

# SZTUCZNA INTELIGENCJA

## WYKŁAD 18. ALGORYTMY EWOLUCYJNE - ZASTOSOWANIA

Dr hab. inż. Grzegorz Dudek  
Wydział Elektryczny  
Politechnika Częstochowska

Częstochowa 2014

Zadanie załadunku (plecakowe) [Michalewicz96, str. 110] –

Dany jest zbiór  $n$  artykułów. Każdy artykuł ma objętość  $w(i)$  oraz przydatność  $p(i)$ . Należy wybrać podzbiór artykułów, tak aby suma ich objętości nie przekroczyła pojemności plecaka  $C$  i aby suma ich przydatności była maksymalna:

$$\sum_{i=1}^n x(i)w(i) \leq C$$

$$P(\mathbf{x}) = \sum_{i=1}^n x(i)p(i) \rightarrow \max$$

gdzie  $\mathbf{x} = [x(1), x(2), \dots, x(n)]$  – wektor binarny; 1 oznacza, że  $i$ -ty artykuł jest wybrany do plecaka, 0 – że nie jest wybrany.

Rozważa się następujące warianty:

- $w(i)$  jest nieskorelowane z  $p(i)$
- $w(i)$  jest słabo skorelowane z  $p(i)$
- $w(i)$  jest silnie skorelowane z  $p(i)$

Większe korelacje utrudniają rozwiązanie zadania.

Ponadto rozważa się zadanie:

- z ograniczoną pojemnością – plecak zawiera niewiele artykułów
- ze średnią pojemnością – plecak zawiera ok. połowy artykułów

Zaprogramowano i przetestowano trzy metody uwzględnienia ograniczeń funkcji celu:

- z funkcją kary
- z metodami naprawy
- z dekodernami

## Algorytm z funkcją kary

Chromosomy reprezentują łańcuchy binarne  $\mathbf{x}$ .

Funkcja przystosowania ma postać:

$$F(\mathbf{x}) = \sum_{i=1}^n x(i)p(i) - f_k(\mathbf{x}) \rightarrow \max$$

Przetestowano trzy rodzaje funkcji kary:

$$f_{k1}(\mathbf{x}) = \log_2 [1 + \rho(\sum_{i=1}^n x(i)w(i) - C)]$$

$$f_{k2}(\mathbf{x}) = \rho(\sum_{i=1}^n x(i)w(i) - C)$$

$$f_{k3}(\mathbf{x}) = [\rho(\sum_{i=1}^n x(i)w(i) - C)]^2$$

gdzie  $\rho = \max\{p(i)/w(i)\}$

### Algorytm z metodami naprawy

Chromosomy reprezentują łańcuchy binarne  $x$ .

Funkcja przystosowania ma postać:

$$F(\mathbf{x}) = \sum_{i=1}^n x'(i)p(i) \rightarrow \max$$

gdzie  $x'$  jest naprawioną wersją oryginalnego chromosomu  $x$ .

Naprawa polega na usuwaniu artykułów z plecaka, po kolei wg kryterium losowego lub deterministycznego, do momentu spełnienia ograniczenia.

Kryterium losowe brzmi: usuń losowo wybrany artykuł z plecaka.

Kryterium deterministyczne (zachłanne) brzmi: usuń z plecaka ten element, który ma najmniejszy stosunek przydatności do objętości.

Naprawione chromosomy mogą wymieniać chromosomy oryginalne z pewnym prawdopodobieństwem. Według badań eksperymentalnych wymiana 5% naprawionych osobników daje najlepsze efekty.

### Algorytm z dekodera

Stosuje się reprezentację całkowitoliczbową – porządkową.

Rozwiązanie jest reprezentowane przez chromosom  $\mathbf{x}$  jako ciąg  $n$  artykułów;  $i$ -ty element tego ciągu jest liczbą z zakresu od 1 do  $n-i+1$ .

Dana jest lista artykułów  $L$ . Artykuły na tej liście mogą być w porządku dowolnym lub w porządku malejącym względem stosunku przydatności do objętości.

Dekodowanie za pomocą wektora  $\mathbf{x}$  następuje przez wybór artykułów z listy  $L$ .

Np. dla  $L=(1,2,3,4,5,6)$  dekodowanie za pomocą chromosomu  $\mathbf{x} = [4, 3, 4, 1, 1, 1]$  daje następujący ciąg artykułów: 4, 3, 6, 1, 2, 5.

Rozwiązanie tworzymy w ten sposób, że do plecaka dołączamy kolejne artykuły do momentu jego zapelnienia.

W algorytmie z dekoderni można stosować krzyżowanie jednopunktowe dwóch rodziców (takie krzyżowanie daje dopuszczalnego potomka).

Mutacja polega na zmianie genu na wartość losową z zakresu  $[1, n-i+1]$ .

Wyniki i wnioski z działania tak zdefiniowanego AE dla tego zadania można znaleźć w [Michalewicz96, str. 114].

Zadanie transportowe [Michalewicz96, str. 215] –

Poszukuje się planu najtańszego transportu jednego towaru z  $n$  punktów nadania do  $k$  punktów odbioru. Dane są poziomy zapasów w każdym punkcie nadania  $s(i)$ , poziomy zapotrzebowania w każdym punkcie odbioru  $d(j)$  oraz jednostkowy koszt transportu między  $i$ -tym punktem nadania a  $j$ -tym punktem odbioru  $j - c(i,j)$

$$\sum_{i=1}^n \sum_{j=1}^k f_{ij}(x_{ij}) \rightarrow \min$$

$$\sum_{j=1}^k x_{ij} \leq s(i), \quad \text{dla } i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} \geq d(i), \quad \text{dla } j = 1, 2, \dots, k$$

$$x_{ij} \geq 0, \quad \text{dla } i = 1, 2, \dots, n; j = 1, 2, \dots, k$$

gdzie:  $x_{ij}$  – ilość towaru transportowanego z punktu  $i$  do  $j$ ;  $f_{ij}$  – koszt transportu towaru z punktu  $i$  do  $j$  (może być liniowy –  $c(i,j) \cdot x_{ij}$  lub nieliniowy).

## ZADANIE TRANSPORTOWE

Zakładamy, że mamy do czynienia z całkowitoliczbowym, zbilansowanym zadaniem transportowym – całkowite zapasy są równe całkowitemu zapotrzebowaniu.

### Reprezentacja macierzowa

|          |          | <i>j</i> |          |     |          |
|----------|----------|----------|----------|-----|----------|
|          |          | 1        | 2        | ... | <i>k</i> |
| <i>i</i> | 1        | $x_{11}$ | $x_{12}$ | ... | $x_{1k}$ |
|          | 2        | $x_{21}$ | $x_{22}$ | ... | $x_{2k}$ |
|          | ...      | ...      | ...      | ... | ...      |
|          | <i>n</i> | $x_{n1}$ | $x_{n2}$ | ... | $x_{nk}$ |

**Inicjalizacja** - tworzenie rozwiązania, które spełnia wszystkie ograniczenia

1. Losujemy liczbę  $q$  z zakresu od 1 do  $k \cdot n$ .
2. Obliczamy  $i = \text{floor}((q-1)/k+1)$ .
3. Obliczamy  $j = (q-1) \bmod k + 1$ .
4. Przyjmujemy  $x_{ij} = \min[s(i), d(j)]$
5. Obliczamy  $s(i) = s(i) - x_{ij}$ .
6. Obliczamy  $d(j) = d(j) - x_{ij}$ .
7. Powtarzamy kroki 1 – 7 do momentu wypełnienia macierzy rozwiązania.

## ZADANIE TRANSPORTOWE

Założmy, że mamy 3 miejsca nadania o poziomach zapasów  $s=(15, 25, 5)$  i 4 miejsca odbioru o poziomach odbioru  $d=(5, 15, 15, 10)$ .

1. Losujemy liczbę - 10.

2.  $i = \text{floor}((10-1)/4+1) = 3$ .

3.  $j = (10-1) \bmod 4 + 1 = 2$ .

4.  $x_{32} = \min(s(3), d(2)) = 5$

5.  $s(3) = s(3) - x_{32} = 0$ .

6.  $d(2) = d(2) - x_{32} = 10$ .

|           |    |           |          |    |    |
|-----------|----|-----------|----------|----|----|
|           |    | pozostało |          |    |    |
|           |    | 5         | 10       | 15 | 10 |
| pozostało | 15 |           |          |    |    |
|           | 25 |           |          |    |    |
|           | 0  |           | <b>5</b> |    |    |

W kolejnych krokach wylosowano 8, 5, 3, 1, 11, 4,

12, 7, 6, 9, i 2 co dało macierz końcową:

Sprawdzić!

|          |           |           |           |
|----------|-----------|-----------|-----------|
| <b>0</b> | <b>0</b>  | <b>15</b> | <b>0</b>  |
| <b>5</b> | <b>10</b> | <b>0</b>  | <b>10</b> |
| <b>0</b> | <b>5</b>  | <b>0</b>  | <b>0</b>  |

### Mutacja

1. Losujemy wiersze i kolumny macierzy  $V$  do mutacji, np.  $i = \{2, 4\}$ ,  $j = \{2, 3, 5\}$  (liczba wylosowanych kolumn jest z zakresu od 2 do  $k$ , a wierszy od 2 do  $n$ ).
2. Z elementów leżących na przecięciu tych wierszy i kolumn tworzymy macierz  $W$ .
3. Obliczamy poziomy zapasów i odbiorów z tej macierzy (sumując wiersze i kolumny).
4. Inicjalizujemy macierz  $W$ .
5. Wstawiamy elementy macierzy  $W'$  do macierzy  $V$ .

## Mutacja

$$i = \{2, 4\}, j = \{2, 3, 5\}$$

**V** =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 5 | 0 | 3 |
| 0 | 4 | 0 | 0 | 0 |
| 0 | 0 | 5 | 7 | 0 |
| 3 | 1 | 0 | 0 | 2 |

**W** =

|   |   |   |   |
|---|---|---|---|
|   | 5 | 0 | 2 |
| 4 | 4 | 0 | 0 |
| 3 | 1 | 0 | 2 |

**W'** =

|   |   |   |   |
|---|---|---|---|
|   | 5 | 0 | 2 |
| 4 | 2 | 0 | 2 |
| 3 | 3 | 0 | 0 |

**V** =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 5 | 0 | 3 |
| 0 | 2 | 0 | 0 | 2 |
| 0 | 0 | 5 | 7 | 0 |
| 3 | 3 | 0 | 0 | 0 |

### Krzyżowanie

1. Wybieramy macierze do krzyżowania  $V1$  i  $V2$ .
2. Tworzymy dwie macierze tymczasowe  $DIV = [div_{ij}]$  i  $REM = [rem_{ij}]$ :

$$div_{ij} = \text{floor}((v1_{ij} + v2_{ij})/2)$$

$$rem_{ij} = (v1_{ij} + v2_{ij}) \bmod 2$$

3. Rozkładamy macierz  $REM$  na dwie macierze tak, aby

$$REM = REM1 + REM2$$

$$s_{REM1}(i) = s_{REM2}(i) = s_{REM}(i)/2, \text{ dla } i=1,2,\dots,n$$

$$d_{REM1}(j) = d_{REM2}(j) = d_{REM}(j)/2, \text{ dla } j=1,2,\dots,k$$

4. Tworzymy potomków:

$$V1' = DIV + REM1$$

$$V2' = DIV + REM2$$

### Krzyżowanie

V1 =

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 7 | 0 |
| 0 | 4 | 0 | 0 | 0 |
| 2 | 1 | 4 | 0 | 5 |
| 0 | 0 | 6 | 0 | 0 |

V2 =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 5 | 0 | 3 |
| 0 | 4 | 0 | 0 | 0 |
| 0 | 0 | 5 | 7 | 0 |
| 3 | 1 | 0 | 0 | 2 |

DIV =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 2 | 3 | 1 |
| 0 | 4 | 0 | 0 | 0 |
| 1 | 0 | 4 | 3 | 2 |
| 1 | 0 | 3 | 0 | 1 |

REM =

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

## Krzyżowanie

REM1 =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |

REM2 =

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |

V1' =

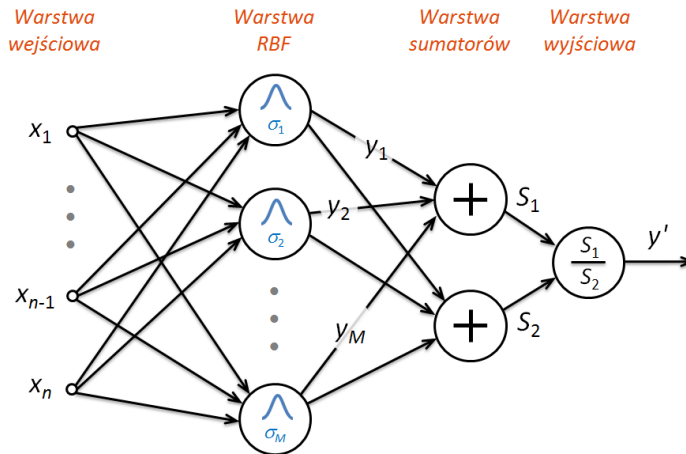
|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 3 | 3 | 2 |
| 0 | 4 | 0 | 0 | 0 |
| 1 | 1 | 4 | 4 | 2 |
| 2 | 0 | 3 | 0 | 1 |

V2' =

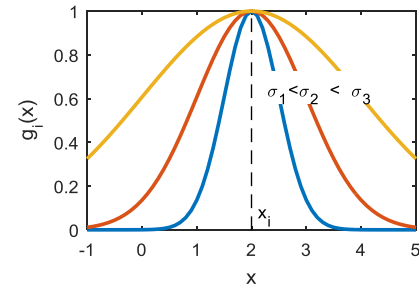
|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 2 | 4 | 1 |
| 0 | 4 | 0 | 0 | 0 |
| 1 | 0 | 5 | 3 | 3 |
| 1 | 1 | 3 | 0 | 1 |

# UCZENIE SIECI NEURONOWEJ GRNN

## Uczenie sieci neuronowej GRNN (sieć regresji uogólnionej)



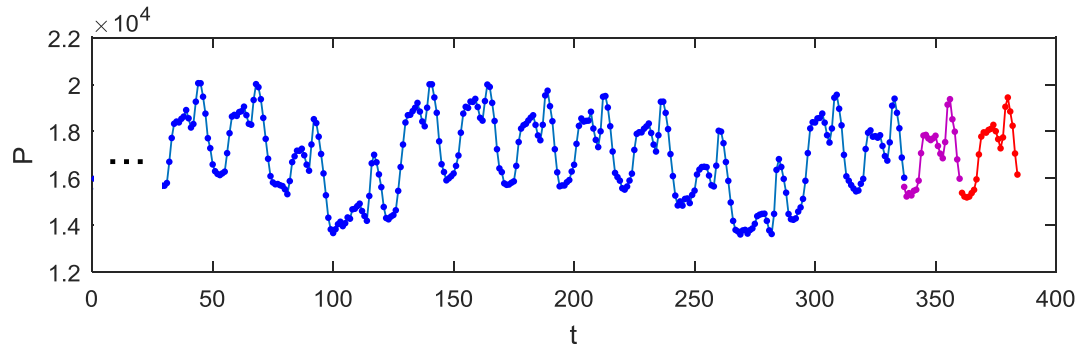
$$g_i(\mathbf{x}) = \exp\left(-\frac{d^2(\mathbf{x}, \mathbf{x}_i)}{2\sigma_i^2}\right)$$



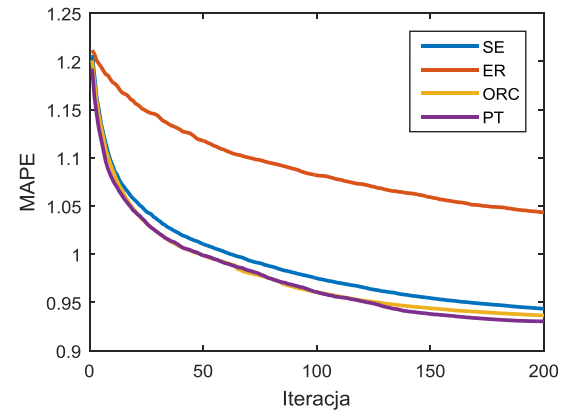
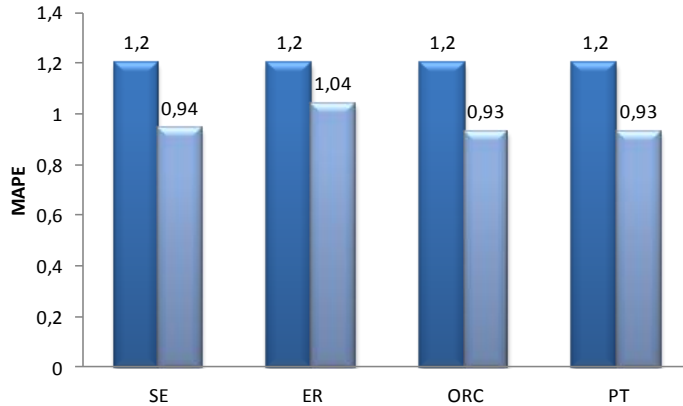
Problem: aproksymacja funkcji  $\mathbf{y} = f(\mathbf{x})$

$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_{24}]^T$  - zakodowane obciążenia systemu elektroenergetycznego w kolejnych godzinach doby poprzedzającej prognozę

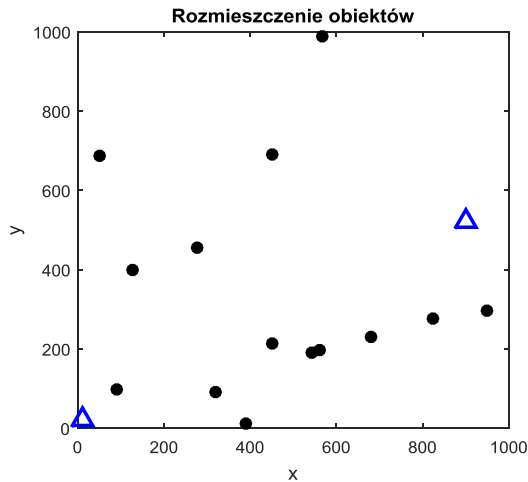
$\mathbf{y} = [y_1 \ y_2 \ \dots \ y_{24}]^T$  - zakodowane obciążenia SE w kolejnych godzinach prognozowanej doby



Funkcja błędu:  $MAPE = \frac{100}{M} \sum_{i=1}^M \left| \frac{P_i - \hat{P}_i}{P_i} \right|$



# LOKALIZACJI STACJI ELEKTROENERGETYCZNEJ NA TERENIE ZAKŁADU PRZEMYSŁOWEGO



Dane:

- położenia obiektów
- moce obiektów
- przekroje i koszty jednostkowe linii kablowych
- dopuszczalne spadki napięć

## LOKALIZACJI STACJI ELEKTROENERGETYCZNEJ

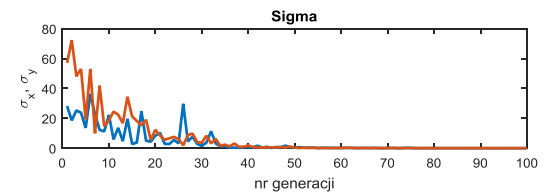
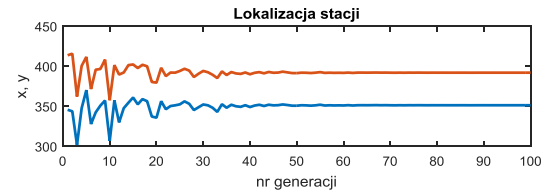
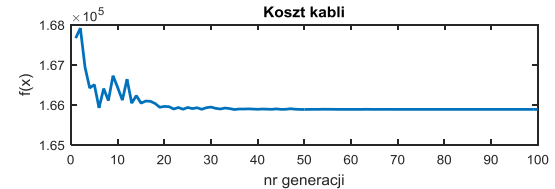
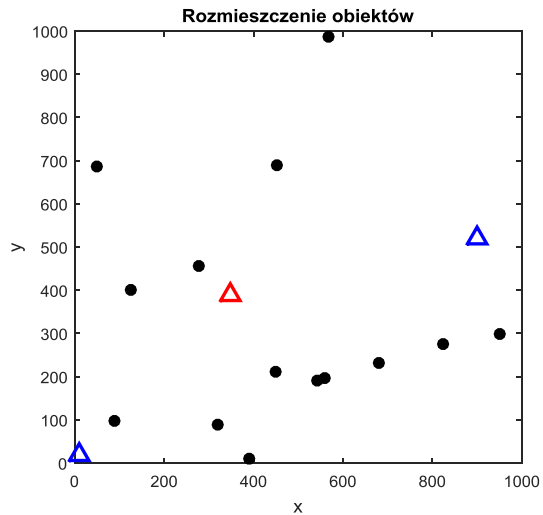
---

Funkcja celu: suma kosztów linii kablowych łączących obiekty  $K = \sum_{i=1}^m d_i k_i \rightarrow \min$

Ograniczenia: spadek napięcia na każdej linii jest mniejszy od dopuszczalnego  
 $\Delta U_{\%i} \leq \Delta U_{\%di}, \quad i = 1, 2, \dots, m$

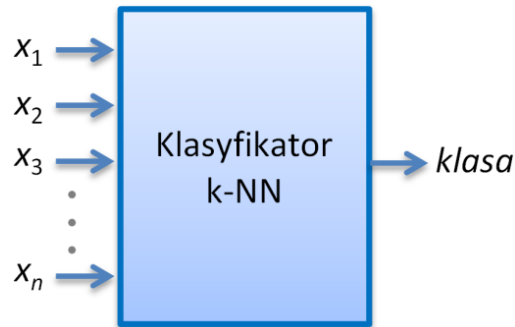
Zastępcza funkcja celu:  $F = K + a \sum_{i=1}^m (\Delta U_{\%di} - \Delta U_{\%i}) \rightarrow \min$

# LOKALIZACJI STACJI ELEKTROENERGETYCZNEJ



# DOBÓR ZMIENNYCH WEJŚCIOWYCH DO MODELU DECYZYJNEGO

---



Problem:

Znaleźć podzbiór zmiennych wejściowych  $x_i$ , dla którego dokładność modelu jest największa

## DOBÓR ZMIENNYCH WEJŚCIOWYCH DO MODELU DECYZYJNEGO

---

Reprezentacja:

$$\mathbf{z} = [1 \ 0 \ 1 \ 0 \ 1 \ \dots \ 1] \in \{0,1\}^n$$

1 na  $i$ -tej pozycji oznacza, że  $x_i$  jest wejściem modelu, 0 - że  $x_i$  jest pomijana

Przystosowanie (ocena modelu przy  $\mathbf{z}$ ):

$$\text{Acc}(\mathbf{z})$$

## DOBÓR ZMIENNYCH WEJŚCIOWYCH DO MODELU DECYZYJNEGO

| Zbiór danych | Bez selekcji | PT           | SW    | AG           | PL    | SDZ          | SEZ          |
|--------------|--------------|--------------|-------|--------------|-------|--------------|--------------|
| Ionosphere   | 84,33        | <b>94,78</b> | 94,00 | 94,40        | 92,17 | 94,02        | 94,30        |
| Cancer       | 96,61        | <b>98,25</b> | 97,97 | 98,11        | 97,57 | 97,72        | 97,54        |
| Heart        | 82,22        | <b>86,25</b> | 85,92 | 86,07        | 85,67 | 85,93        | 85,93        |
| Wine         | 96,07        | <b>98,88</b> | 98,62 | 98,69        | 98,35 | <b>98,88</b> | 97,75        |
| Glass        | 65,89        | <b>74,77</b> | 74,75 | <b>74,77</b> | 74,67 | 73,36        | <b>74,77</b> |
| Diabetes     | 73,96        | <b>77,21</b> | 77,20 | <b>77,21</b> | 77,09 | 76,30        | 76,30        |