

Algorytmy ewolucyjne

Algorytmy ewolucyjne wykorzystują mechanizmy naturalnej ewolucji (selekcja, przetrwanie osobników najlepiej przystosowanych, reprodukcja).

Podstawowy algorytm genetyczny został wprowadzony przez Johna Hollanda i rozwinięty dzięki jego pracom z lat 60 i 70.

Idea algorytmów ewolucyjnych

1. Przyjmujemy początkową populację osobników żyjących w danym środowisku
1. Za pomocą odpowiednio zdefiniowanej funkcji przystosowania sprawdzamy ich stopień przystosowania
1. Osobniki wymieniają między sobą materiał genetyczny i powstają nowe osobniki
1. Przeżywają osobniki najlepiej przystosowane

Problemy optymalizacji a algorytmy ewolucyjne

Metody poszukiwania optymalnych rozwiązań:

- Analityczne – rozwiązanie układu równań
- Przeglądowe – sprawdzenie całej przestrzeni poszukiwań
- Losowe – losowe sprawdzenie przestrzeni poszukiwań

Problemy optymalizacji a algorytmy ewolucyjne

Metody ewolucyjne różnią się od klasycznych następującymi cechami:

- Nie przetwarzają bezpośrednio parametrów zadania lecz ich zakodowaną postać
- Korzystają tylko z funkcji celu a nie z jej pochodnych lub innych pomocniczych informacji
- Prowadzą przeszukiwanie wychodząc nie z pojedynczego punktu, lecz z pewnej ich populacji
- Stosują probabilistyczne a nie deterministyczne reguły wyboru

Algorytmy ewolucyjne - definicje

chromosom	łańcuch, ciąg uporządkowany genów; zakodowana postać potencjalnego rozwiązania (punktu przestrzeni poszukiwań)
gen	pojedynczy element chromosomu
allel	wartość genu; w PAG równa 0 lub 1
locus	pozycja (miejsce) genu w chromosomie
genotyp	zespół chromosomów danego osobnika; w PAG genotyp stanowi jeden chromosom
osobnik	potencjalne rozwiązanie (punkt przestrzeni poszukiwań); w postaci genotypu lub fenotypu
fenotyp	odpowiednik genotypu, czyli osobnik w formie nie zakodowanej; punkt przestrzeni poszukiwań, którego zakodowaną postacią jest chromosom (genotyp)
populacja	zbiór osobników (chromosomów) o określonej liczebności
przystosowanie	przystosowanie osobników w populacji, oceniane za pomocą tzw. <i>funkcji przystosowania</i> (nazywanej też <i>funkcją dopasowania</i> lub <i>funkcją oceny</i>)

Operatory genetyczne

Krzyżowanie

- Wybieramy pulę rodzicielską i kojarzymy chromosomy w pary
- Losujemy pozycję genu (locus) w chromosomie określającą punkt krzyżowania. Jeśli każdy z rodziców składa się z L genów to punkt krzyżowania jest liczbą l z zakresu $[1, L-1]$.
- W wyniku krzyżowania otrzymuje się parę potomków:
 1. Potomek, którego chromosom składa się z genów na pozycjach od 1 do l pochodzących od pierwszego rodzica i pozostałych genów pochodzących od drugiego rodzica
 2. Potomek, którego chromosom składa się z genów na pozycjach od 1 do l pochodzących od drugiego rodzica i pozostałych genów pochodzących od pierwszego rodzica

Operatory genetyczne

Krzyżowanie jednopunktowe - przykład

$$\left. \begin{array}{l} ch_3 = [1 \ 1 \mid 1 \ 0 \ 1] \\ ch_1 = [1 \ 0 \mid 1 \ 0 \ 1] \end{array} \right\} \xrightarrow{\text{krzyżowanie}} \begin{array}{l} ch_3 = [1 \ 1 \ 1 \ 0 \ 1] \\ ch_1 = [1 \ 0 \ 1 \ 0 \ 1] \end{array}$$

locus=2

$$\left. \begin{array}{l} ch_7 = [1 \mid 0 \ 1 \ 0 \ 1] \\ ch_5 = [1 \mid 0 \ 0 \ 1 \ 1] \end{array} \right\} \xrightarrow{\text{krzyżowanie}} \begin{array}{l} ch_6 = [1 \ 0 \ 0 \ 1 \ 1] \\ ch_1 = [1 \ 0 \ 1 \ 0 \ 1] \end{array}$$

locus=1

$$\left. \begin{array}{l} ch_5 = [1 \ 0 \ 0 \mid 1 \ 1] \\ ch_3 = [1 \ 1 \ 1 \mid 0 \ 1] \end{array} \right\} \xrightarrow{\text{krzyżowanie}} \begin{array}{l} [1 \ 0 \ 0 \ 0 \ 1] \\ [1 \ 1 \ 1 \ 1 \ 1] \end{array}$$

locus=3

Operatory genetyczne

Krzyżowanie wielopunktowe

Wybierane są dwa lub więcej punkty krzyżowania chromosomów

Usprawnia proces krzyżowania w przypadku korzystania z długich chromosomów



locus=2 i 5

Operatory genetyczne

Krzyżowanie wielopunktowe dla czterech punktów krzyżowania

Wylosowano następujące miejsca krzyżowania: 1, 4, 6 i 9.

$$\begin{array}{l} [1|101|00|111|0] \\ [1|011|11|110|0] \end{array} \xrightarrow{\text{krzyżowanie}} \begin{array}{l} [1|011|00|110|0] \\ [1|101|11|111|0] \end{array}$$

Operatory genetyczne

Mutacja

Zmienia wartość wybranego losowo genu w chromosomie na przeciwny (1 na 0, 0 na 1)

Mutacja zachodzi bardzo rzadko – prawdopodobieństwo mutacji jest zwykle dużo mniejsze niż krzyżowania

Celem mutacji jest wprowadzenie różnorodności populacji

$$ch_i = [1 \ 0 \ 1 \ 0 \ 1] \xrightarrow{\text{mutacja}} ch_i' = [1 \ 0 \ 1 \ 1 \ 1]$$

locus=4

Operatory genetyczne

Inwersja

Działa na pojedynczym chromosomie, zmieniając kolejność alleli między dwoma losowo wybranymi pozycjami chromosomu.

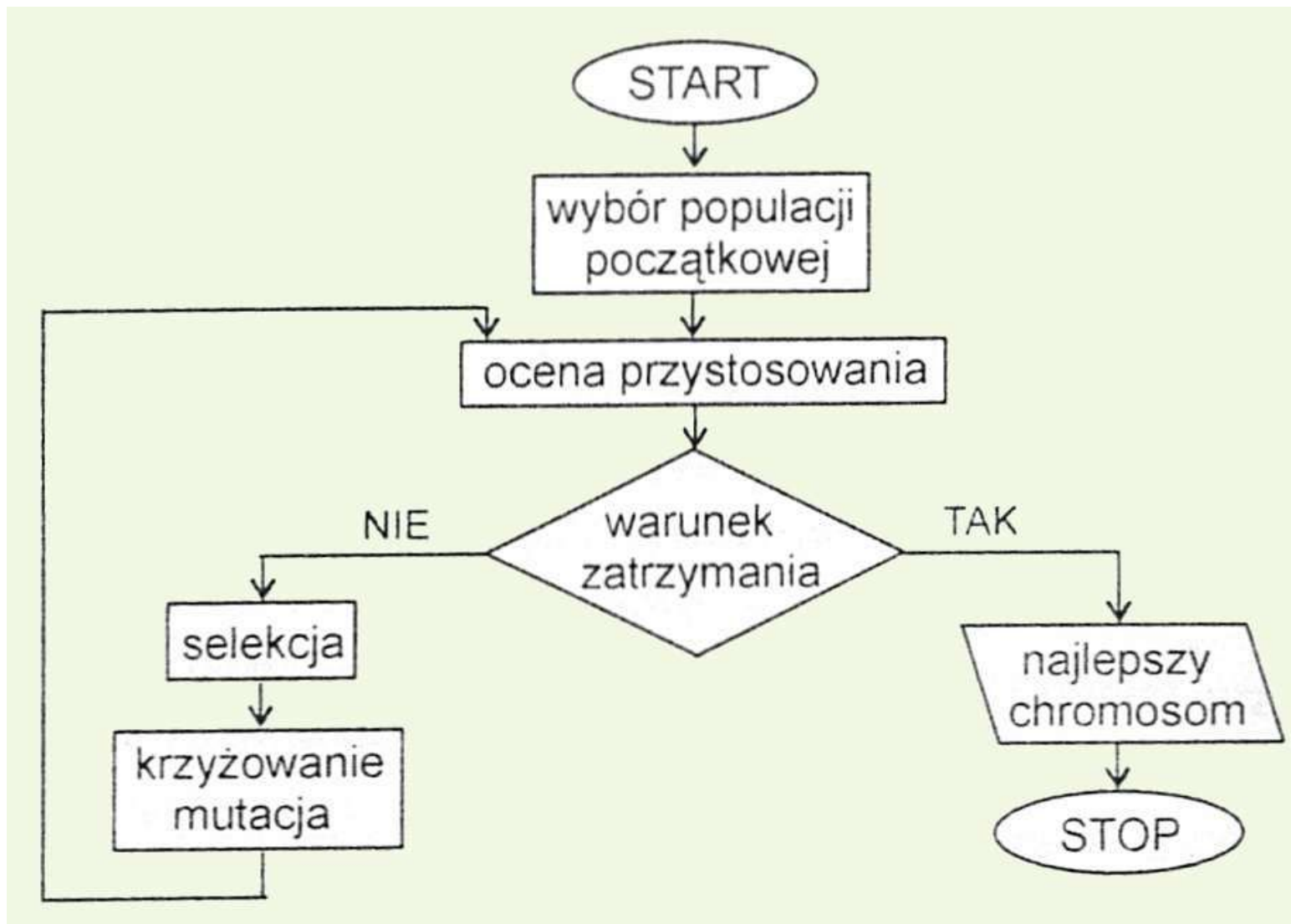
Nie jest to operator często stosowany w algorytmach genetycznych.

Przykład:

Wylosowano pozycje 4 i 10.

$[1101|001110|0011] \xrightarrow{\text{inwersja}} [1101|011100|0011]$

Klasyczny algorytm genetyczny

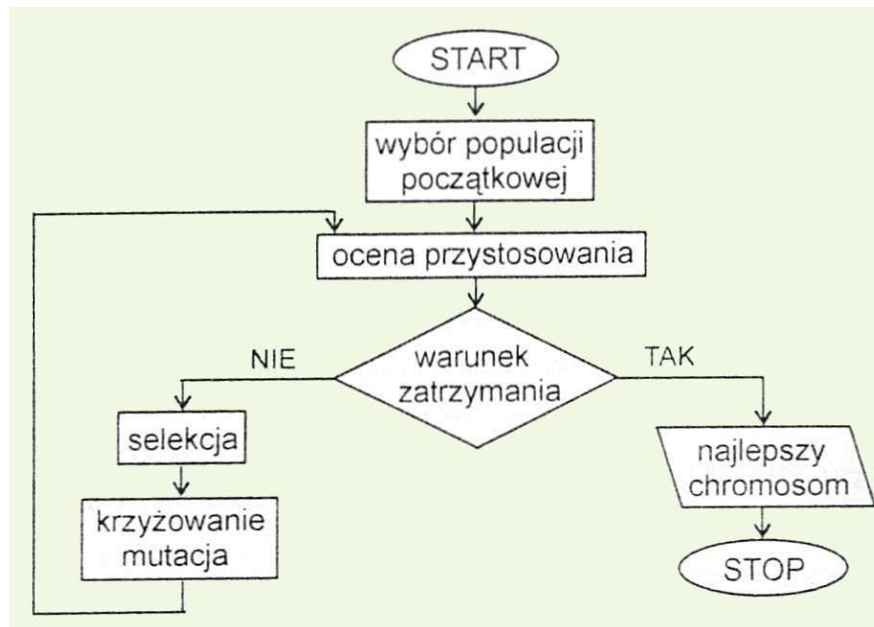


Klasyczny algorytm genetyczny



1. Inicjacja, czyli utworzenie populacji początkowej, polega na losowym wyborze żądanej liczby chromosomów (osobników) reprezentowanych przez ciągi binarne o określonej długości.

Klasyczny algorytm genetyczny



2. Ocena przystosowania chromosomów – obliczenie wartości funkcji przystosowania dla każdego z chromosomów. Postać funkcji zależy od rozwiązywanego problemu, przyjmuje zawsze wartości nieujemne.

Klasyczny algorytm genetyczny



3. Sprawdzenie warunku zatrzymania. Warunek zatrzymania to może być określona wartość błędu, sytuacja gdy dalsze działanie algorytmu nie poprawia uzyskanej, najlepszej wartości, przekroczenie określonego czasu działania lub liczby iteracji algorytmu.

Klasyczny algorytm genetyczny



4. Selekcja chromosomów polega na wybraniu (na podstawie wartości funkcji przystosowania), tych chromosomów, które będą brały udział w tworzeniu potomków do następnej generacji.

W wyniku procesu selekcji powstaje populacja rodzicielska o **takiej samej liczebności** jak bieżąca populacja.

Klasyczny algorytm genetyczny



5. Utworzenie nowej populacji rodzicielskiej po zastosowaniu operatorów krzyżowania i mutacji.

Klasyczny algorytm genetyczny



6. Wyprowadzenie najlepszego chromosomu. Po spełnieniu warunku zatrzymania należy wyprowadzić najlepszy chromosom czyli podać rozwiązanie problemu. Najlepszym rozwiązaniem jest chromosom o największej wartości funkcji przystosowania.

Metody selekcji

Metoda koła ruletki

Selekcja dokonuje się, poprzez wybór chromosomów, którym na kole (koło ruletki) przydzielono sektory proporcjonalne do wartości przystosowania

Większa wartość przystosowania = częstszy wybór do populacji rodzicielskiej

Lepiej przystosowane chromosomy mogą być wybierane wielokrotnie

Skutek: miejsce „słabszych” zajmują „mocniejsi”

Metody selekcji

Metoda koła ruletki

Wielkość sektorów na *kole ruletki* przydzielane są według następujących wzorów:

Wielkość procentowa sektorów:

$$v(ch_i) = p_s(ch_i) \cdot 100\%$$

Prawdopodobieństwo selekcji chromosomu:

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

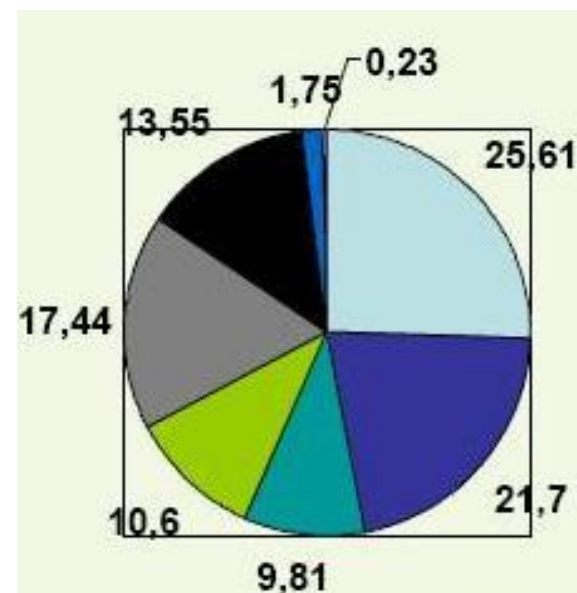
Średnie przystosowanie populacji:

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

Metody selekcji

Metoda koła ruletki – przykład

chromosom	fenotyp	funkcja przystosowania	wielkość procentowa sektora
1111110	(126)	31754	25,61%
1110100	(116)	26914	21,70%
1001110	(78)	12170	9,81%
1001111	(79)	12484	10,6%
1101000	(104)	21634	17,44%
1011011	(91)	16564	13,35%
0100001	(33)	2180	1,75%
0001100	(12)	290	0,23%



1111110
1110100
1001110
1001111
1101000
1011011
0100001
0001100

Metody selekcji

Selekcja rankingowa

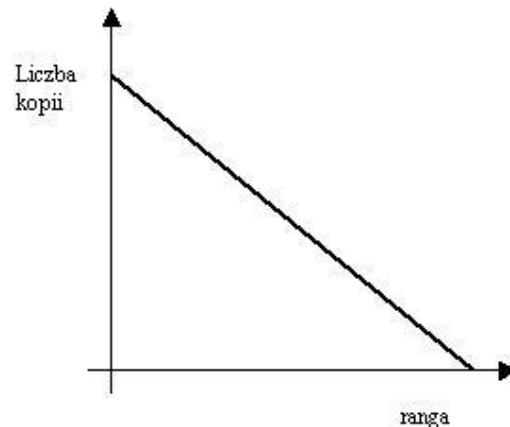
Osobniki populacji są ustawiane kolejno w zależności od wartości ich funkcji przystosowania.

Powstaje „lista rankingowa” od najlepiej do najgorzej przystosowanych osobników (lub odwrotnie).

Każdemu osobnikowi jest przypisana liczba określająca jego pozycję na liście (ranga).

Liczba kopii każdego osobnika wprowadzana do populacji rodzicielskiej jest ustalana zgodnie z wcześniej zdefiniowaną funkcją i zależy od rangi.

Przykład funkcji:



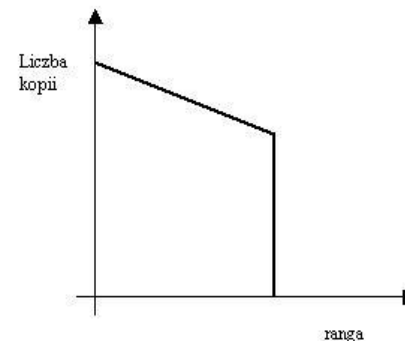
Metody selekcji

Selekcja turniejowa

Dzieli się osobniki na grupy a następnie z każdej grupy wybiera się osobnika o najlepszym przystosowaniu. Podgrupy mogą być dowolnego rozmiaru (najczęściej 2 lub 3 osobniki).

Selekcja progowa

Modyfikacja selekcji rankingowej, w której funkcja określająca prawdopodobieństwo przejścia osobnika do puli rodzicielskiej ma postać progu. Przykładowa funkcja:



Algorytm genetyczny – przykład 1

Szukanie maksimum funkcji $y=2x+1$ dla $x \in [0,31]$

x – parametr zadania.

Zbiór $\{0,1,2,\dots,31\}$ – przestrzeń poszukiwań a jednocześnie zbiór potencjalnych rozwiązań zadania

Rozwiązania kodujemy binarnie za pomocą 5 bitów.

Ciągi kodowe to chromosomy a w tym przypadku także genotypy.

Jako funkcję przystosowania przyjmujemy $y=2x+1$

Algorytm genetyczny – przykład 1

1. Losujemy populację początkową

W wyniku losowania otrzymujemy:

$$ch_1 = [00110]$$

$$ch_2 = [00101]$$

$$ch_3 = [01101]$$

$$ch_4 = [10101]$$

$$ch_5 = [11010]$$

$$ch_6 = [10010]$$

$$ch_7 = [01000]$$

$$ch_8 = [00101]$$

$$ch_1^* = 6$$

$$ch_2^* = 5$$

$$ch_3^* = 13$$

$$ch_4^* = 21$$

$$ch_5^* = 26$$

$$ch_6^* = 18$$

$$ch_7^* = 8$$

$$ch_8^* = 5$$

co odpowiada fenotypom:

Algorytm genetyczny – przykład 1

2. Obliczamy funkcję przystosowania

$$F(\text{ch}_1) = 2 \cdot \text{ch}_1^* + 1 = 13$$

$$F(\text{ch}_2) = 11$$

$$F(\text{ch}_3) = 27$$

$$F(\text{ch}_4) = 43$$

$$F(\text{ch}_5) = 53$$

$$F(\text{ch}_6) = 37$$

$$F(\text{ch}_7) = 17$$

$$F(\text{ch}_8) = 11$$

$$\text{Suma} = 212$$

Algorytm genetyczny – przykład 1

3. Selekcja chromosomów – koło ruletki

Na podstawie wzorów:

Wielkość procentowa sektorów:

$$v(ch_i) = p_s(ch_i) \cdot 100\%$$

Prawdopodobieństwo selekcji chromosomu:

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

obliczamy wycinki koła ruletki wyrażone w procentach:

$$v(ch_1) = (13/212) \cdot 100\% = 6,13\%$$

$$v(ch_2) = 5,19\%$$

$$v(ch_3) = 12,74\%$$

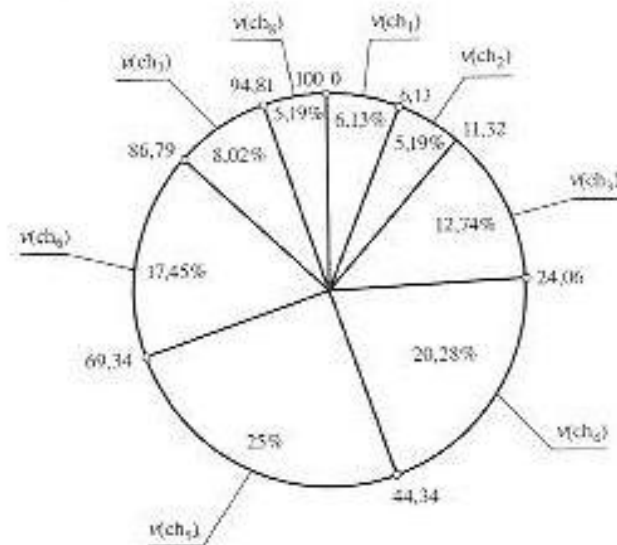
$$v(ch_4) = 20,28\%$$

$$v(ch_5) = 25\%$$

$$v(ch_6) = 17,45\%$$

$$v(ch_7) = 8,02\%$$

$$v(ch_8) = 5,19\%$$



Algorytm genetyczny – przykład 1

3. Selekcja chromosomów – koło ruletki

Za pomocą koła ruletki losujemy 8 nowych chromosomów.

Założmy, że wylosowano następujące liczby:

79 44 9 74 45 86 48 23

Oznacza to wybór następujących chromosomów:

ch_6 ch_4 ch_2 ch_6 ch_5 ch_6 ch_5 ch_3

Te chromosomy tworzą pulę rodzicielską.

Algorytm genetyczny – przykład 1

4. Operacje genetyczne

Założmy, że wylosowano prawdopodobieństwo mutacji $p_m=0,2$ i prawdopodobieństwo krzyżowania $p_k=0,75$

Krzyżowanie:

Kojarzymy osobniki w pary tak jak są ułożone w puli rodzicielskiej. Dla każdej pary losujemy liczbę z przedziału $[0,1]$. Jeśli dana liczba będzie mniejsza od $p_k=0,75$ to nastąpi krzyżowanie. Założmy, że wylosowano: 0,12 0,73 0,65 0,95.

Oznacza to, że trzy pierwsze pary zostaną poddana krzyżowaniu a czwarta para nie.

Dodatkowo dla każdej pary podlegającej krzyżowaniu losujemy punkt krzyżowania (liczba całkowita z przedziału $[1,4]$).

Algorytm genetyczny – przykład 1

4. Operacje genetyczne - krzyżowanie

Po operacji krzyżowania otrzymujemy następującą populację potomków o fenotypach

$$ch_1 = [10001]$$

$$ch_2 = [10110]$$

$$ch_3 = [00100]$$

$$ch_4 = [10011]$$

$$ch_5 = [11010]$$

$$ch_6 = [10010]$$

$$ch_7 = [11010]$$

$$ch_8 = [01101]$$

$$ch_1^* = 17$$

$$ch_2^* = 22$$

$$ch_3^* = 4$$

$$ch_4^* = 19$$

$$ch_5^* = 26$$

$$ch_6^* = 18$$

$$ch_7^* = 26$$

$$ch_8^* = 13$$

Algorytm genetyczny – przykład 1

4. Operacje genetyczne - mutacja

Dla każdego osobnika po krzyżowaniu losujemy liczbę z zakresu od 0 do 1. Załóżmy, że wylosowano

$ch_1=[10001]$	0,56
$ch_2=[10110]$	0,15
$ch_3=[00100]$	0,48
$ch_4=[10011]$	0,59
$ch_5=[11010]$	0,06
$ch_6=[10010]$	0,89
$ch_7=[11101]$	0,39
$ch_8=[01010]$	0,76

Algorytm genetyczny – przykład 1

4. Operacje genetyczne – mutacja

Mutacji podlegają te osobniki, dla których wylosowana liczba jest większa niż prawdopodobieństwo mutacji $p_m = 0,2$. Dla osobników podlegających mutacji losujemy miejsce mutacji, liczbę całkowitą z zakresu $[1,5]$

ch ₁ =[10001]	0,56	NIE
ch ₂ =[10110]	0,15	TAK I=3
ch ₃ =[00100]	0,48	NIE
ch ₄ =[10011]	0,59	NIE
ch ₅ =[11010]	0,06	TAK I=2
ch ₆ =[10010]	0,89	NIE
ch ₇ =[11101]	0,39	NIE
ch ₈ =[01010]	0,76	NIE

Algorytm genetyczny – przykład 1

4. Operacje genetyczne - mutacja

Po operacji mutacji otrzymujemy następującą populację potomków o fenotypach

$$ch_1 = [10001]$$

$$ch_2 = [10010]$$

$$ch_3 = [00100]$$

$$ch_4 = [10011]$$

$$ch_5 = [10010]$$

$$ch_6 = [10010]$$

$$ch_7 = [11010]$$

$$ch_8 = [01101]$$

$$ch_1^* = 17$$

$$ch_2^* = 18$$

$$ch_3^* = 4$$

$$ch_4^* = 19$$

$$ch_5^* = 18$$

$$ch_6^* = 18$$

$$ch_7^* = 26$$

$$ch_8^* = 13$$

Algorytm genetyczny – przykład 1

5. Obliczamy funkcje przystosowania dla nowej populacji

$$F(ch_1) = 2 \cdot ch_1^* + 1 = 35$$

$$F(ch_2) = 37$$

$$F(ch_3) = 9$$

$$F(ch_4) = 39$$

$$F(ch_5) = 37$$

$$F(ch_6) = 37$$

$$F(ch_7) = 59$$

$$F(ch_8) = 27$$

$$\text{Suma} = 280$$

Strategie ewolucyjne

Tak jak algorytmy genetyczne operują na populacjach potencjalnych rozwiązań i korzystają z zasad selekcji i przetwarzania osobników najlepiej przystosowanych.

Działają na wektorach liczb zmiennoprzecinkowych a nie binarnych. W procedurze selekcji tworzona jest tymczasowa populacja której wielkość różni się od populacji rodzicielskiej. Kolejna generacja osobników powstaje przez wybór najlepszych osobników.

Osobniki rodzicielskie wybierane są bez powtórzeń.

Najpierw osobniki podlegają krzyżowaniu i mutacji a potem następuje selekcja z powstałej populacji tymczasowej. Wybiera się tyle najlepszych osobników ile było rodziców.

Parametry takie jak prawdopodobieństwo krzyżowania i mutacji mogą się zmieniać w czasie trwania algorytmu.

Strategia ewolucyjna (1+1)

Przetwarzany jest jeden chromosom bazowy x , którego wartość początkowa jest ustalana losowo

W każdej generacji w wyniku mutacji powstaje nowy osobnik y

Po porównaniu funkcji przystosowania $F(x)$ i $F(y)$ wybierany jest lepszy osobnik, który zostaje nowym osobnikiem bazowym x

W algorytmie nie występuje operator krzyżowania

Chromosom y powstaje przez dodanie do każdego genu chromosomu x pewnej liczby losowej generowanej zgodnie z rozkładem normalnym.

Strategia ewolucyjna ($\mu+\lambda$)

Algorytm rozpoczyna się od wylosowania początkowej populacji rodzicielskiej P składającej się z μ osobników

Tworzy się populacja tymczasowa T zawierająca λ osobników ($\lambda \geq \mu$). Populacja ta powstaje poprzez losowanie λ osobników z populacji P (losowanie ze zwracaniem)

Osobniki z populacji T podlegają krzyżowaniu i mutacji i w ten sposób powstaje populacja potomna O zawierająca λ osobników

Z obu populacji $P \cup O$ wybieramy μ najlepszych osobników, które tworzą nową populację rodzicielską P .

Strategia ewolucyjna (μ, λ)

Algorytm rozpoczyna się od wylosowania początkowej populacji rodzicielskiej P składającej się z μ osobników

Tworzy się populacja tymczasowa T zawierająca λ osobników ($\lambda \geq \mu$). Populacja ta powstaje poprzez losowanie λ osobników z populacji P (losowanie ze zwracaniem)

Osobniki z populacji T podlegają krzyżowaniu i mutacji i w ten sposób powstaje populacja potomna O zawierająca λ osobników

Z populacji O wybieramy μ najlepszych osobników, które tworzą nową populację rodzicielską P .