

Cel ćwiczenia:

1. Nabycie umiejętności wykorzystania algorytmów genetycznych do eksploracji danych

Środki:

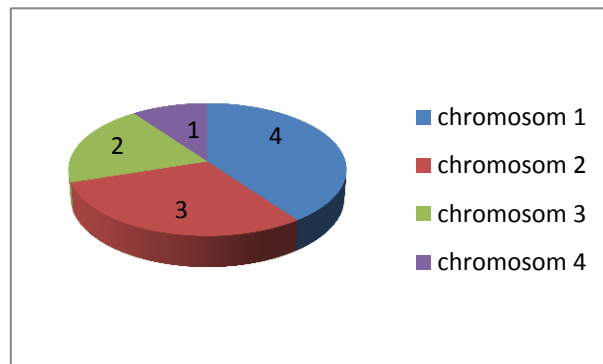
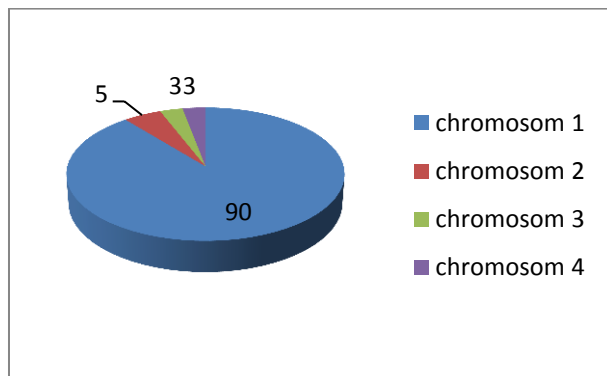
1. Serwer baz danych Oracle Enterprise Edition 11g.
2. Aplikacja SQL Developer.
3. Pakiet PL/SQL tworzony skryptem: ED_LAB_03_ag_pak.sql.

Przebieg:

1. Wgraj na swój schemat w bazie danych schemat obiekty ze skryptu ED_LAB_03_ag_tabs.sql.
2. Wgraj na swój schemat w bazie danych schemat pakiet ze skryptu ED_LAB_03_ag_pak.sql.
3. Przetestuj funkcjonowanie pakietu i przeanalizuj kolejne etapy jego działanie – wykonaj testy dla wariantów: selekcji proporcjonalnej i ze skalowanie poprzez sigma a także dla krzyżowanie jednopunktowe oraz równomierne.
4. Na podstawie wiedzy z wykładu (poniżej skrót) rozbuduj pakiet w następujący sposób – dobierz się dwuosobowe zespoły:
 - a. Jedna osoba rozbuduje pakiet o możliwość wyboru trzeciego wariantu selekcji: selekcji rankingowej.
 - b. Druga osoba w tym czasie rozbuduje pakiet o możliwość wyboru kodowania Graya.
5. Każda z osób przetestuje wpływ zastosowania swojej modyfikacji na uzyskiwane wyniki – średnią wartość funkcji dopasowania po wybranych liczbach pokoleń.
6. Następnie przetestować należy użycie obu dodanych możliwości jednocześnie i porównać wyniki.

Selekcja rankingowa (rank based selection)

- Ustawia chromosomy w kolejności o najlepszego do najgorszego, w zależności od wartości $f(x)$.
- Zabezpiecza to przed naciskiem selektywnym obecnym w tradycyjnej metodzie proporcjonalnej (z kołem ruletki).
- Prawdopodobieństwo selekcji nie zależy od pierwotnej (np. dużej) różnicy w wartości $f(x)$.



Kodowanie Graya:

System dziesiętny	Kod binarny	Kod Graya
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Liczba binarna $b = \langle b_1, \dots, b_m \rangle$

Liczba w kodzie Graya $g = \langle g_1, \dots, g_m \rangle$

procedure Binary-to-Gray

begin

$g_1 = b_1$

for $k = 2$ **to** m **do**

$g_k = b_{k-1} \text{ XOR } b_k$

end

procedure Gray-to-Binary

begin

$value = g_1$

$b_1 = value$

for $k = 2$ **to** m **do**

begin

if $g_k = 1$ **then** $value = NOT\ value$

$b_k = value$

end

end