

# Užívateľská príručka

## Obsah

Použitie knižnice .....	2
Funkcia selbest – elitaristický výber .....	4
Funkcia selsort – elitaristický výber a usporiadanie .....	5
Funkcia seldiv – výber na základe diverzity .....	6
Funkcia selrand – náhodný výber .....	7
Funkcia selsus – výber váhovaným ruletovým kolesom .....	8
Funkcia seltourn – turnajový výber .....	9
Funkcia mutx – všeobecná mutácia .....	10
Funkcia muta – aditívna mutácia (lokálna) .....	11
Funkcia mutm – multiplikatívna mutácia .....	12
Funkcia intmedx – medziľahlé kríženie .....	13
Funkcia crossgrp – kríženie medzi viacerými rodičmi .....	14
Funkcia crossov - kríženie .....	15
Funkcia swapgen – výmena génov .....	16
Funkcia swappart – výmena častí reťazcov .....	17
Funkcia genrpop – generovanie populácie .....	18

# Použitie knižnice

Knižnica Genetic v jazyku Python sa nachádza v súbore `genetic_all.py`, ktorý obsahuje hlavné funkcie určené na prácu s GA a pomocné funkcie pre hlavné funkcie a testovanie. Knižnicu je možné používať dvomi spôsobmi:

1. Ak ide o prácu v rámci jedného priečinka, stačí knižnicu umiestniť do jedného priečinka spolu s pracovným súborom. Následne stačí knižnicu pripojiť k pracovnému súboru rovnako ako iné moduly cez príkaz

```
1 import genetic_all
```

Pre ľahšiu prácu je možnú využiť jednu z alternatív

```
1 import genetic_all as ga
2 from genetic_all import *
```

Prvá alternatíva skráti názov knižnice pre následný prístup cez bodkovú notáciu objektov (ang. dot notation). Druhý príkaz vloží do pracovného súboru všetky funkcie knižnice, takže nie je potrebný názov knižnice, používajú sa priamo funkcie. Druhá alternatíva však nie je vhodná kvôli neprehľadnosti, možným duplikátnym názvom funkcií a celkovo zvýšenej veľkosti projektu.

2. Ak ide o projekt, ktorý je potrebné štruktúrovať do viacerých priečinkov a je vhodné mať knižnice mimo priečinka s pracovnými súborami, je nutné upraviť predošlé príkazy. Ak sa jedná o nasledujúcu štruktúru projektu

```
▼ test
  ▼ libs
    /* genetic_all.py
    /* main.py
```

stačí pridať názov priečinku ako bodkovú notáciu pred názov knižnice

```
1 import libs.genetic_all
```

3. V prípade, ak sa nachádzajú súbory v rôznych podpriechinkoch, napríklad takto

```
▼ test
  ▼ libs
    /* genetic_all.py
  ▼ main
    /* main.py
```

je možné využiť knižnicu `sys` a pridať absolútnu cestu priečinka s knižnicou do zoznamu existujúcich ciest, z ktorých je možné pridávať moduly do pracovného súboru.

```
1 import sys
2 sys.path.insert(0, 'D:/test/libs')
3 import genetic_all
```

Takto importovaná knižnica je pripravená na používanie. Podľa spôsobu importovania (z predošlého bodu č.1) sa funkcie používajú priamo (napr. `new_pop = selrand(old_pop, N)`), alebo cez bodkovú notáciu kde sa knižnica chápe ako objekt a jej funkcie ako prvky objektu (napr. `new_pop = ga.selrand(old_pop, N)`).

Funkcie sú prispôsobené na prácu s knižnicou NumPy. Očakávaný vstup a výstup funkcií teda nie je klasický list vytvorený príkazom

```
1 import genetic_all as ga
2
3 space = [[0, 0], [5, 5]]
4 pop = ga.genrpop(5, space)
```

ale NumPy pole

```
1 import numpy as np
2 import genetic_all as ga
3
4 space = np.array([[0, 0], [5, 5]])
5 pop = ga.genrpop(5, space)
```

Použitie listov namiesto polí môže spôsobiť chyby alebo nesprávne a neočakávané výsledky.

## Funkcia selbest – elitaristický výber

**Syntax:**  $new\_pop, new\_fit = selbest(old\_pop, old\_pop\_fit, n\_list, reverse=False)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  zo starej populácie  $old\_pop$  na základe hodnôt účelovej funkcie, ktoré sú vo vektore  $old\_pop\_fit$ . Vektor  $n\_list$  určuje počet reťazcov ktoré budú skopírované do novej populácie, pričom prvá hodnota určuje koľkokrát sa skopíruje najúspešnejší jedinec, druhá určuje druhý najúspešnejší jedinec atď. Najúspešnejší jedinec je jedinec s najmenšou hodnotou účelovej funkcie. Voliteľný argument  $reverse$ , ktorý je predvolene nastavený na  $False$  je možné použiť na výber najhorších jedincov (alebo ak je najvyššia hodnota účelovej funkcie najlepšia). Funkcia taktiež vráti vektor účelových funkcií novej populácie  $new\_fit$ .

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = selbest(Pop, Fit, [1, 2, 0, 1])$$

$$new\_pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 4 \end{bmatrix}$$

## Funkcia selsort – elitaristický výber a usporiadanie

**Syntax:**  $new\_pop, new\_fit = selsort(old\_pop, old\_pop\_fit, n, reverse=False)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  zo starej populácie  $old\_pop$  na základe hodnôt účelovej funkcie, ktoré sú vo vektore  $old\_pop\_fit$ . Stará populácia bude zoradená podľa hodnôt vektora  $old\_pop\_fit$ , a do novej populácie sa skopíruje prvých  $n$  najúspešnejších jedincov. Najúspešnejší jedinec je jedinec s najmenšou hodnotou účelovej funkcie. Voliteľný argument  $reverse$ , ktorý je predvolene nastavený na  $False$  je možné použiť na výber najhorších jedincov (alebo ak je najvyššia hodnota účelovej funkcie najlepšia). Funkcia taktiež vráti vektor účelových funkcií novej populácie  $new\_fit$ .

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = selsort(Pop, Fit, 3)$$

$$new\_pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

## Funkcia seldiv – výber na základe diverzity

**Syntax:**  $new\_pop, new\_fit = seldiv(old\_pop, old\_pop\_fit, n\_list, mode, reverse=False)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  zo starej populácie  $old\_pop$  na základe hodnôt účelovej funkcie, ktoré sú vo vektore  $old\_pop\_fit$ . Stará populácia bude zoradená podľa hodnôt vektora  $old\_pop\_fit$ , na základe rozdielu od referenčného reťazca. Referenčný reťazec môže byť najlepší reťazec populácie ( $mode\ 1$ ) alebo je to reťazec priemerov génov všetkých jedincov ( $mode\ 0$ ). Vektor  $n\_list$  určuje počet reťazcov ktoré budú skopírované do novej populácie, pričom prvá hodnota určuje koľkokrát sa skopíruje najodlišnejší jedinec, druhá určuje druhý najodlišnejší jedinec atď. Voliteľný argument  $reverse$ , ktorý je predvolene nastavený na  $False$  je možné použiť na výber najmenej odlišných jedincov. Funkcia taktiež vráti vektor účelových funkcií novej populácie  $new\_fit$ .

### Príklad č.1:

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = seldiv(Pop, Fit, [2, 2], 0)$$

$$new\_pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 1 \\ 1 \\ 5 \\ 5 \end{bmatrix}$$

### Príklad č.2:

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = seldiv(Pop, Fit, [1, 0, 1], 1)$$

$$new\_pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

## Funkcia selrand – náhodný výber

**Syntax:**  $new\_pop, new\_fit = selrand(old\_pop, old\_pop\_fit, n)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  zo starej populácie  $old\_pop$ . Reťazce budú vybrané z celej populácie náhodne. Parameter  $n$  určuje počet reťazcov ktoré budú skopírované do novej populácie. Funkcia taktiež vráti vektor účelových funkcií novej populácie  $new\_fit$ .

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = selrand(Pop, Fit, 5)$$

$$new\_pop = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 & 5 \\ 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 2 \\ 3 \\ 5 \\ 2 \\ 4 \end{bmatrix}$$

## Funkcia selsus – výber váhovaným ruletovým kolesom

**Syntax:**  $new\_pop, new\_fit = selsus(old\_pop, old\_pop\_fit, n, reverse=False)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  zo starej populácie  $old\_pop$  na základe hodnôt účelovej funkcie, ktoré sú vo vektore  $old\_pop\_fit$ . Funkcia prideliť reťazcom výseky ruletového kolesa podľa ich hodnôt úspešnosti, pričom najúspešnejší reťazec dostane najväčší výsek. Funkcia následne podľa parametra  $n$  vyberie jedince do novej populácie. Funkcia taktiež vráti vektor účelových funkcií novej populácie  $new\_fit$ .

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = selsus(Pop, Fit, 5)$$

$$new\_pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 5 \\ 3 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$



## Funkcia seltourn – turnajový výber

**Syntax:**  $new\_pop, new\_fit = seltourn(old\_pop, old\_pop\_fit, n, reverse=False)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  zo starej populácie  $old\_pop$  na základe hodnôt účelovej funkcie, ktoré sú vo vektore  $old\_pop\_fit$ . Funkcia náhodne vyberie  $n$  párov reťazcov, ktorých úspešnosti budú porovnané, a úspešnejší reťazec bude vybraný do novej populácie. Voliteľný argument  $reverse$ , ktorý je predvolene nastavený na  $False$  je možné použiť na výber horších jedincov (alebo ak je vyššia hodnota účelovej funkcie lepšia). Funkcia taktiež vráti vektor účelových funkcií novej populácie  $new\_fit$ .

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Fit = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$new\_pop, new\_fit = seltourn(Pop, Fit, 6)$$

$$new\_pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \quad new\_fit = \begin{bmatrix} 1 \\ 4 \\ 1 \\ 1 \\ 3 \\ 2 \end{bmatrix}$$

## Funkcia *mutx* – všeobecná mutácia

**Syntax:** *mutx(old\_pop, rate, space)*

**Popis:** Funkcia zmutuje vstupnú populáciu *old\_pop*. Počet zmutovaných prvkov sa určuje podľa veľkosti populácie a parametra *rate*, ktorý má hodnotu v rozsahu  $<0; 1>$ . Funkcia môže zmutovať jeden gén viackrát. Nové hodnoty génov budú náhodné čísla z rozsahu populácie, ktorý sa nachádza v matrice *space*, kde prvý riadok určuje minimálne hodnoty a druhý riadok maximálne. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia *old\_pop*.

**Príklad:**

$$Pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad Space = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

*mutx(Pop, 0.2, Space)*

$$Pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 8.99 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5.39 & 6.15 \\ 5 & 1.45 & 5 & 5 & 5 \\ 5 & 5 & 5 & 2.34 & 5 \end{bmatrix}$$

## Funkcia muta – aditívna mutácia (lokálna)

**Syntax:** *muta(old\_pop, rate, amp, space)*

**Popis:** Funkcia zmutuje vstupnú populáciu *old\_pop*. Počet zmutovaných prvkov sa určuje podľa veľkosti populácie a parametra *rate*, ktorý má hodnotu v rozsahu  $<0; 1>$ . Funkcia môže zmutovať jeden gén viackrát. Ku náhodne vybraným génom bude pripočítaná náhodne vybraná hodnota, ktorej rozsah určuje vektor absolútnych hodnôt *amp*. Funkcia nakoniec skontroluje, či nové hodnoty patria do rozsahu populácie, ktorý sa nachádza v matice *space*, kde prvý riadok určuje minimálne hodnoty a druhý riadok maximálne. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia *old\_pop*.

**Príklad:**

$$Pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

$$Space = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

$$Amp = [3 \quad 3 \quad 3 \quad 3 \quad 3]$$

*muta(Pop, 0.2, Amp, Space)*

$$Pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 6.76 \\ 5 & 7.28 & 4.35 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 3.30 & 5 & 5 \\ 4.99 & 5 & 5 & 5 & 5 \end{bmatrix}$$

## Funkcia mutm – multiplikatívna mutácia

**Syntax:** *mutm(old\_pop, rate, amp, space)*

**Popis:** Funkcia zmutuje vstupnú populáciu *old\_pop*. Počet zmutovaných prvkov sa určuje podľa veľkosti populácie a parametra *rate*, ktorý má hodnotu v rozsahu  $<0; 1>$ . Funkcia môže zmutovať jeden gén viackrát. Náhodne vybrané gény budú vynásobené náhodnou hodnotou, ktorej rozsah určuje dvojriadková matica *amp*. Funkcia nakoniec skontroluje, či nové hodnoty patria do rozsahu populácie, ktorý sa nachádza v matici *space*, kde prvý riadok určuje minimálne hodnoty a druhý riadok maximálne. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia *old\_pop*.

**Príklad:**

$$Pop = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

$$Space = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

$$Amp = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \end{bmatrix}$$

*mutm(Pop, 0.2, Amp, Space)*

$$Pop = \begin{bmatrix} 5 & 5 & 5 & 5.64 & 5 \\ 5 & 6.17 & 5 & 5 & 6.96 \\ 5 & 5 & 5 & 5 & 6.69 \\ 5 & 5 & 5 & 7.05 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

## Funkcia `intmedx` – medzil'ahlé kríženie

**Syntax:** `intmedx(old_pop, alpha, mode)`

**Popis:** Funkcia skombinuje páry vstupnej populácie `old_pop` a podľa vzorca vytvorí nové reťazce. Podľa parametra `mode` budú vybrané ako dvojice buď susedné (`mode 0`) alebo náhodné (`mode 1`) reťazce. Každý reťazec bude vybraný najviac jedenkrát. Z každej dvojice vzniknú 2 reťazce. Parameter `alpha` označuje faktor intenzity kríženia génov. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia `old_pop`.

**Príklad č.1:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

`intmedx(Pop, 0.25, 0)`

$$Pop = \begin{bmatrix} 1.13 & 1.19 & 1.06 & 1.03 & 1.02 \\ 1.89 & 1.93 & 1.96 & 1.85 & 1.83 \\ 3.01 & 3.13 & 3.17 & 3.03 & 3.12 \\ 3.91 & 3.76 & 3.90 & 3.84 & 3.93 \\ 5.19 & 5.02 & 5.10 & 5.08 & 5.12 \\ 5.92 & 5.76 & 5.82 & 5.99 & 5.87 \end{bmatrix}$$

**Príklad č.2:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 \end{bmatrix}$$

`intmedx(Pop, 0.5, 1)`

$$Pop = \begin{bmatrix} 1.00 & 1.86 & 1.81 & 1.58 & 1.45 \\ 2.00 & 2.00 & 2.00 & 2.00 & 2.00 \\ 3.53 & 3.47 & 3.71 & 3.76 & 3.80 \\ 4.16 & 4.16 & 4.29 & 4.41 & 4.28 \\ 4.18 & 4.85 & 4.78 & 4.57 & 4.28 \\ 4.95 & 4.54 & 5.55 & 5.55 & 5.54 \\ 6.07 & 6.50 & 6.28 & 6.30 & 6.54 \end{bmatrix}$$

## Funkcia `crossgrp` – kríženie medzi viacerými rodičmi

**Syntax:**  $new\_pop = crossgrp(old\_pop, n)$

**Popis:** Funkcia vytvorí novú populáciu  $new\_pop$  s  $n$  reťazcami. Gény nových reťazcov budú náhodne vybrané zo všetkých reťazcov vstupnej populácie  $old\_pop$ .

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

$$new\_pop = crossgrp(Pop, 5)$$

$$new\_pop = \begin{bmatrix} 7 & 7 & 4 & 2 & 2 \\ 5 & 7 & 2 & 1 & 6 \\ 1 & 3 & 5 & 3 & 4 \\ 4 & 2 & 5 & 7 & 1 \\ 5 & 7 & 3 & 2 & 7 \end{bmatrix}$$

## Funkcia crossov - kríženie

**Syntax:** *crossov(old\_pop, pts, mode)*

**Popis:** Funkcia skombinuje páry vstupnej populácie *old\_pop* a vytvorí z nich nové reťazce prehodením častí reťazca v náhodne vybraných zlomových bodoch. Počet zlomových bodov určuje parameter *pts*. Nie je ošetrovaný stav ak užívateľ použije väčšie množstvo bodov ako je veľkosť reťazcov. Podľa parametra *mode* budú vybrané ako dvojice buď susedné (*mode 0*) alebo náhodné (*mode 1*) reťazce. Každý reťazec bude vybraný najviac jedenkrát. Z každej dvojice vzniknú 2 reťazce. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia *old\_pop*.

**Príklad č.1:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

*crossov(Pop, 1, 1)*

$$Pop = \begin{bmatrix} 1 & 1 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 2 & 2 & 6 & 6 \\ 3 & 3 & 3 & 5 & 5 & 5 & 5 & 5 \\ 4 & 4 & 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 5 & 5 & 3 & 3 & 3 & 3 & 3 \\ 6 & 6 & 6 & 6 & 6 & 6 & 2 & 2 \end{bmatrix}$$

**Príklad č.2:**

$$Pop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{bmatrix}$$

*crossov(Pop, 3, 0)*

$$Pop = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 2 & 2 & 2 \\ 2 & 1 & 2 & 2 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 4 & 3 & 4 & 4 \\ 4 & 4 & 4 & 4 & 3 & 4 & 3 & 3 \\ 5 & 6 & 5 & 6 & 6 & 6 & 6 & 6 \\ 6 & 5 & 6 & 5 & 5 & 5 & 5 & 5 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{bmatrix}$$

## Funkcia swapgen – výmena génov

**Syntax:** *swapgen(old\_pop, rate)*

**Popis:** Funkcia zmutuje vstupnú populáciu *old\_pop*. Počet zmutovaných prvkov sa určuje podľa veľkosti populácie a parametra *rate*, ktorý má hodnotu v rozsahu  $<0; 1>$ . Funkcia môže zmutovať jeden gén viackrát. Mutácia sa vykoná výmenou pozície dvoch génov jedného reťazca. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia *old\_pop*.

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

*swapgen(Pop, 0.2)*

$$Pop = \begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 \\ 1 & 2 & 4 & 3 & 4 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 2 & 3 & 1 & 5 & 6 \\ 1 & 3 & 2 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$



## Funkcia swappart – výmena častí reťazcov

**Syntax:** *swappart(old\_pop, rate)*

**Popis:** Funkcia zmutuje vstupnú populáciu *old\_pop*. Počet zmutovaných prvkov sa určuje podľa veľkosti populácie a parametra *rate*, ktorý má hodnotu v rozsahu  $<0; 1>$ . Funkcia môže zmutovať jeden gén viackrát. Mutácia sa vykoná rozdelením reťazca v náhodnej pozícii a následnou výmenou poradia vzniknutých dvoch častí. Funkcia nemá návratovú hodnotu, mení sa priamo vstupná populácia *old\_pop*.

**Príklad:**

$$Pop = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

*swappart(Pop, 0.2)*

$$Pop = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 1 \\ 4 & 5 & 6 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

## Funkcia genrpop – generovanie populácie

**Syntax:**  $pop = \text{genrpop}(pop\_size, space)$

**Popis:** Funkcia vygeneruje populáciu  $pop$  podľa parametru  $pop\_size$ , ktorý určuje počet reťazcov a dvojriadkovej matice  $space$  ktorá určuje počet génov a ich ohraničenie. Prvý riadok matice určuje minimálne hodnoty a druhý riadok maximálne.

**Príklad:**

$$Space = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

$$Pop = \text{genrpop}(6, Space)$$

$$Pop = \begin{bmatrix} 9.02 & 2.36 & 9.54 & 6.09 & 8.19 \\ 7.65 & 7.51 & 4.55 & 6.62 & 7.46 \\ 9.79 & 4.36 & 4.77 & 5.74 & 9.98 \\ 6.54 & 1.40 & 0.63 & 3.49 & 6.44 \\ 1.55 & 0.73 & 7.40 & 7.44 & 4.16 \\ 0.26 & 7.04 & 7.65 & 2.70 & 6.75 \end{bmatrix}$$