

www.kwmimkm.polisi.pl

METODY HEURYSTYCZNE

wykład 3

www.kwmimkm.polisi.pl

ALGORYTMY GENETYCZNE

www.kwmimkm.polisi.pl

SCHEMAT DZIAŁANIA AG:

```

procedure algorytm_genetyczny
begin
  t:=0
  wybierz populację początkową P(t)
  oceń P(t)
  while (not warunek_zakończenia) do
  begin
    t:=t+1
    wybierz P(t) z P(t-1) (selekcja)
    zmień P(t) (działanie operatorów genetycznych)
    oceń P(t)
  end
end
  
```

www.kwmimkm.polisi.pl

METODY ANALITYCZNE kontra AG

METODY ANALITYCZNE

ZALETY	WADY
☺ ścisłe rozwiązanie	☹ Funkcja celu musi być ciągła
☺ wysoka szybkość działania	☹ Hessian funkcji celu musi być dodatnio określony
	☹ Istnieje duże ryzyko zbiegnięcia się algorytmu do optimum lokalnego
	☹ Obliczenia rozpoczynają się z jednego punktu ograniczając obszar poszukiwań optimum
	☹ Wybór punktu startowego wpływa na zbieżność metody

www.kwmimkm.polisi.pl

METODY ANALITYCZNE kontra AG

ALGORYTMY GENETYCZNE

ZALETY	WADY
☺ jedyną informacją potrzebną do działania jest wartość funkcji celu	☹ Stosunkowo wolne
☺ praca na populacji dopuszczalnych rozwiązań	☹ Trudności z precyzyjnym znalezieniem optimum
☺ przeszukiwanie wielokierunkowe	

www.kwmimkm.polisi.pl

Przykład:

Znaleźć $\max \{ f(x)=x^2 \}$

dla wartości całkowitych x z zakresu 0-31.

Populacja w chwili t :

$$P(t) = \{x^t_1, \dots, x^t_n\}$$

Założenia:

- łańcuchy 5-bitowe ($x=0,1,\dots,31$);
- liczebność populacji $n=4$

Ścisłe rozwiązanie:

www.kwmimkm.polisi.pl

$$x = \begin{matrix} 1 & 1 & 1 & 1 & 1 \end{matrix}$$

$$x = 31;$$

$$x^2 = 961.$$

Populacja początkowa (losowanie):

$$x^0_1 = \begin{matrix} 1 & 1 & 0 & 0 & 0 \end{matrix}$$

$$x^0_2 = \begin{matrix} 0 & 1 & 0 & 1 & 1 \end{matrix}$$

$$x^0_3 = \begin{matrix} 0 & 1 & 0 & 0 & 0 \end{matrix}$$

$$x^0_4 = \begin{matrix} 1 & 0 & 0 & 1 & 1 \end{matrix}$$

7

Sytuacja początkowa:

www.kwmimkm.polisi.pl

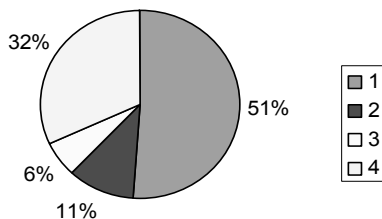
Nr osobnika	Osobnik	Wartość x	Przystosowanie $f(x)=x^2$	Prawd. wylosowania osobnika $p_i = \frac{f_i}{\sum f_i}$	Oczekiwana liczba kopii f_i / \bar{f}
1	11000	24	576	0.51	2.05
2	01011	11	121	0.11	0.43
3	01000	8	64	0.06	0.23
4	10011	19	361	0.32	1.27
Suma			1122	1.00	4.00
Średnia			281	0.25	1.00
Max			576	0.51	2.05

8

Selekcja:

www.kwmimkm.polisi.pl

Każdemu ciągowi kodowemu odpowiada sektor koła ruletki o polu proporcjonalnym do przystosowania.



$\langle 0, 0.51 \rangle \Rightarrow 1, \quad \langle 0.51, 0.62 \rangle \Rightarrow 2, \dots$

9

Po selekcji:

www.kwmimkm.polisi.pl

Nr osobnika	Oczekiwana liczba kopii	Liczba wylosowanych kopii	Osobnik po selekcji	Wartość x	Przystosowanie $f(x)=x^2$	Prawd. wylosowania osobnika
1	2.05	2	11000	24	576	0.35
2	0.43	1	01011	11	121	0.07
3	0.23	0	11000	24	576	0.35
4	1.27	1	10011	19	361	0.23
Suma					1634	1.0
Średnia					409	0.25
Max					576	0.35 (x2)

10

Krzyżowanie:

www.kwmimkm.polisi.pl

$$p_c = 0.9$$

Nr osobnika	Osobnik po selekcji	Partner	Punkt krzyżowania	Osobnik po krzyżowaniu	Wartość x	Przystosowanie $f(x)=x^2$	Prawd. wylosowania osobnika
1	11000	2	4	11001	25	625	0.36
2	01011	1	4	01010	10	100	0.06
3	11000	4	2	11011	27	729	0.43
4	10011	3	2	10000	16	256	0.15
Suma						1710	1.0
Średnia						428	0.25
Max						729	0.43

było: 576

11

Mutacja:

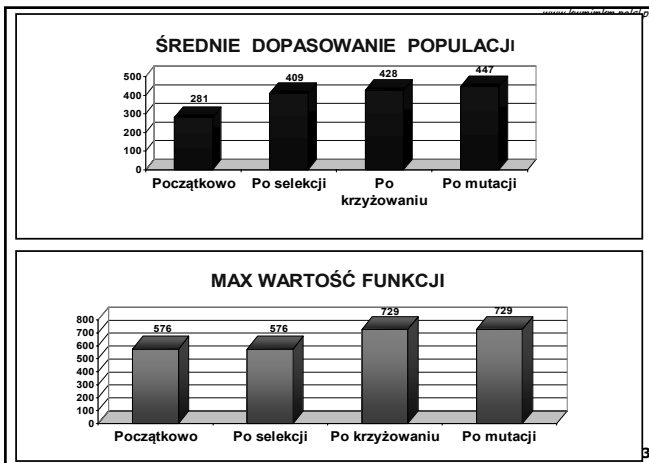
www.kwmimkm.polisi.pl

$$p_m = 0.05$$

Nr osobnika	Osobnik po krzyżowaniu	Mutacja?	Osobnik po mutacji	Wartość x	Przystosowanie $f(x)=x^2$	Prawd. wylosowania osobnika
1	11001	NNNN	11001	25	625	0.35
2	01010	NNNN	01010	10	100	0.06
3	11011	NNNN	11011	27	729	0.41
4	10000	NNNT	10010	18	324	0.18
Suma					1778	1.0
Średnia					447	0.25
Max					729	0.41

było: 729

12



www.kwmimkm.polsl.pl

Rozpatrywać można zawsze zadanie maksymalizacji:

$$g(x) = -f(x)$$

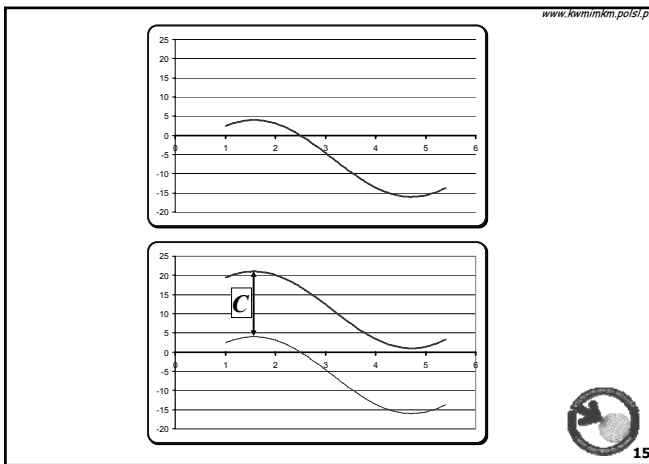
$$\min f(x) = \max g(x) = \max \{-f(x)\}$$

Zakłada się również, iż funkcja jest dodatnia w całej dziedzinie (selekcja!)

$$\max g(x) = \max \{g(x) + C\}$$

(jeśli $g(x)$ jest ograniczona z dołu)

14



www.kwmimkm.polsl.pl

Wielkości zadawane przed uruchomieniem AG:

- liczebność populacji,
- prawdopodobieństwo krzyżowania,
- prawdopodobieństwo mutacji,
- inne (zależą od algorytmu).

Zazwyczaj wartości dobiera się eksperymentalnie (metodą prób i błędów), indywidualnie dla rozwiązywanego problemu...

choć istnieją pewne ogólne zalecenia...

16

www.kwmimkm.polsl.pl

KODOWANIE LICZB RZECZYWISTYCH

Założenia:

- $f(x_1, \dots, x_k): R_k \rightarrow R$
- $D_i = [a_i, b_i] \subseteq R$
- $f(x_1, \dots, x_k) > 0$ dla każdego $x_i \in D_i$
- dokładność do c liczb znaczących po przecinku

Wykonanie:

1. Podział $D_i = [a_i, b_i]$ na $r = (b_i - a_i) \cdot 10^c$ podprzedziałów.
2. Wyznaczenie najmniejszej liczby całkowitej m :

$$(b_i - a_i) \cdot 10^c \leq 2^m - 1$$

17

www.kwmimkm.polsl.pl

ODKODOWYWANIE:

1. Przekształcenie łańcucha binarnego o długości m na liczbę dziesiętną x' ;
2. Obliczenie rzeczywistej wartości liczby:

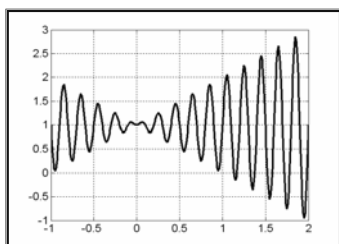
$$x = a_i + \frac{(b_i - a_i)x'}{2^m - 1}$$

18

Przykład:

$$f(x) = x \cdot \sin(10\pi x) + 1$$

$$\max\{f(x)\}$$



- Dziedzina funkcji: $x \in [-1, 2]$
- Liczba miejsc po przecinku: $c = 6$

Na ilu bitach trzeba zakodować liczbę (wyznaczenie m)?



$$a_i = -1; b_i = 2; c = 6$$

liczba podprzedziałów:

$$r = (b_i - a_i) \cdot 10^c = 3 \cdot 10^6$$

$$3 \cdot 10^6 \leq 2^m - 1$$

$$3\,000\,001 \leq 2^m$$

$$2\,097\,152 = 2^{21} \leq 3\,000\,001 \leq 2^{22} = 4\,194\,304$$

$$m = 22$$



Mając ciąg bitów:

1100101010010101010101

Odkodowywanie:

1. Przekształcenie łańcucha na liczbę dziesiętną x' :

$$x' = 3319125$$

2. Obliczenie rzeczywistej wartości liczby:

$$x = a_i + \frac{(b_i - a_i)x'}{2^m - 1} = -1 + \frac{3 \cdot 3319125}{4194304 - 1}$$

$$= 1.374025$$

$$a_i = -1; b_i = 2; c = 6$$



00000000000000000000

11111111111111111111



$$x' = 0$$



$$x' = 4194303$$



$$x = -1 + \frac{0}{4194304 - 1} = -1$$



$$x = -1 + \frac{3 \cdot 4194303}{4194304 - 1} = 2$$

$$a_i = -1; b_i = 2$$

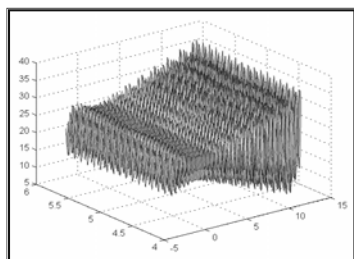


Przykład:

$$f(x) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2)$$

$$\max\{f(x_1, x_2)\}$$

- $x_1 \in [-3.0, 12.1]$
- $x_2 \in [4.1, 5.8]$
- $c_1 = c_2 = 4$



x_1 :

$$r_1 = (b_1 - a_1) \cdot 10^c = 15.1 \cdot 10^4$$

$$15.1 \cdot 10^4 \leq 2^m - 1$$

$$131\,032 = 2^{17} \leq 151\,001 \leq 2^{18} = 262\,144$$

$$m_1 = 18$$

x_2 :

$$r_2 = (b_2 - a_2) \cdot 10^c = 1.7 \cdot 10^4$$

$$1.7 \cdot 10^4 \leq 2^m - 1$$

$$16\,384 = 2^{14} \leq 17\,001 \leq 2^{15} = 32\,768$$

$$m_2 = 15$$

$$m = m_1 + m_2 = 33$$

11001010100101011001110101010101



Jeden z celów zmodyfikowanego kodowania:

- przybliżenie algorytmu do przestrzeni zadania.

Dogodne jest, by dwa punkty leżące blisko siebie w przestrzeni reprezentacji (genotyp) leżały również blisko siebie w przestrzeni zadania (fenotyp).

(Nie zawsze prawdziwe przy kodowaniu binarnym)

np.:

Binarnie	Całkowitoliczbowo
0111	7
1000	8



KOD GRAYA

procedure GrayToBin

```

begin
  value := g1
  b1 := value
  for k := 2 to m do
    begin
      if gk = 1 then value := NOT value
      bk := value
    end
  end
end

```

procedure BinToGray

```

begin
  g1 := b1
  for k := 2 to m do
    gk := bk-1 XOR bk
  end

```

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

$b = \langle b_1, b_2, \dots, b_m \rangle$ – liczba binarna

$g = \langle g_1, g_2, \dots, g_m \rangle$ – liczba w kodzie Graya;

m – długość ciągu kodowego.



Binarnie	Kod Graya
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

Zmiana 1 bitu w kodzie powoduje, iż otrzymana liczba ma szansę być liczbą bezpośrednio bliższą liczbie przed zmianą.



Jednakże jeżeli:

- 100 zmiennych;
- dziedzina z zakresu [-500 500];
- żądana dokładność 6 miejsc po przecinku;

To:

- długość łańcucha binarnego wynosi **3000**;
- przestrzeń poszukiwań rzędu **10¹⁰⁰⁰**.

Dla tak wielkich przestrzeni AG działają słabo...

Zasada minimalnego alfabetu:

Należy wybrać najmniejszy alfabet, w którym zadanie wyraża się w sposób naturalny.



KODOWANIE LOGARYTMICZNE

Stosowane w celu zmniejszenia długości łańcucha binarnego.

$$[b_1 b_2 \text{ bin}] = (-1)^{b_2} e^{(-1)^{b_1} [\text{bin}]_{10}}$$

b_1 – bit znaku wykładnika funkcji wykładniczej;

b_2 – bit znaku funkcji wykładniczej;

bin – reprezentacja wykładnika funkcji wykładniczej

$[\text{bin}]_{10}$ – wartość dziesiętna liczby zakodowanej binarnie.

$$[10110] = (-1)^0 e^{(-1)^1 [110]_{10}} = e^{-6} = 0.002478752$$

$$[01011] = (-1)^1 e^{(-1)^0 [011]_{10}} = -e^3 = -20.08553692$$



Za pomocą 5 bitów możliwe jest zakodowanie liczb z zakresu $[-e^7, e^7]$

(w kodowaniu binarnym $[0, 31]$).

Dalszą modyfikacją jest zastosowanie

KODOWANIA ZMIENNOPOZYCYJNEGO.



Modyfikacje:

- łańcuchy o zmiennej długości;
- struktury bogatsze od łańcuchów (np. macierze);
- zmodyfikowane operatory;
- nowe operatory (inwersja, klonowanie, itp.)
- inna niż binarna reprezentacja zadania;
- „pamięć” chromosomu;
- ...

„zmieniony AG”, „ulepszony AG”, „zmodyfikowany AG”,

...



31

Różnorodne programy opierające się na zasadzie ewolucji mogą się różnić:

- strukturą danych;
- operatorami;
- metodami tworzenia populacji początkowej;
- sposobami uwzględniania ograniczeń zadania;
- parametrami.

Zasada działania nie zmienia się:

populacja osobników podlega pewnej transformacji, zaś osobniki starają się przetrwać w procesie ewolucji.



32

ALGORYTMY EWOLUCYJNE



33

Dla danego problemu można określić wiele sposobów kodowania i zdefiniować szereg operatorów (np. zadanie komiwojażera).

AE to rozwinięcie i uogólnienie AG

Należy jednoznacznie określić:

- schemat działania AE;
- metodę selekcji;
- sposób kodowania;
- operatory genetyczne;
- środowisko działania AE.



34

procedure Algorytm_Ewolucyjny

begin

$t:=0$

wybierz populację początkową $P(t)$

ocień $P(t)$

while (not warunek_zakończenia) **do**

begin

wybierz $T(t)$ z $P(t)$ (reprodukcja)

utwórz $O(t)$ z $T(t)$ (działanie operatorów ewolucyjnych)

ocień $O(t)$

utwórz $P(t+1)$ z $O(t)$ i $P(t)$ (sukcesja)

$t:=t+1$

end

end

T – temporary
 O – offspring



35

SELEKCJA

=

REPRODUKCJA

(preselekcja)

+

SUKCESJA

(postselekcja)

procedure Algorytm_Ewolucyjny

begin

$t:=0$

wybierz populację początkową $P(t)$

ocień $P(t)$

while (not warunek_zakończenia) **do**

begin

wybierz $T(t)$ z $P(t)$ (reprodukcja)

utwórz $O(t)$ z $T(t)$ (operator)

ocień $O(t)$

utwórz $P(t+1)$ z $O(t)$ i $P(t)$ (sukcesja)

$t:=t+1$

end

end

Reprodukcja – tworzenie populacji tymczasowej $T(t)$, która jest poddawana działaniu operatorów genetycznych tworząc populację potomną $O(t)$.

Sukcesja – tworzenie nowej populacji bazowej $P(t+1)$ z populacji potomnej $O(t)$ oraz starej populacji bazowej $P(t)$.



36

Superosobniki:

- Niepożądane w początkowej fazie działania (*przedwczesna zbieżność*).
- Pozytywne pod koniec pracy algorytmu (*zawężenie przestrzeni poszukiwań*).



Napór selekcyjny (selektywny nacisk):

tendencja algorytmu do poprawiania wartości średniej przystosowania.

Algorytm charakteryzuje się tym większym naporem selekcyjnym, im większa jest oczekiwana liczba kopii lepszego osobnika w porównaniu z oczekiwaną liczbą kopii osobnika gorszego.

37

- Twarda (brutalna) selekcja – wybór do populacji potomnej i powielanie tylko najlepszego osobnika (metoda stochastycznego wzrostu).
- Przyjęcie jednakowego prawdopodobieństwa - algorytm błędzi przypadkowo (brak selekcji).

W AE - metoda pośrednia, zwana „miękką selekcją”.



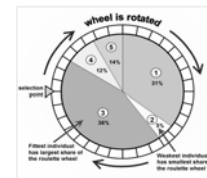
38

METODY REPRODUKCJI I SUKCESJI



39

KOŁO RULETKI



– jak w AG...

SELEKCJA TURNIEJOWA

Wybór k osobników (*rozmiar turnieju, zwykle $k=2$*) i selekcja najlepszego z grupy.

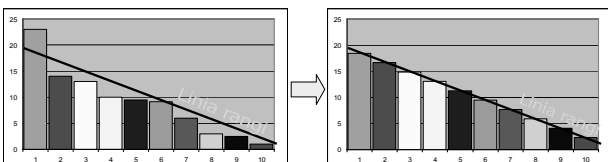
- Powtarzane pop_size razy.



40

SELEKCJA RANKINGOWA

Szeregowanie osobników według wartości przystosowania i selekcja zgodnie z kolejnością (wg tzw. *linii rangi*):



- ⊕ Zapobiega powstawaniu supersobników.
- ⊖ Pomija informację o względnych ocenach osobników.



41

SUKCESJA TRYWIALNA

(z całkowitym zastępowaniem)

Nową populacją bazową staje populacja potomna

$$P(t+1) = O(t)$$

(jak w AG).

- ⊕ Najbardziej odporna na przedwczesną zbieżność.
- ⊖ Najwolniej prowadzi do rozwiązania optymalnego.
- ⊖ Może powodować, że najlepsze rozwiązania z populacji $P(t)$ nie znajdują się w populacji $P(t+1)$.



42

SUKCESJA Z CZĘŚCIOWYM ZASTĘPOWANIEM

W nowej populacji bazowej są osobniki z populacji potomnej i ze starej populacji bazowej:

$$P(t+1) = O(t) + P(t)$$

- ⊕ Prowadzi zwykle do stabilniejszej pracy AE.
- ⊕ Może spowodować tendencję do osiągnięcia maksimum lokalnych.

Mechanizm usuwania (warianty):

- usuwanie najgorzej przystosowanych osobników;
- usuwanie osobników podobnych do potomnych;
- usuwanie losowo wybranych osobników.



43

SUKCESJA ELITARNA

Gwarantuje przeżycie co najmniej najlepszego osobnika poprzez odpowiedni wybór osobników z $P(t)$ do $P(t+1)$

- ⊕ Wzrost wielkości elity powoduje przyspieszenie zbieżności algorytmu.
- ⊕ Wzrost wielkości elity powoduje większe prawdopodobieństwo osiągnięcia ekstremów lokalnych.
- Wartość wielkości elity δ decyduje o naporze selekcyjnym ($\delta=0$ – sukcesja trywialna).
- Najkorzystniej – jeden, ew. kilka osobników).



44

TYPOWE OPERATORY KRZYŻOWANIA



45

Para rodziców – para potomków

Zwykle:

2 osobniki rodzicielskie dają 2 (sprzężone) osobniki potomne.

Pojedynczy osobnik potomny

- wariant dwuosobniczy – para osobników rodzicielskich;
- wariant globalny – jeden wiodący i n pomocniczych osobników rodzicielskich (po jednym dla każdego genu).

Krzyżowanie wieloosobnicze:

- z wieloma osobnikami potomnymi;
- z jednym osobnikiem potomnym.



46

OPERATORY KRZYŻOWANIA WYMIENIAJĄCEGO

- Tworzą chromosomy potomne przez składanie ich z wartości genów chromosomów rodzicielskich.
- Mogą być wykorzystywane zarówno przy kodowaniu binarnym, jak i rzeczywistoliczbowym.
- Nie dochodzi do modyfikacji wartości genów zawartych w chromosomach krzyżowanych osobników rodzicielskich (tylko ich przetasowanie).



47

KRZYŻOWANIE JEDNOPUNKTOWE (proste)

- wybór (z rozkładem jednostajnym) liczby c (punkt rozcięcia) ze zbioru $\{1, 2, \dots, n-1\}$
 n - długość osobnika;

- Podział chromosomów X^1 i X^2 poddawanych krzyżowaniu na dwie części i ich sklejanie:

$$Y = [X^1_1, \dots, X^1_c, X^2_{c+1}, \dots, X^2_n].$$

W wersji z 2 osobnikami potomnymi drugi potomek:

$$Z = [X^2_1, \dots, X^2_c, X^1_{c+1}, \dots, X^1_n]$$



48

www.kwmimkm.polisi.pl

X^1_i	X^2_i		Y_i	Z_i
3.24	2.22		3.24	2.22
-0.22	3.14		-0.22	3.14
1.32	7.72	c	1.32	7.72
3.22	1.22		1.22	3.22
1.20	2.40		2.40	1.20
7.23	4.28		4.28	7.23
-2.21	-2.42		-2.42	-2.21

49

www.kwmimkm.polisi.pl

KRZYŻOWANIE DWUPUNKTOWE

- wybór 2 punktów rozcięcia c_1 i c_2 ;
- Podział chromosomów X^1 i X^2 poddawanych krzyżowaniu na 3 części i wymiana środkowej części:

$$Y = [X^1_1, \dots, X^1_{c_1}, X^2_{c_1+1}, \dots, X^2_{c_2}, X^1_{c_2+1}, \dots, X^1_n]$$

W wersji z 2 osobnikami potomnymi drugi potomek:

$$Z = [X^2_1, \dots, X^2_{c_1}, X^1_{c_1+1}, \dots, X^1_{c_2}, X^2_{c_2+1}, \dots, X^2_n]$$

$c_1 = c_2$ – krzyżowanie jednopunktowe

50

www.kwmimkm.polisi.pl

X^1_i	X^2_i		Y_i	Z_i
3.24	2.22		3.24	2.22
-0.22	3.14		-0.22	3.14
1.32	7.72	c_1	1.32	7.72
3.22	1.22		1.22	3.22
1.20	2.40		2.40	1.20
7.23	4.28	c_2	4.28	7.23
-2.21	-2.42		-2.21	-2.42

51

www.kwmimkm.polisi.pl

KRZYŻOWANIE RÓWNOMIERNE

Chromosom potomny:

jeśli wylosowano liczbę $< p_e$;
w przeciwnym razie.

p_e – parametr krzyżowania (typowo $p_e=0.5$)

W wersji z 2 osobnikami potomnymi drugi potomek:

jeśli $Y_i = X^1_i$;
w przeciwnym razie.

52

p_e=0.5

www.kwmimkm.polisi.pl

X^1_i	X^2_i	wylosowano	Y_i	Z_i
3.24	2.22	0.092699	3.24	2.22
-0.22	3.14	0.158384	-0.22	3.14
1.32	7.72	0.697190	7.72	1.32
3.22	1.22	0.315814	3.22	1.22
1.20	2.40	0.821422	2.40	1.20
7.23	4.28	0.399981	7.23	4.28
-2.21	-2.42	0.428556	-2.21	-2.42

53

www.kwmimkm.polisi.pl

KRZYŻOWANIE DIAGONALNE

- Jest krzyżowaniem wieloosobniczym.
- Tworzy r potomków z r rodziców przy $c = r - 1$ punktach krzyżowania.
- Osobniki potomne powstają w wyniku składania fragmentów kodu po przekątnej.

Dla 3 osobników:

$$Y = [X^1_1, \dots, X^1_{c_1}, X^2_{c_1+1}, \dots, X^2_{c_2}, X^3_{c_2+1}, \dots, X^3_n]$$

$$Z = [X^2_1, \dots, X^2_{c_1}, X^3_{c_1+1}, \dots, X^3_{c_2}, X^1_{c_2+1}, \dots, X^1_n]$$

$$W = [X^3_1, \dots, X^3_{c_1}, X^1_{c_1+1}, \dots, X^1_{c_2}, X^2_{c_2+1}, \dots, X^2_n]$$

54

www.kwmimkm.polst.pl

W wersji 1 potomkiem – tylko potomek Y

55

OPERATORY KRZYŻOWANIA UŚREDNIAJĄCEGO

www.kwmimkm.polst.pl

- Są specyficzne dla kodowania rzeczywistoliczbowego;
- Oddziałują na wartości genów chromosomów poddawanych krzyżowaniu;
- Wartości każdego genu chromosomów potomnych są liczbami zawierającymi się między największą i najmniejszą wartością genu chromosomów rodzicielskich.

56

KRZYŻOWANIE ARYTMETYCZNE

www.kwmimkm.polst.pl

- generowanie liczby losowej k z zakresu $(0,1)$ lub jej arbitralny wybór;
- uśrednianie arytmetyczne wartości genów chromosomów rodzicielskich:

$$Y = X^1 + k(X^2 - X^1)$$

W wersji z 2 osobnikami potomnymi drugi potomek:

$$Z = X^2 + X^1 - Y^1$$

57

www.kwmimkm.polst.pl

X^1
4.0
-0.2
1
3.22
1.21
-3.42
-2.21

X^2
2.22
3.14
7.72
1.28
2.45
-5.14
2.43

$k=0.5$

Y
3.11
1.56
4.36
2.25
1.83
-4.28
0.11

Z
3.11
1.56
4.36
2.25
1.83
-4.28
0.11

58

KRZYŻOWANIE HEURYSTYCZNE

www.kwmimkm.polst.pl

- Nie jest krzyżowaniem uśredniającym!
- Generowanie liczby losowej k z zakresu $(0,1)$;
- Tworzy się (maksymalnie) jednego potomka:

$$Y = k(X^2 - X^1) + X^2$$

przy założeniu, że $X^2 \geq X^1$

- Może utworzyć potomka, który nie jest dopuszczalny, wówczas:
 - » generuje się nową liczbę losową i tworzy nowego potomka;
 - » jeśli po założonej liczbie prób nie utworzono osobnika dopuszczalnego, to nie tworzy się potomka.

59

www.kwmimkm.polst.pl

X^1
1.20
1.0
3.22
1.20

X^2
1.30
2.0
3.22
3.20

→

Y_i
1.31
1.1
3.22
3.40

60

TYPOWE OPERATORY MUTACJI

www.kwmimkm.polsl.pl



61

MUTACJA RÓWNOMIERNY

www.kwmimkm.polsl.pl

- Losowy wybór genu w chromosomie.
- Przyjęcie przez gen wartości losowej (z rozkładem równomiernym) z zakresu dopuszczalnego dla danej zmiennej:

$$Y = [X_1, \dots, X'_k, \dots, X_n],$$

$$X'_k = \langle \text{left}(k), \text{right}(k) \rangle$$

Szczególnie użyteczna we wczesnej fazie działania AE (gdy pożądane jest szerokie przeszukiwanie obszaru poszukiwań optimum).



62

MUTACJA NIERÓWNOMIERNY

www.kwmimkm.polsl.pl

- Należy do grupy tzw. mutacji ze strojeniem.
- Modyfikacja wartości wybranego genu o wartość pewnej funkcji $\Delta(t,y)$:

$$Y = [X_1, \dots, X'_k, \dots, X_n],$$

gdzie:

$$X'_k = X_k + \Delta(t, \text{right}(k) - X_k) \text{ gdy wylosowano } 0$$

$$X'_k = X_k - \Delta(t, X_k - \text{left}(k)) \text{ gdy wylosowano } 1$$



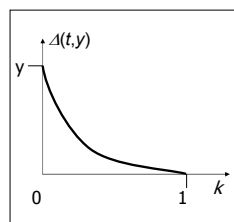
63

- Funkcja $\Delta(t,y)$ przyjmuje wartości z zakresu $[0,y]$;

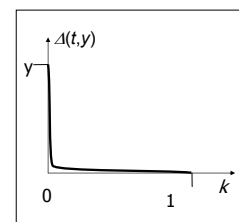
www.kwmimkm.polsl.pl

- Prawdopodobieństwo, że $\Delta(t,y)$ jest bliskie zero wzrasta ze wzrostem czasu obliczeń

(nie zależy jednak od zachowania się AE).



Początkowa faza obliczeń



Pod koniec działania AE



64

MUTACJA BRZEGOWA

www.kwmimkm.polsl.pl

- Jest odmianą mutacji równomiernej, w której:

$$X'_k = \text{left}(k) \text{ gdy wylosowano } 0$$

$$X'_k = \text{right}(k) \text{ gdy wylosowano } 1$$

- Szczególnie użyteczna, gdy rozwiązanie optymalne leży na brzegu obszaru dopuszczalnego lub bardzo blisko tego brzegu.



65

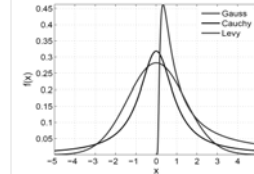
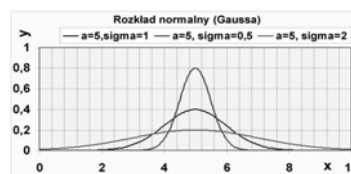
MUTACJA GAUSSOWSKA

www.kwmimkm.polsl.pl

- Przyjęcie przez wylosowany gen wartości losowej (z rozkładem Gaussa) o wartości oczekiwanej równej wartości przed zmianą:

$$Y = [X_1, \dots, X'_k, \dots, X_n],$$

$$X'_k = X_k + N(0, \sigma)$$



66