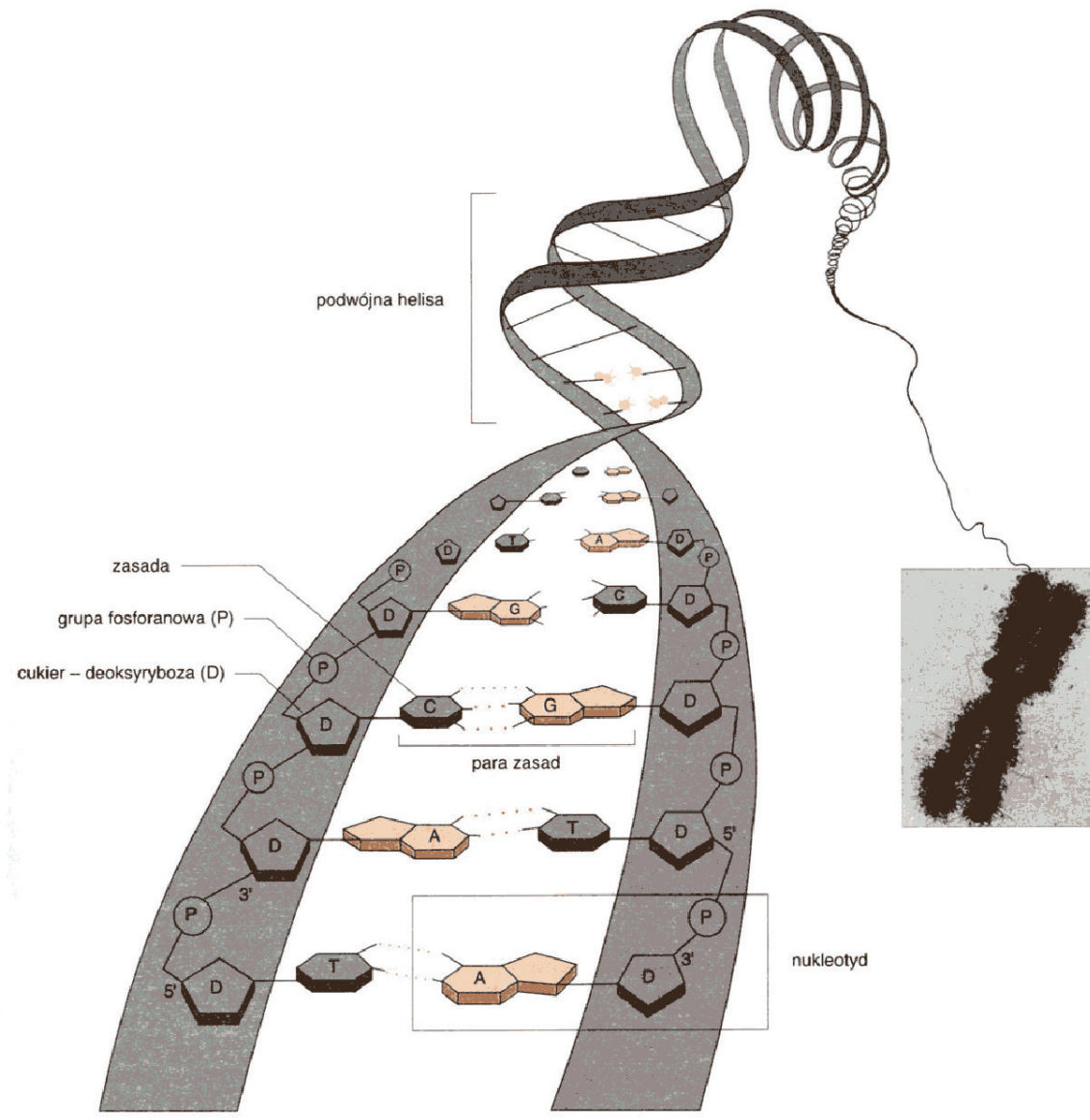
Podstawowe jednostki strukturalne DNA



Model podwójnej helisy DNA



Ułożenie i powiązanie nukleotydów w helisie DNA



Ludzki chromosom

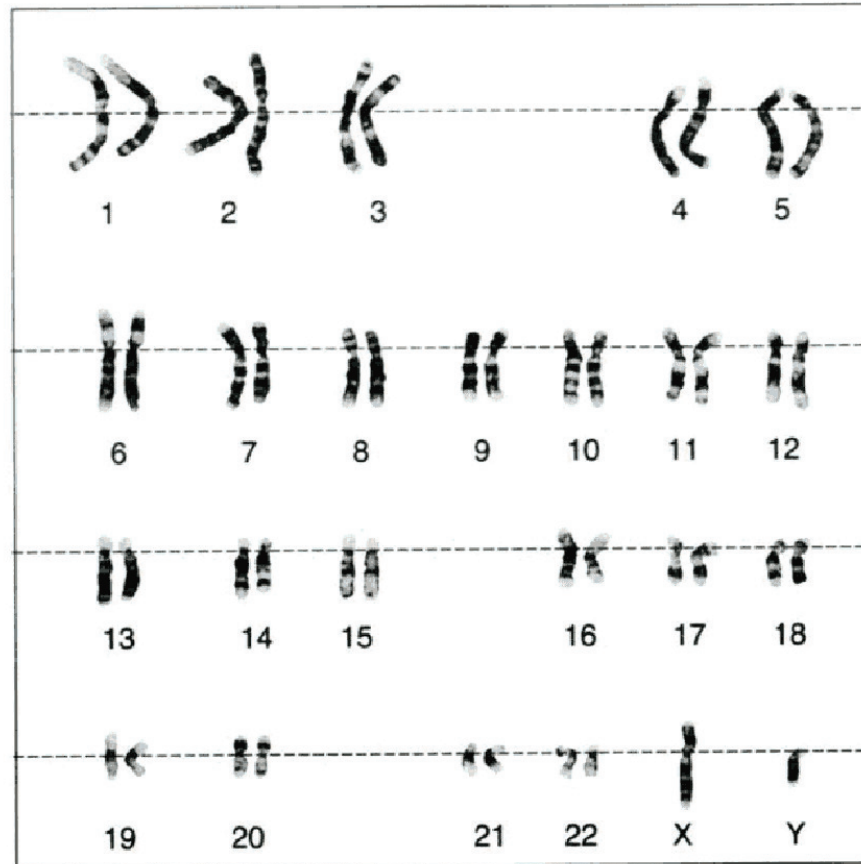


powiększenie
30 000 razy

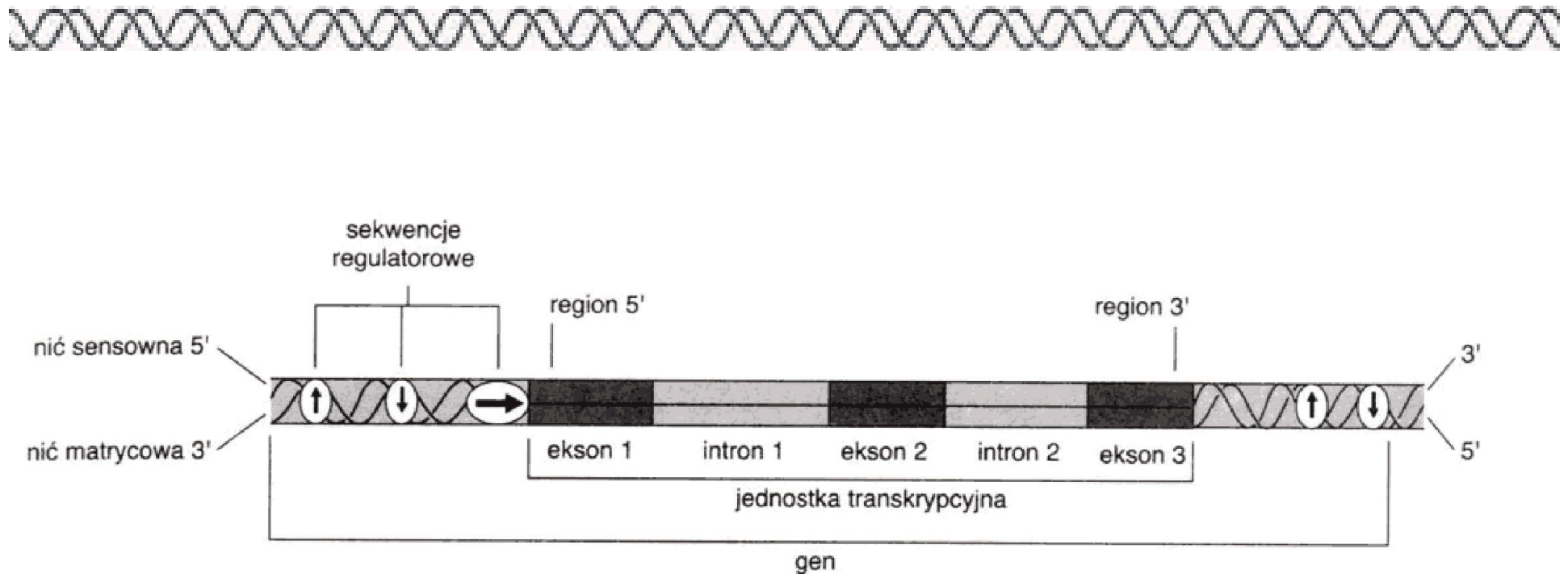


Rysunek schematyczny

Chromosomy jednego gatunku



Elementy DNA wchodzące w skład genu



Typowy gen składa się z krótkich odcinków kodujących, czyli eksonów przegradzanych odcinkami niekodującymi, zwanymi intronami.

Teoria ewolucji



Teoria ewolucji jest podstawą teoretyczną współczesnej biologii, dostarcza ona odpowiedzi na pytania: jak powstały przeróżne formy **organizmów**, jak powstały różne rodzaje organizmów.

Wyjaśnienia tych zagadnień podał **Karol Darwin** formułując teorię ewolucji, według niej głównym procesem kształtującym właściwości organizmów jest **dobór naturalny**, ważnym zagadnieniem teorii jest teza, że wszystkie współcześnie żyjące organizmy połączone są więzami wspólnego pochodzenia.

Teoria ewolucji



Dobór naturalny wynika z właściwości organizmów żywych: ich śmiertelności, zdolności rozmnażania się, dziedziczenia cech, jest także wynikiem ograniczonych zasobów środowiska.

Rośliny, zwierzęta i mikroorganizmy różnią się między sobą, a różnice te są zdeterminowane genetycznie, jeśli różnice ten nie tylko są dziedziczne, lecz także wpływają na przeżywanie i rozmnażanie się tych organizmów, to z pokolenia na pokolenie zmieniać się będzie frekwencja różnych form w populacji

Przyjęcie darwinowskiej teorii ewolucji wiąże się z przyjęciem doboru naturalnego jako najistotniejszego mechanizmu kształtującego właściwości gatunków.

Kumulatywne działanie doboru może być bardzo efektywne, dając w długim czasie, jakim dysponowała ewolucja, struktury bardzo skomplikowane i mało prawdopodobne.

Geneza algorytmów genetycznych



John Holland, Uniwersytet Michigan

Opisanie i wyjaśnienie istoty procesów adaptacyjnych występujących w przyrodzie

Stworzenie oprogramowania, odtwarzającego podstawowe mechanizmy rządzące systemami biologicznymi

J.Holland 1975, Adaptation in Natural and Artificial Systems, The University of Michigan Press.

Algorytmy genetyczne



Algorytmy genetyczne są to algorytmy poszukiwania, które w rozwiązywaniu zadań stosują zasady ewolucji i dziedziczności, posługują się populacją potencjalnych rozwiązań, zawierają pewien proces selekcji, oparty na dopasowaniu osobników, i pewne operatory genetyczne.

Każde rozwiązanie ocenia się na podstawie pewnej miary jego dopasowania - funkcja przystosowania (celu).

Nową populację, w kolejnej iteracji, tworzy się przez selekcję osobników najlepiej dopasowanych.

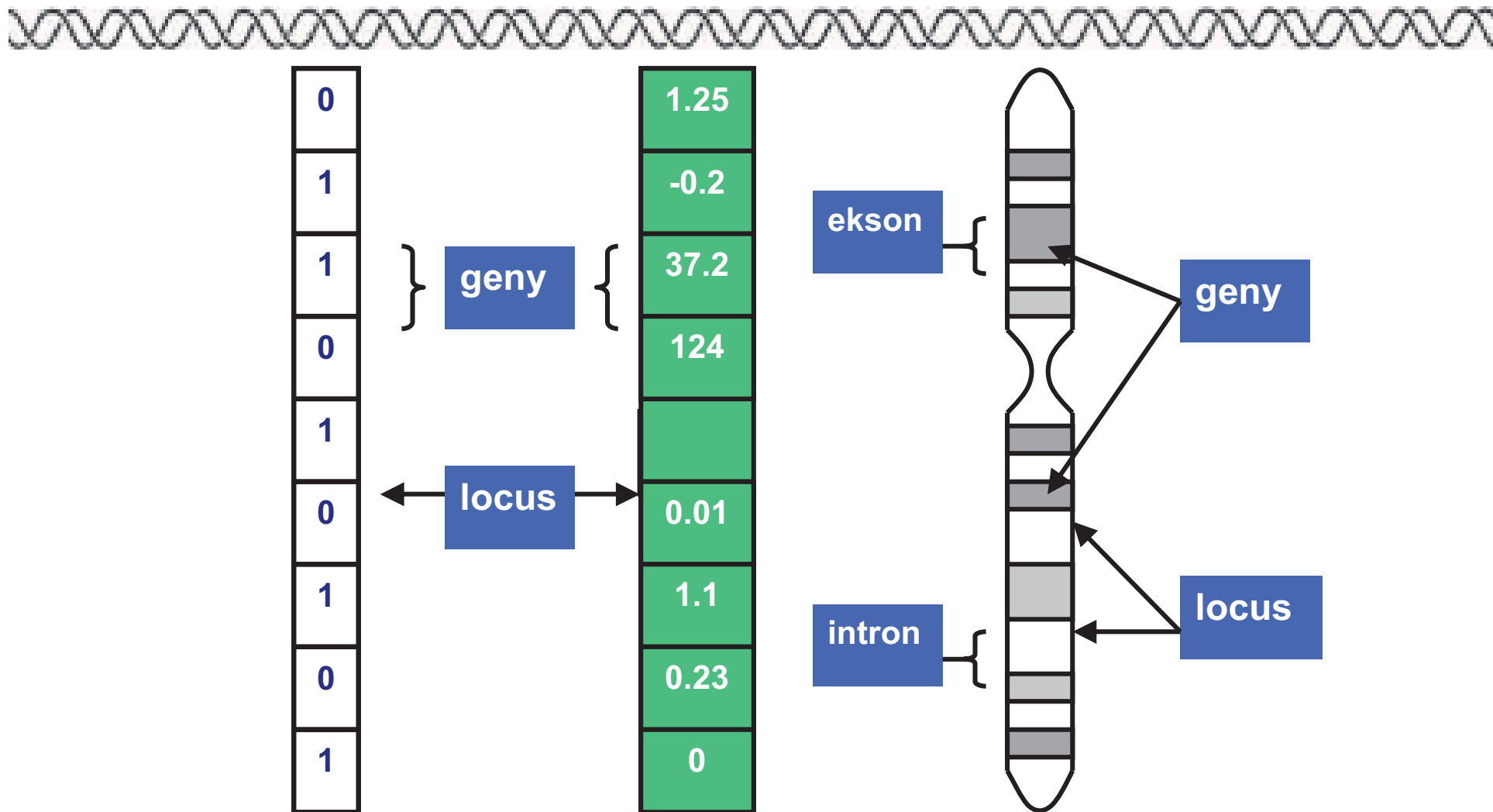
Paradygmat genetyczny



populacyjne sformułowanie zadania

- **funkcja przystosowania**
- **reprezentacja (kodowanie) zadania**
- **selekcja i operatory genetyczne**
- **probabilistyczne reguły wyboru**

Porównanie chromosomów



Porównanie chromosomów: a) chromosom klasycznego algorytmu genetycznego,
b) chromosom algorytmu ewolucyjnego, c) chromosom naturalny.

Algorytmy genetyczne



Elementarny algorytm genetyczny jest skonstruowany z trzech następujących operacji:

reprodukcja - jest procesem, w którym indywidualne ciągi kodowe zostają powielone w stosunku zależnym od wartości, jakie przybiera dla nich funkcja celu, powstaje pula rodzicielska.

- **krzyżowanie** - proces, który polega na losowym skojarzeniu ciągów z puli rodzicielskiej w pary, a następnie krzyżowaniu tzn. wymianie części informacji zawartej w ciągach rodzicielskich i utworzenie potomstwa.

- **mutacja** - polega na sporadycznej, przypadkowej zmianie wartości elementu ciągu kodowego.

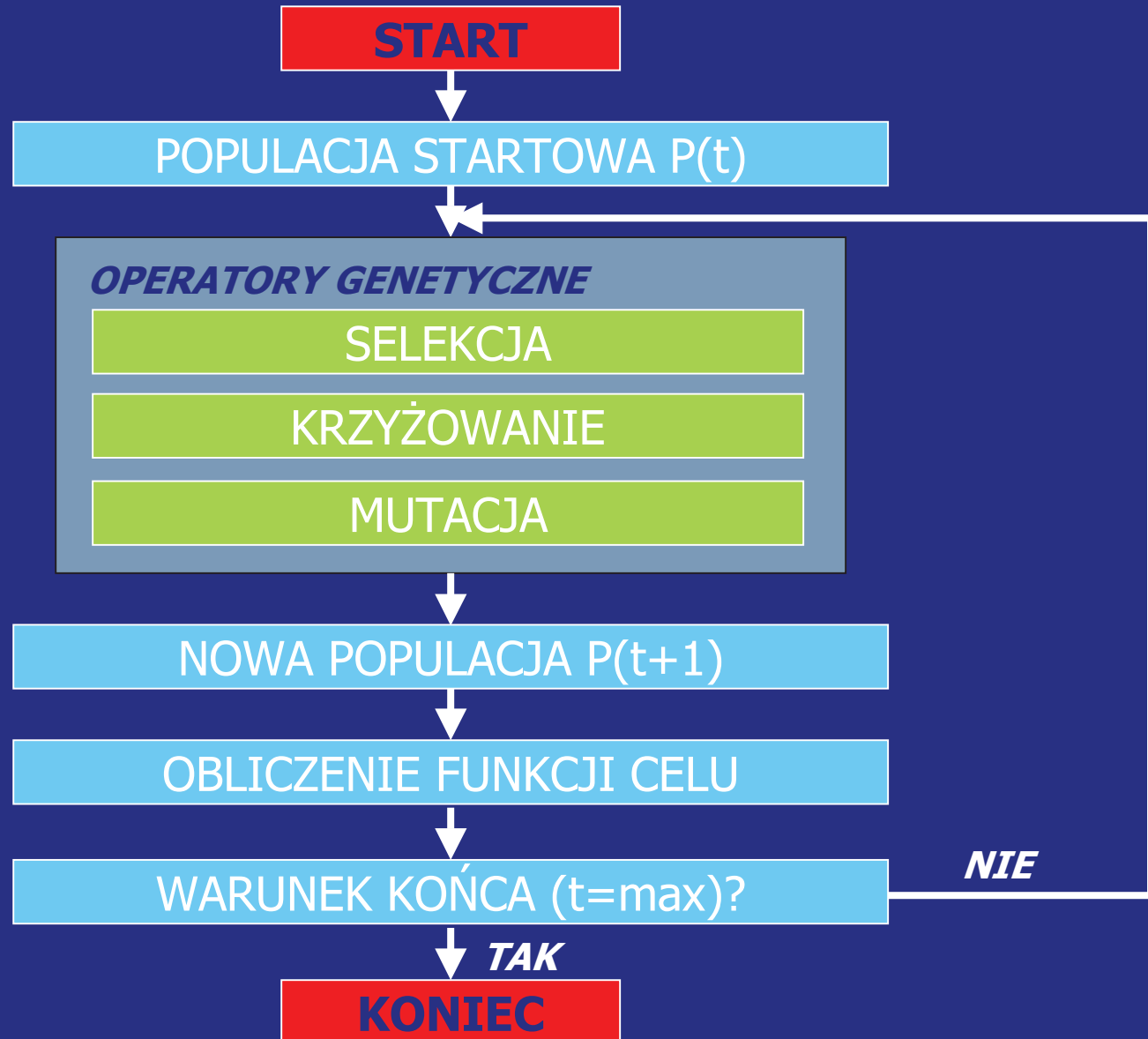
Co to są algorytmy genetyczne?

Algorytmy genetyczne odwzorowują naturalne procesy ewolucyjne zachodzące w czasie, których celem jest maksymalne dopasowanie osobników do istniejących warunków życia.

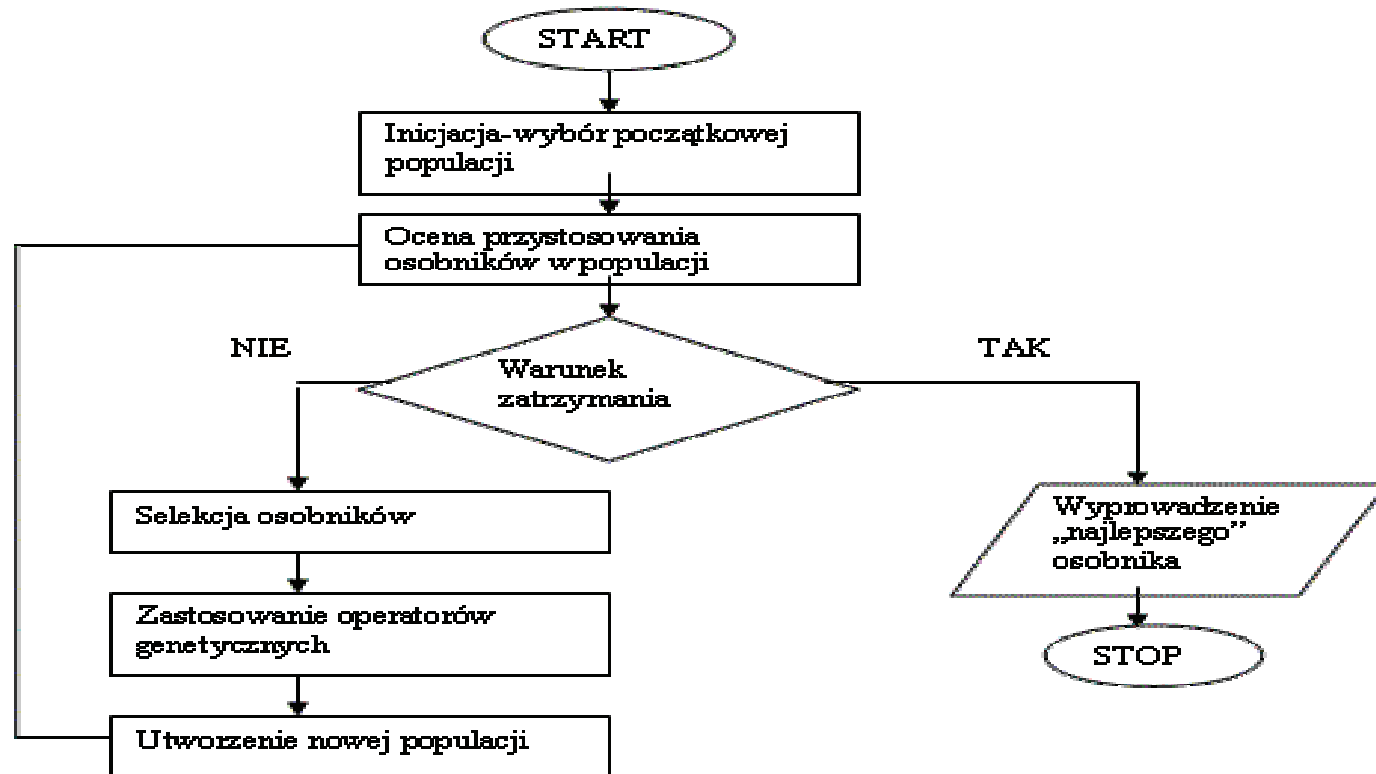
Algorytm genetyczny stanowi wzorowaną na naturalnej ewolucji metodę rozwiązywania problemów, głównie zagadnień optymalizacyjnych. Algorytmy genetyczne są procedurami przeszukiwania opartymi na mechanizmach doboru naturalnego i dziedziczenia. Korzystają z ewolucyjnej zasady Przeżycia osobników najlepiej przystosowanych



Algorytmy genetyczne



Rys 1 Schemat blokowy algorytmu genetycznego



Podstawowe pojęcia algorytmów genetycznych

Populacja — zbiór osobników o określonej liczbie

Osobnikami populacji w algorytmach genetycznych są zakodowane w postaci chromosomów

Chromosomy — inaczej łańcuchy lub ciągi kodowe — to uporządkowane ciągi genów

Gen — najmniejsza składowa chromosomu, decydująca o dziedziczności jednej lub kilku cech



Genotyp czyli *struktura*, to zespół chromosomów danego osobnika

Fenotyp jest zestawem wartości odpowiadających danemu genotypowi

Allel to wartość danego genu, określana też jako *wartość cechy* lub *wariant cechy*.

Locus wskazuje miejsce położenia danego genu w łańcuchu, czyli chromosomie

Funkcja przystosowania jest miarą przystosowania (dopasowania) danego osobnika w populacji



Proste algorytmy genetyczne

W roku 1975 John Holland zaproponował algorytm, którego zadaniem było modelowanie procesu ewolucji. Schemat tego algorytmu, zwanego obecnie **prostym algorytmem genetycznym** (SGA), przedstawia poniższy schemat

Prosty algorytm genetyczny

begin

$t := 0$

inicjacja P^0

ocena P^0

while (not warunek stopu) **do**

begin

$T^t :=$ reprodukcja P^t

$O^t :=$ krzyżowanie i mutacja T^t

ocena O^t

$P^{t+1} = O^t$

$t := t + 1$

end

end



Schemat działania

Algorytm przetwarza populacje:

P^t - zwaną populacją bazową

O^t - zwaną populacją potomną

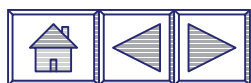
T^t - zwaną populacją tymczasową, w której przechowywane są kopie osobników z P^t .

W każdej z nich zawarta jest jednakowa liczba osobników.

W kroku „inicjacji P^0 ” populacja bazowa wypełniana jest losowo wygenerowanymi osobnikami.

Dla każdego osobnika obliczana jest na etapie oceny wartość funkcji przystosowania.

Po przygotowaniu populacji bazowej P^0 przystępuje się do głównej pętli programu, w której zdefiniowany jest proces sztucznej ewolucji.



Reprodukcja, kopiowanie do populacji tymczasowej T^t losowo wybranych osobników z populacji bazowej. W wyniku reprodukcji w populacji tymczasowej znajdzie się większa liczba lepiej przystosowanych osobników.

Prawdopodobieństwa wylosowania osobników nie są sobie równe.

Szansę na reprodukcję mają osobniki o większej wartości funkcji przystosowania.

Stosowane jest losowanie ze zwracaniem.

Krzyżowanie i mutacja poddaje się im osobniki z T^t . Kojarzy się w pary i dla każdej pary podejmowana jest decyzja o krzyżowaniu.



Decyzja o krzyżowaniu

Pozytywna, następuje krzyżowanie, a powstałe osobniki potomne zastępują swoich rodziców.

Negatywna, para jest pozostawiana bez zmian.

Prawdopodobieństwo zajścia krzyżowania p_c jest parametrem algorytmu.

Mutacja wykonywana dla każdego osobnika z T^t
Jest perturbacją jego genotypu.

Ocena osobników poprzez obliczenie wartości funkcji przystosowania

Populację potomną O^t stanowią osobniki poddane ocenie.

Populacja potomna O^t staje się nową bazową w kolejnym obiegu pętli.

Pętlę wykonuje się dopóty, dopóki nie jest spełniony warunek zatrzymania.



Warunek zatrzymania jest spełniony gdy:

Pojawił się **osobnik** o odpowiednio dużej-
wymaganej wartości funkcji
przystosowania.

Została wykonana założona **Liczba generacji**

Generacja – jedna iteracja- obieg głównej
pętli algorytmu

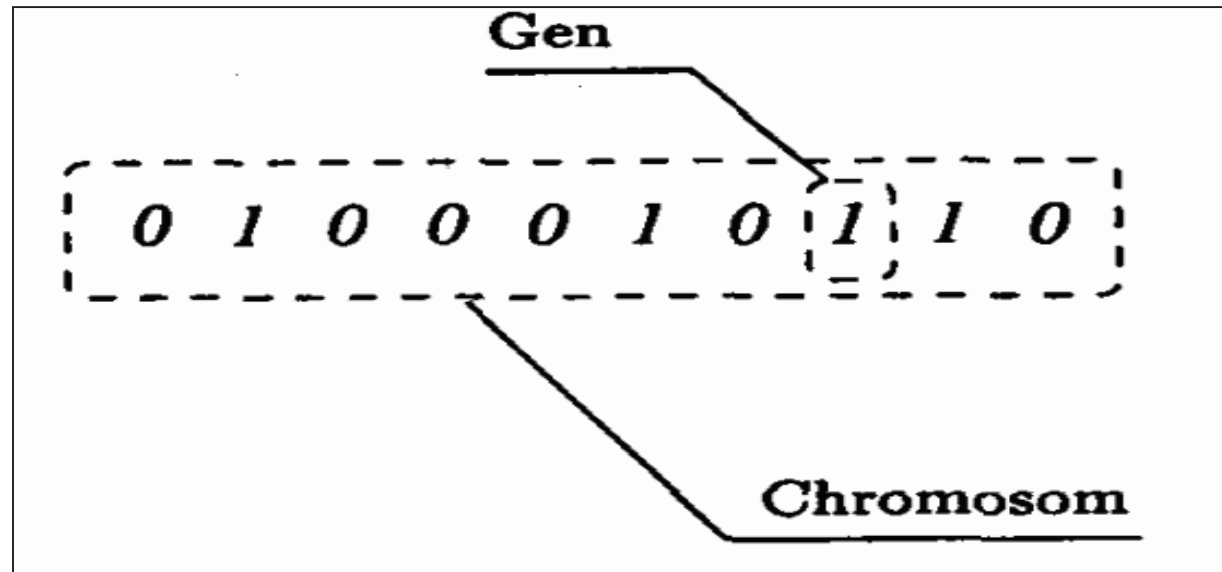
Stan populacji w chwili t określa się
mianem pokolenia P^t



Kodowanie osobników i operacje genetyczne.

Schemat algorytmu zaproponowany przez Hollanda abstrahuje od przyjętego sposobu kodowania osobników i wykonywania operacji krzyżowania i mutacji



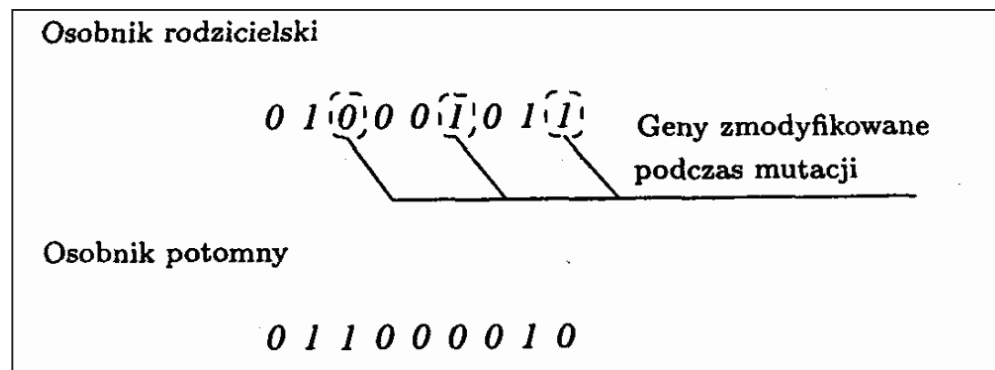


rys.2

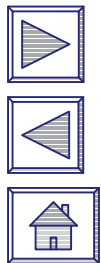
Jednakże algorytm genetyczny najbardziej jest znany z kodowaniem binarnym. W tej wersji chromosom jest więc n -elementowym wektorem genów, z których każdy jest zerem lub jedynką (rys. 2).



Mutacja osobnika jest wykonywana dla każdego genu osobno. Polega ona na tym, że z prawdopodobieństwem p_m , którego wartość jest parametrem algorytmu, wartość genu zmienia się na przeciwną (rys. 3).

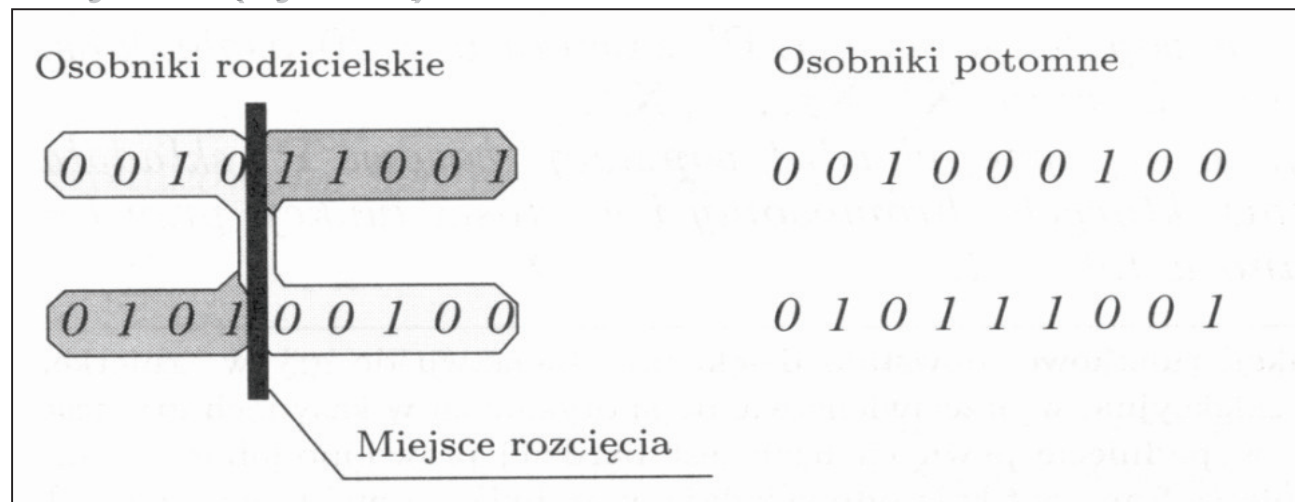


rys. 3



Krzyżowanie osobników udział w nim biorą dwa osobniki rodzicielskie. Operacja wzorowana na procesie przemian w DNA, zachodzących podczas rozmnażania generatywnego.

Chromosomy rodzicielskie rozcinane są na dwa fragmenty. Pierwszy fragment pierwszego chromosomu jest sklejany z drugim fragmentem drugiego chromosomu; podobnie pierwszy fragment drugiego chromosomu z drugim fragmentem pierwszego. Powstałe w ten sposób łańcuchy tworzą chromosomy osobników potomnych (rys.4).



rys.4



Wszystkie chromosomy zawierają jednakową liczbę bitów, dlatego też rozcięcie musi być wykonane w tym samym miejscu dla obu chromosomów.

Miejsce rozcięcia wybierane jest losowo z rozkładem równomiernym

Sposób reprodukcji. Populacja tymczasowa T^t tworzona jest w sposób, preferujący osobniki lepiej przystosowane.



Mutacja



Rodzic

011111000110111110011001010111

Potomek

0111110001101011110011001010111

ZAKRES MUTACJI

Krzyżowanie jednopunktowe



Rodzic 1

011111000110111100011001010111

Rodzic 2

10010000000011100000001000000011

POZYCJA CIĘCIA

Potomek 1

01111100011011000000010000000011

Potomek 2

1001000000001111100011001010111

Krzyżowanie wielopunktowe

Rodzic 1

011111000110101111100011010111

Rodzic 2

10010000000011101100000000000011

POZYCJE CIĘCIA

Potomek 1

011111000110110111100011010111

Potomek 2

10010000000011011100000000000011

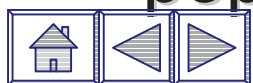
Stosuje się **Reprodukcja proporcjonalną lub ruletkową**.

Określa się w niej zmienną losową na zbiorze dla elementów populacji P^t , o rozkładzie danym wzorem

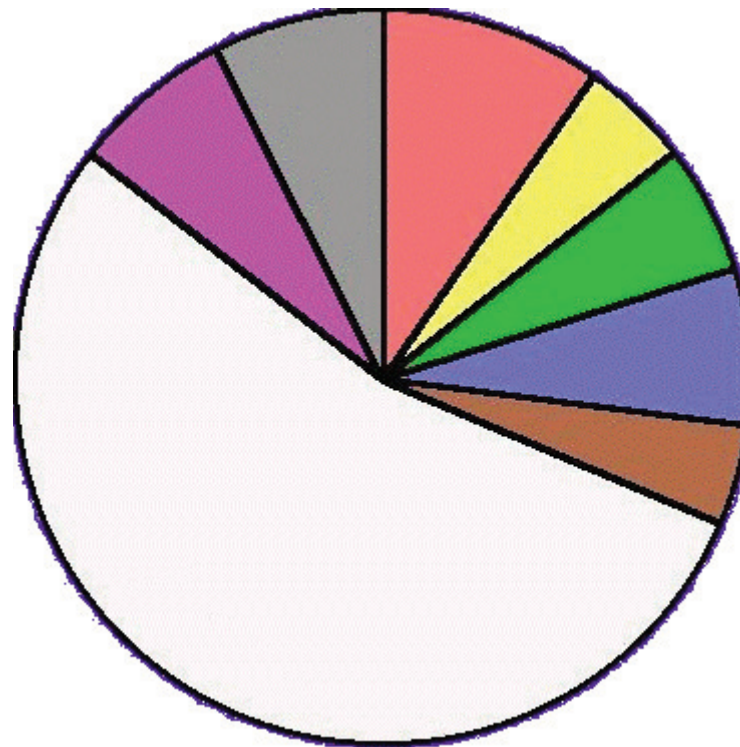
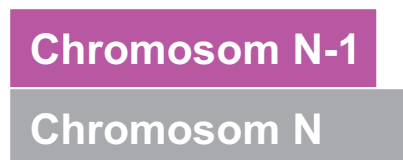
$$p_r(X) = \frac{\phi(X)}{\sum_{Y \in P^t} \phi(Y)}$$

gdzie X oznacza osobnika, zaś $\phi(X)$ jego wartość przystosowania.

Następnie dokonuje się wielokrotnego próbkowania tej zmiennej losowej, za każdym razem kopiując wylosowanego osobnika do populacji tymczasowej T^t .



Selekcja metodą ruletki



PRZYKŁAD

Chromosomy są kodowane binarnie i składają się z $n = 10$ genów bitów

Funkcja celu równa liczbie jedynek w chromosomie

$$\phi(X) = \sum_{j=1}^n X_j$$

Przyjmujemy, że krzyżowanie jednopunktowe jest wykonywane z prawdopodobieństwem $p_c = 0.7$, a prawdopodobieństwo mutacji $p_m = 0.1$. Niech populacja bazowa P^t zawiera $u, = 20$ osobników, ponumerowanych kolejno: $X^1, X^2, \dots, X^u$.



Założmy, że w pewnej chwili t populacja bazowa P^t składała się z osobników, których chromosomy i wartości funkcji przystosowania podano w tab. 1

i	X^i	$\Phi(X^i)$	$p_r(X^i)$	$P_r(X^i)$
1	1111110011	8	0.0824	0.0824
2	0110100100	4	0.0412	0.1237
3	0000110100	3	0.0309	0.1546
4	0110100111	6	0.0618	0.2164
5	1001111001	6	0.0618	0.2783
6	1001110011	6	0.0618	0.3402
7	0000100101	3	0.0309	0.3711
8	1000110100	4	0.0412	0.4123
9	0001011001	4	0.0412	0.4536
10	1011110000	5	0.0515	0.5051
11	0010111101	6	0.0618	0.5670
12	1100010010	4	0.0412	0.6082
13	0111110111	8	0.0824	0.6907
14	0100101100	4	0.0412	0.7319
15	0001100110	4	0.0412	0.7731
16	1001001101	5	0.0515	0.8247
17	1010000110	4	0.0412	0.8659
18	0010101100	4	0.0412	0.9072
19	1101000011	5	0.0515	0.9587
20	0010010011	4	0.0412	1.0000
$\overline{\Phi(X)}$		4.85		

Tab.1



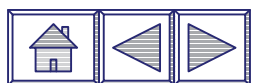
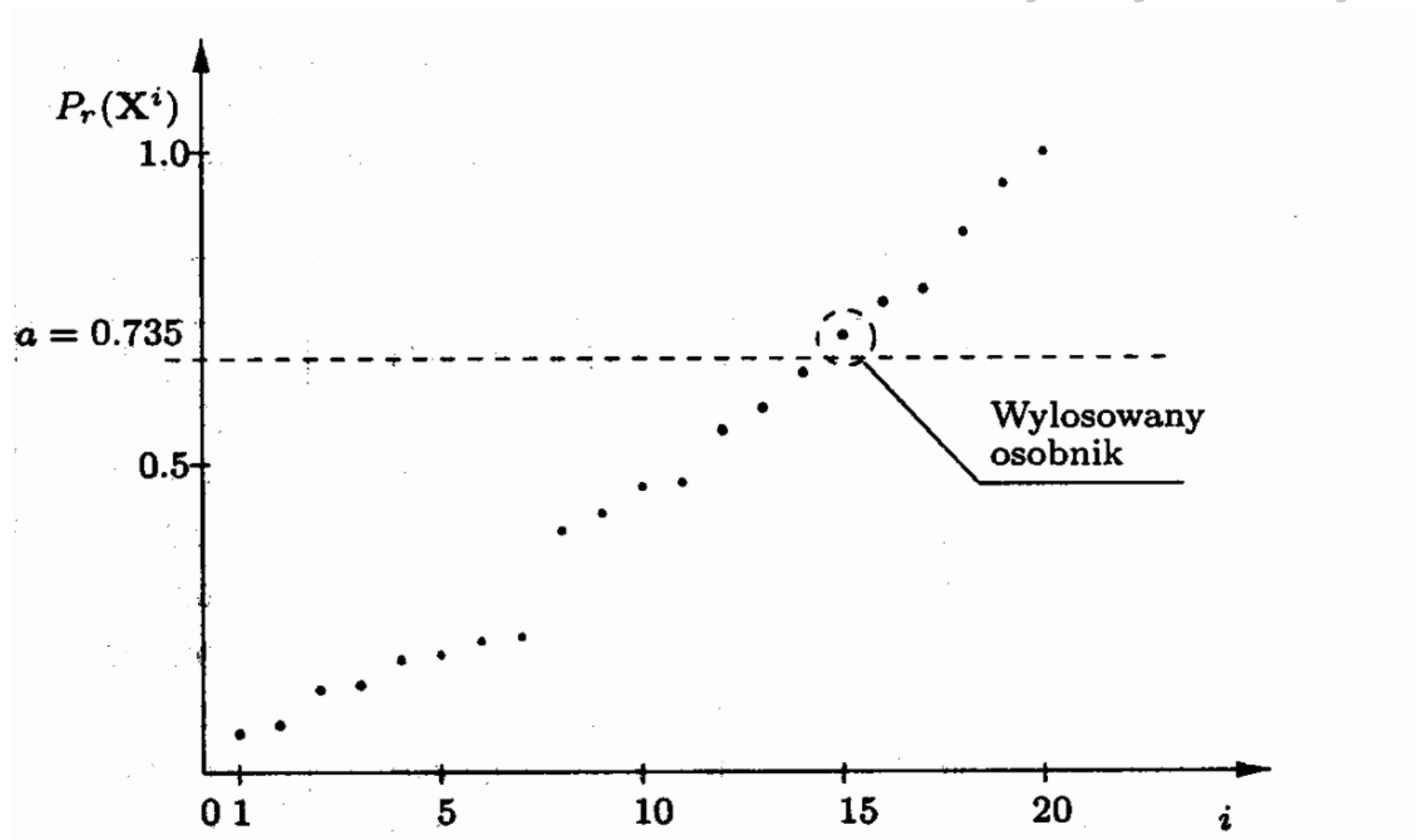
Na podstawie wartości funkcji przystosowania można określić rozkład prawdopodobieństwa reprodukcji $P_r(X^i)$ jego dstrybuantę $p_r(X^i)$
W celu wyznaczenia osobnika do reprodukcji, dokonujemy z rozkładem jednostajnym losowania liczby a z przedziału $(0, 1)^*$. Jeśli spełnia ona warunek

$$a \leq P_r(X^1)$$

to reprodukcji podlega osobnik X , w przeciwnym przypadku reprodukowany jest osobnik X' , dla którego zachodzi (rys.5)



Rys. 5 Sposób realizacji reprodukcji ruletkowej.
Punktami zaznaczono wartości dystrybuanty



W tabeli 2 przedstawiono kolejne kroki tworzenia populacji potomnej O^t na podstawie populacji bazowej w wyniku reprodukcji, krzyżowania i mutacji.

Pary zreprodukowanych osobników poddano krzyżowaniu z prawdopodobieństwem p_c . Pary, dla których nie zapadła decyzja o krzyżowaniu, przechodzą przez ten etap niezmiennie. *Następnie, produkty krzyżowania (lub prostej reprodukcji) ulegają dodatkowo mutacji w taki sposób, że dla każdego genu podejmuje się losowo decyzję o jego mutacji.*

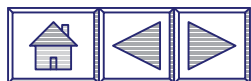


Tab. 2 Tabela cyklu, krzyżowania i mutacji. W kolejnych kolumnach znajdują się: ξ_r - wartość zmiennej losowej wykorzystywanej do określenia numeru reprodukowanego osobnika, i - jego numer, X^i - jego genotyp, $\Phi(X^i)$ - jego przystosowanie, ξ_c - wartość zmiennej losowej opisującej decyzję o krzyżowaniu, k - pozycja rozcięcia do skrzyżowania, Z^i - genotyp osobnika potomnego, $\Phi(Z^i)$ - jego przystosowanie, ξ_m - wektor decyzji o mutacji poszczególnych genów osobnika powstałych w wyniku krzyżowania, Y^i - genotyp powstały w wyniku mutacji $\Phi(Y^i)$ - przystosowanie wynikowego osobnika.



Tab.2

ξ_r	i	X^i	$\Phi(X^i)$	ξ_c	k	Z^i	$\Phi(Z^i)$	ξ_m	Y^i	$\Phi(Y^i)$
0.0267	1	1111110011	8	1	1	1111110011	8	000001000	1111111011	9
0.0428	1	1111110011	8	1	1	1111110011	8	000100000	1110110011	7
0.3426	7	0000100101	3	0	-	0000100101	3	000000100	0000100001	2
0.6351	13	0111110111	8	0	-	0111110111	8	010001100	0011111011	7
0.6975	14	0100101100	4	1	8	0100101100	4	000000001	0100101101	5
0.8737	18	0010101100	4	1	8	0010101100	4	001000000	0000101100	3
0.6515	13	0111110111	8	1	4	0111110000	5	000010000	0111100000	4
0.4964	10	1011110000	5	1	4	1011110111	8	000000000	1011110111	8
0.1706	4	0110100111	6	1	5	0110110111	7	100000000	1110110111	8
0.6526	13	0111110111	8	1	5	0111100111	7	000000000	0111100111	7
0.8075	16	1001001101	5	1	3	1001011001	5	000000001	1001011000	4
0.4402	9	0001011001	4	1	3	0001001101	4	100000100	1001001001	4
0.6374	13	0111110111	8	1	4	0111110011	7	000000000	0111110011	7
0.3319	6	1001110011	6	1	4	1001110111	7	000000000	1001110111	7
0.1149	2	0110100100	4	0	-	0110100100	4	000000000	0110100100	4
0.1312	3	0000110100	3	0	-	0000110100	3	000000000	0000110100	3
0.6967	14	0100101100	4	0	-	0100101100	4	000100000	0101101100	5
0.9837	20	0010010011	4	0	-	0010010011	4	000000100	0010010111	5
0.3618	7	0000100101	3	1	6	0000100000	1	000001100	0000101100	3
0.4661	10	1011110000	5	1	6	1011110101	7	000100000	1010110101	6
$\overline{\Phi(X^i)}$			5.40				5.40			5.40



Średnia wartość przystosowania zreprodukowanych osobników w populacji T^t jest większa niż średnie przystosowanie, w populacji bazowej P^t .

Można było oczekiwać tego zjawiska, gdyż mechanizm reprodukcji tak skonstruowano, aby promować lepsze osobniki. Zmniejsza się jednak różnorodności T^t w stosunku do P^t . Operatory genetyczne przywracają różnorodność w O^t , nie gwarantując poprawy średniego przystosowania. W podanym przykładzie, ze względu na rodzaj funkcji przystosowania, operator krzyżowania nie



wprowadza zmian wartości średniej, gdyż nie zmienia liczby jedynek w chromosomie. Podobnie mutacja, która z jednakowym prawdopodobieństwem zamienia, jedynkę na zero i odwrotnie, ma z tych samych powodów neutralny (w sensie wartości oczekiwanej) wpływ na średnie przystosowanie. Nie musi tak być (i w większości praktycznych problemów nie jest) i wówczas za poprawę różnorodności przychodzi zapłacić zmniejszeniem średniej wartości funkcji przystosowania. Spróbujmy teraz uściślić ocenę zmian różnorodności populacji poczynioną jedynie na podstawie



obserwacji zawartości populacji. Dla potrzeb przykładu założmy, że o różnorodności populacji świadczy wektor wartości własnych macierzy kowariancji genotypów w tej populacji. W przypadku idealnego zróżnicowania, wartości własne powinny być sobie równe. Im większe zależności między genotypami osobników, tym mniejsza różnorodność populacji i większe wartości wektora kowariancji oraz większe ich zróżnicowanie *W tabeli 3 podano wartości własne macierzy kowariancji populacji wyjściowej, populacji potomnej po reprodukcji, po krzyżowaniu i po mutacji.*



Tabela 3 Zmiany wartości własnych kowariancji genotypów w kolejnych etapach cyklu algorytmu genetycznego

Przed reprodukcją	Po reprodukcji	Po krzyżowaniu	Po mutacji
0.6798	0.8701	0.8954	0.8121
0.4677	0.5259	0.4417	0.4587
0.3087	0.2949	0.2476	0.3664
0.2859	0.2028	0.2092	0.2125
0.2127	0.1443	0.1772	0.1598
0.1661	0.1229	0.1174	0.1453
0.1630	0.0926	0.0924	0.1017
0.0991	0.0359	0.0785	0.0878
0.0636	0.0271	0.0415	0.0565
0.0508	0.0045	0.0200	0.0307

Tab.3

Nietrudno zauważyć, że największa zmiana różnorodności następuje podczas reprodukcji. Jest to spowodowane faktem, że niektóre chromosomy zreprodukowano wielokrotnie, niektóre zaś pominięto w tym procesie, w zależności od wartości funkcji przystosowania



Krzyżowanie nie wprowadza nowego materiału genetycznego (wariancja poszczególnych genów pozostaje bez zmian), lecz powoduje jego „przetasowanie”. Prowadzi to do zwiększenia różnorodności, czyli poprawy rozkładu wartości własnych macierzy kowariancji. Z kolei mutacja wprowadza modyfikacje materiału genetycznego, co, podobnie jak w krzyżowaniu, owocuje poprawą rozkładu wartości własnych. Stopień poprawy jest nieco mniejszy niż w przypadku krzyżowania wynika to z faktu, że mutacja jest operatorem oddziałującym na pojedyncze geny, podczas gdy krzyżowanie na ich całe bloki



Implementacja binarna

Przypuśćmy, że chcemy maksymalizować funkcję k zmiennych $f(x_1, \dots, x_k): R^k \rightarrow R$.

Zakładamy, że każda zmienna x_i może przybierać wartości z przedziału

$$D_i = [a_i, b_i] \subseteq R \text{ i } f(x_1, \dots, x_k) > 0$$

dla wszystkich $x \in D_i$.

Chcemy optymalizować funkcje f z pewną żadaną dokładnością. Zakładamy że jest to sześć najbardziej znaczących cyfr dziesiętnych w każdej zmiennej.



Aby uzyskać taką dokładność , każdy przedział D_i dzielimy na

$(b_i - a_i) * 10^6$ podprzedziałów.

Oznaczamy prze m_i najmniejszą liczbę całkowitą taką że

$$(b_i - a_i) * 10^6 < 2^{m_i} - 1$$

Wtedy reprezentacja, w której każda zmienna x_i jest zakodowana jako łańcuch binarny o długości m_i , będzie spełniała wymagania dokładności. Wartość tego łańcucha można wyrazić w postaci dziesiętnej za pomocą wzoru:

$$x_i = a_i + \frac{decimal(1001...001_2) \cdot (b_i - a_i)}{2^{m_i} - 1}$$

gdzie $decimal(lancuch_2)$

jest równy dziesiętnej wartości łańcucha binarnego.



Każdy chromosom (jako potencjalne rozwiązanie) jest reprezentowany przez łańcuch o długości

$$m = \sum_{i=1}^k m_i$$

Pierwsze m_1 bitów odpowiada zmiennej $x_1 \in [a_1, b_1]$

Druga grupa m_2 opowiada zmiennej $x_2 \in [a_2, b_2]$

▪

▪

m_i odpowiada zmiennej $x_i \in [a_i, b_i]$

▪

▪

m_k odpowiada zmiennej $x_k \in [a_k, b_k]$

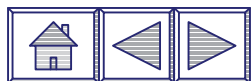
W celu ustalenia populacji początkowej możemy wygenerować liczbę `pop_size` chromosomów losowo, bit po bicie.

Jeżeli dysponujemy pewną wiedzą o rozkładzie możliwych optimów, możemy użyć tej informacji do odpowiedniego rozmieszczenia początkowych rozwiązań

W każdym pokoleniu oceniamy wszystkie chromosomy (używając funkcji f ograniczonej do zakodowanych sekwencji zmiennych).

Wybieramy nową populację zgodnie z rozkładem prawdopodobieństwa określonym na wartościach dopasowań i zmieniamy chromosomy w nowej populacji za pomocą operatorów mutacji i krzyżowania.

Po pewnej liczbie pokoleń, kiedy nie obserwujemy już poprawy, najlepsze chromosomy reprezentują (mamy nadzieję, że globalne) rozwiązanie optymalne.



W procesie selekcji używa się ruletki o wielkości pól zgodnej z wartościami dopasowań.

Ruletkę konstruuje się następująco:

- Oblicz wartość dopasowania $eval(\mathbf{v}_i)$ dla każdego chromosomu $\mathbf{v}_i$ ($i=1, \dots, pop_size$),
- Oblicz całkowite dopasowanie populacji

$$F = \sum_{i=1}^{pop_size} eval(\mathbf{v}_i)$$

- Oblicz prawdopodobieństwo wyboru p_i każdego chromosomu $\mathbf{v}_i$

$$p_i = \frac{eval(\mathbf{v}_i)}{F}$$

- Oblicz dystrybuantę q_i dla każdego chromosomu $\mathbf{v}_i$

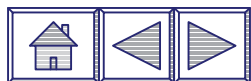
$$q_i = \sum_{j=1}^i p_j$$



W procesie selekcji używa się ruletki o wielkości pól zgodnej z wartościami dopasowań.

Proces selekcji jest oparty na obrocie ruletką *pop_size* razy i wyborze za każdym razem jednego chromosomu do nowej populacji w następujący sposób:

- Wygeneruj zmiennopozycyjną liczbę r z zakresu $[0,1]$
- Jeżeli $r < q_1$ to wybierz chromosom v_1 ; jeżeli nie to wybierz chromosom v_i ($2 < i < \text{pop_size}$), dla którego zachodzi $q_{i-1} < r < q_i$

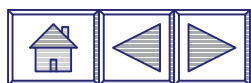


Pewne chromosomy będą wybrane więcej niż raz. Jest to zgodne z twierdzeniem o schematach które mówi, że z najlepszych chromosomów powstaje więcej kopii, ze średnich tyle samo, a najgorsze wymierają.

Można zastosować operator mieszania, to znaczy krzyżowanie, do osobników z nowej populacji.

Jednym z parametrów systemu genetycznego jest prawdopodobieństwo krzyżowania p_c .

Umożliwia nam ono obliczenie oczekiwanej liczby chromosomów $p_c \bullet pop_size$, które ulegną operacji krzyżowania.



Postępujemy w następujący sposób:

Dla każdego chromosomu z (nowej) populacji:

- wygeneruj (zmiennopozycyjną) liczbę losową r z zakresu $[0, 1]$,
- jeżeli $r < p_c$, to wybierz rozpatrywany chromosom do krzyżowania

Następnie dobieramy wybrane chromosomy losowo w pary. Dla każdej pary chromosomów generujemy losową liczbę całkowitą pos z zakresu $[1, m - 1]$ (m jest całkowitą długością - liczbą bitów - chromosomu).



Liczba pos określa punkt cicia chromosomu do wymiany

$(b_1 \ b_2 \ \dots \ b_{pos} \ b_{pos+1} \ \dots \ b_m)$

$(c_1 \ c_2 \ \dots \ c_{pos} \ c_{pos+1} \ \dots \ c_m)$

Są zamieniane na

$(b_1 \ b_2 \ \dots \ b_{pos} \ c_{pos+1} \ \dots \ c_m)$

$(c_1 \ c_2 \ \dots \ c_{pos} \ b_{pos+1} \ \dots \ b_m)$

Następna operacja, mutacja, jest wykonywana na bitach. Na podstawie parametru systemu genetycznego, prawdopodobieństwa mutacji p_m , możemy obliczyć oczekiwaną liczbę zmutowanych bitów $p_m \bullet m \bullet pop_size$



Każdy bit (we wszystkich chromosomach i w całej populacji) ma równe szanse na mutację, to znaczy zmiany z 0 na 1 lub odwrotnie. Postępujemy więc następująco:

Dla każdego chromosomu bieżącej (po krzyżowaniu) populacji i dla każdego bitu w chromosomie:

- wygeneruj (zmiennopozycyjną) liczbę losową r z zakresu $[0, 1]$
- jeżeli $r < p_m$, to zmutuj bit.

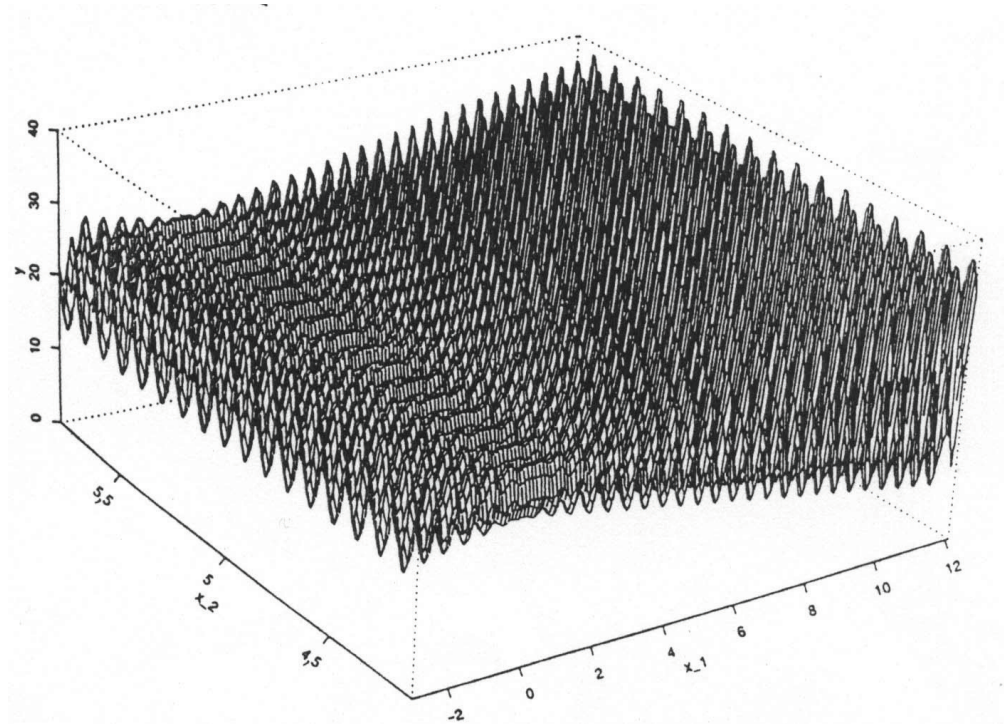


Przykład

Maksymalizujemy funkcję:

$$f(x_1, x_2) = 21,5 + x_1 \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Przy ograniczeniach $-3,0 < x_1 < 12,1$ i $4,1 < x_2 < 5,8$



Zakładamy że żądana dokładność wynosi cztery najbardziej znaczące cyfry dla każdej zmiennej. Przedział zmienności x_1 ma długość 15,1. Z żądanej dokładności wynika, że przedział $[-3,0, 12,1]$ należy podzielić na przynajmniej $15,1 * 10\ 000$ równych podprzedziałów. Oznacza to, że początkowa część chromosomu musi zawierać $m_1 = 18$ bitów, gdyż

$$2^{17} < 151000 < 2^{18}$$



Przedział zmienności x_2 ma długość 1,7. Z żądanej dokładności wynika, że przedział [4,1, 5,8] należy podzielić na przynajmniej $1,7 * 10\ 000$ równych podprzedziałów. Oznacza to, że końcowa część chromosomu musi zawierać $m_2 = 15$ bitów, gdyż $2^{14} < 17\ 000 < 2^{15}$

Całkowita długość chromosomu (wektora rozwiązań) wynosi więc $m = m_1 + m_2 = 18 + 15 = 33$ bitów.

W pierwszych 18 bitach jest zakodowane x_1 , a w pozostałych 15 bitach (19 - 33) x_2



Rozważmy chromosom:

$$v_i = (010001001011010000111110010100010)$$

$$\begin{array}{c} x_1 \\ m_1=18 \end{array}$$

$$\begin{array}{c} x_2 \\ m_2=15 \end{array}$$

$$x_1 = -3,0 + decimal(010001001011010000_2)(12,1 - (-3,0)) / (2^{18} - 1)$$

$$x_1 = 1,052426$$

$$x_2 = 4,1 + decimal(111110010100010_2)(5,8 - 4,1) / (2^{15} - 1)$$

$$x_2 = 5,755330$$

$$f(x_1, x_2) = f(1,052426; 5,755330) =$$

Do optymalizacji funkcji f za pomocą algorytmu genetycznego utworzymy populację o liczbie `pop_size = 20` chromosomów.

Wszystkie 33 bity w każdym chromosomie są na początku wybierane losowo

Zakładamy że po procesie początkowego wyboru chromosomów otrzymaliśmy następującą populację



$V_1 = (100110100000001111111010011011111)$
 $V_2 = (111000100100110111001010100011010)$
 $V_3 = (000010000011001000001010111011101)$
 $V_4 = (100011000101101001111000001110010)$
 $V_5 = (000111011001010011010111111000101)$
 $V_6 = (000101000010010101001010111111011)$
 $V_7 = (001000100000110101111011011111011)$
 $V_8 = (100001100001110100010110101100111)$
 $V_9 = (010000000101100010110000001111100)$
 $V_{10} = (000001111000110000011010000111011)$
 $V_{11} = (110100010111101101000101010000000)$
 $V_{12} = (111011111010001000110000001000110)$
 $V_{13} = (010010011000001010100111100101001)$
 $V_{14} = (011001111110110101100001101111000)$
 $V_{15} = (1110111011011110000100011111011110)$
 $V_{16} = (110011110000011111100001101001011)$
 $V_{17} = (011010111111001111010001101111101)$
 $V_{18} = (011101000000001110100111110101101)$
 $V_{19} = (0001010100111111111110000110001100)$
 $V_{20} = (101110010110011110011000101111110)$



W kroku oceny dekodujemy każdy chromosom i obliczamy funkcję oceniającą dla właśnie rozkodowanych wartości (x_1 , x_2). Dostajemy:

$$\text{eval}(V_1) = f(6,084492, 5,652242) = 26,019600$$

$$\text{eval}(V_2) = f(10,348434, 4,380264) = 7,580015$$

$$\text{eval}(V_3) = f(-2,516603, 4,390381) = 19,526329$$

$$\text{eval}(V_4) = f(5,278638, 5,593460) = 17,406725$$

$$\text{eval}(V_5) = f(-1,255173, 4,734458) = 25,341160$$

$$\text{eval}(V_6) = f(-1,811725, 4,391937) = 18,100417$$

$$\text{eval}(V_7) = f(-0,991471, 5,680258) = 16,020812$$

$$\text{eval}(V_8) = f(4,9i06i8, 4, 7031Ji8) = 17,959701$$

$$\text{eval}(V_9) = f(0,795406, 5,381472) = 16,127799$$

$$\text{eval}(V_{10}) = f(-2,554851, 4,793707) = 21,278435$$

$$\text{eval}(V_{11}) = f(3,130078, 4,996097) = 23,410669$$

$$\text{eval}(V_{12}) = f(9,356179, 4,239457) = 15,011619$$

$$\text{eval}(V_{13}) = f(11,134646, 5,378671) = 27,316702$$

$$\text{eval}(V_{14}) = f(1,335944, 5,151378) = 19,876294$$

$$\text{eval}(V_{15}) = f(11,089025, 5,054515) = 30,060205$$

$$\text{eval}(V_{16}) = f(9,211598, 4,993762) = 23,867227$$

$$\text{eval}(V_{17}) = f(3,367514, 4,571343) = 13,696165$$

$$\text{eval}(V_{18}) = f(3,843020, 5,158226) = 15,414128$$

$$\text{eval}(V_{19}) = f(-1746635, 5,395584) = 20,095903$$

$$\text{eval}(V_{20}) = f(7,935998, 4,757338) = 13,666916$$



Kiedy zatrzymać algorytm genetyczny?

Najlepiej nigdy ponieważ algorytm genetyczny jest procesem losowym i nie sposób zagwarantować osiągnięcia wyniku w skończonej liczbie generacji.

