

TOOLBOX - GENETICKÉ ALGORITMY

pre riešenie optimalizačných problémov v prostredí Matlab®

POUŽÍVATEĽSKÁ PRÍRUČKA

Ivan Sekaj

Úvod

Toolbox - Genetické algoritmy je určený na použitie v programovom prostredí Matlab pre riešenie optimalizačných problémov, ktoré používajú predovšetkým reálnečíselné kódovanie reťazcov. Jeho ďalším špecifikom je, že je minimalizujúci, čiže za lepšie jedince, ktoré sú preferované, považuje tie, ktoré dosahujú menších hodnôt účelovej funkcie resp. fitness funkcie. V prípade maximalizácie sa použije doplnková účelová funkcia. Hodnoty účelovej funkcie alebo fitness môžu nadobúdať kladné aj záporné reálnečíselné hodnoty.

Prehľad funkcií

among - medziľahlé kríženie.....	37
change - mutácia duplicitných reťazcov.....	38
crosgp - kríženie medzi viacerými rodičmi	39
crossov - viacbodové kríženie dvoch rodičov.....	40
genrpop - vygenerovanie náhodnej reálnečíselnej populácie.....	41
invfit - doplnok účelovej funkcie	42
muta - aditívna mutácia	43
mutm - multiplikatívna mutácia.....	44
mutx - obyčajná mutácia.....	45
selbest - výber najlepších reťazcov.....	46
seldiv - výber podľa maximálnej diverzity.....	47
selrand - náhodný výber reťazcov.....	48
selsort - výber a zotriedenie reťazcov podľa úspešnosti.....	49
selsus - výber pomocou váhovaného ruletového kola	50
shake - náhodné premiešanie poradia reťazcov v populácii.....	51
swapgen - mutácia poradia génov v reťazci (permutačné).....	52
swappart - vzájomná výmena poradia dvoch častí reťazcov (permutačné)	53
intmedx - medziľahlé kríženie (nové)	
crosgord - kríženie permutačného typu	
invord - inverzia subreťazcov (permutačné)	

among - medziľahlé kríženie

Charakteristika:

Funkcia vytvorí novú populáciu reťazcov, ktorá vznikne medziľahlým skrížením susedných alebo náhodne vytvorených dvojíc reťazcov pôvodnej populácie. Z každej dvojice rodičov vzniknú dvaja noví potomkovia využitím vzťahu

$$P = R_1 + \alpha(R_2 - R_1)$$

kde P je reťazec potomka, R_1 a R_2 sú rodičovské reťazce, α je matica obsahujúca v diagonále náhodné čísla $0 \leq \alpha_i \leq 1$ a mimo diagonály 0. Vzniknutí potomkovia budú lokalizovaní v priestore "medzi" obomi rodičmi. Krížené sú všetky reťazce z populácie Oldpop (ak ich je párny počet).

Syntax: Newpop=among(Oldpop,sel)

Newpop - novovytvorená populácia

Oldpop - pôvodná populácia

sel - spôsob výberu krížených dvojíc:

0-náhodné dvojice

1-susedné dvojice v populácii

Príklad:

```
A = [ 1   1   1   1
      2   2   2   2
      3   3   3   3
      4   4   4   4
      5   5   5   5
      6   6   6   6 ]
```

B=among(A,1)

```
B = [ 1.8460  1.8415  1.4154  1.4679
      1.4121  1.2693  1.5373  1.2872
      3.1783  3.5717  3.0331  3.4985
      3.1537  3.8024  3.5344  3.9554
      5.7483  5.8907  5.8420  5.2128
      5.5546  5.6248  5.1598  5.7147 ]
```

change - eliminácia duplicitných reťazcov

Charakteristika:

Funkcia vyhladá a zmení všetky duplicitné reťazce v populácii. V závislosti od premennej *option* sa duplicitné reťazce buď zmutujú v jednom géne (aditívne alebo multiplikatívne) alebo sa nahradia reťazcom náhodných čísel.

Syntax:

Newpop=change(Oldpop,option,Space)

Newpop - nová zmenená (výstupná) populácia

Oldpop - stará (vstupná) populácia

option - 0 - duplicitné reťazce sa aditívne zmutujú v jednom géne
hodnotami z rozsahu Space

1 - duplicitné reťazce sa multiplikatívne zmutujú v jednom géne
hodnotami z rozsahu Space

2 - duplicitné reťazce sa nahradia reťazcami náhodných čísel
z rozsahu Space

Space - matica o dvoch riadkoch a toľkých stĺpcoch ako je dĺžka
reťazca, prvý riadok matice určuje dolné ohraničenia
jednotlivých génov reťazcov a druhý riadok horné ohraničenia.

Príklad:

```
A = [ 1  2  3  4  5
      4  8  9  3  7
      1  2  3  4  5
      1  1  1  1  1
      1  2  3  4  5
      1  2  3  4  5
      7  8  6  3  0
      1  1  1  1  1 ]
```

```
space=[ 0  0  0  0  0
        10 10 10 10 10 ]
```

B=change(A,0,space)

```
B = [ 1.0000  2.0000  3.0000  4.0000  5.0000
      4.0000  8.0000  9.0000  3.0000  7.0000
      1.0000  2.0000  3.0000  4.1600  5.0000
      1.0000  1.0000  1.0000  1.0000  1.0000
      1.0000  2.0000  3.0000  9.1032  5.0000
      1.0000  2.0000  3.0000  2.6245  5.0000
      7.0000  8.0000  6.0000  3.0000  0
      7.3608  1.0000  1.0000  1.0000  1.0000 ]
```

crosgrp - kríženie medzi viacerými rodičmi

Charakteristika:

Funkcia skríži viac rodičov a vytvorí určený počet potomkov, ktorých gény sú náhodne zdedené od jednotlivých rodičov. Počet rodičov môže byť väčší ako 2. Počet potomkov a rodičov nemusí byť rovnaký.

Syntax:

`Newgroup=crosgrp(Oldgroup,num)`

Newgroup - novovytvorená skupina skrížených potomkov

Oldgroup - skupina rodičov

num - počet novovytvorených potomkov

Príklad:

```
A = [ 0  0  0  0
      2  2  2  2
      3  3  3  3
      4  4  4  4
      5  5  5  5
      9  9  9  9 ]
```

`B=crosgrp(A, 10)`

```
B=[ 3  2  4  3
    9  5  3  0
    5  3  4  5
    9  5  2  3
    5  9  3  9
    0  3  5  0
    0  2  2  4
    2  2  0  5
    3  9  3  3
    4  4  2  5 ]
```

CROSSOV - kríženie

Charakteristika:

Funkcia vytvorí novú populáciu reťazcov, ktorá vznikne skrížením všetkých reťazcov starej populácie 1- až 4-bodovým krížením. Krížené sú všetky reťazce (ak je ich párny počet). Výber párov je buď náhodný alebo sú vybrané susedné reťazce v populácii podľa voľby parametra sel.

Syntax:

Newpop=crossov(Oldpop,num,sel)

Newpop - matica skríženej (výstupnej) populácie

Oldpop - pôvodná (vstupná) populácia

num - počet bodov kríženia (miest rozdelenia) od 1 do 4

sel - spôsob výberu dvojíc:

0 - náhodný

1 - susedné dvojice v populácii

Príklad:

A = [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]

B=crossov(A,1,0)

B = [1 1 5 5 5
 2 2 3 3 3
 3 3 2 2 2
 4 4 4 4 6
 5 5 1 1 1
 6 6 6 6 4]

genrpop - vygenerovanie nahodnej reálnečíselnej populácie

Charakteristika:

Funkcia vygeneruje populáciu zvoleného počtu reťazcov. Jednotlivé gény reťazcov obsahujú náhodné reálne čísla z definovaného rozsahu. Rozsah sa definuje pre každý gén samostatne pomocou matice *Space*. Prvý riadok tejto matice určuje dolné ohraničenia a druhý riadok horné ohraničenia jednotlivých génov.

Syntax:

Newpop=genrpop(popsize ,Space)

Newpop - nová, zmutovaná populácia

popsize - počet reťazcov generovanej populácie

Space - matica obmedzení, ktorej 1. riadok je vektor minimálnych a 2. riadok je vektor maximálnych prípustných hodnôt génov

Príklad:

$$S = \begin{bmatrix} -10 & 0 & 0 & 0 & -1 \\ 10 & 1 & 10 & 10 & 0 \end{bmatrix}$$

A=genrpop(6 ,S)

$$A = \begin{bmatrix} -7.502 & 0.521 & 1.778 & 0.832 & -0.336 \\ 2.771 & 0.199 & 5.080 & 2.518 & -0.248 \\ 3.960 & 0.784 & 3.961 & 3.087 & -0.589 \\ -5.048 & 0.419 & 8.332 & 3.669 & -0.400 \\ 5.129 & 0.802 & 0.710 & 7.887 & -0.054 \\ 9.362 & 0.542 & 6.120 & 1.722 & -0.963 \end{bmatrix}$$

genrpop - vygenerovanie náhodnej reálnečíselnej populácie

Charakteristika:

Funkcia vygeneruje populáciu reťazcov náhodných reálnych čísel z priestoru ohraničeného prvkami dvojriadkovej matice *Space*. Prvý riadok tejto matice určuje dolné ohraničenia jednotlivých génov reťazcov a druhý riadok horné ohraničenia. Dĺžka reťazcov je určená počtom prvkov v riadku v matici obmedzení.

Syntax:

Newpop=genrpop(popsize, Space)

Newpop - matica vytvorenej populácie

popsize - požadovaný počet reťazcov (chromozómov) v populácii

Space - matica obmedzení, ktorej 1.riadok je vektor minimálnych
a 2. riadok je vektor maximálnych prípustných hodnôt génov
reťazcov

Príklad:

$$S = \begin{bmatrix} 0 & 0 & 0 & -10 \\ 1 & 10 & 100 & 0 \end{bmatrix}$$

A=genrpop(10,S)

$$A = \begin{bmatrix} 0.5192 & 7.7194 & 6.5356 & -5.5721 \\ 0.9772 & 4.6774 & 32.9056 & -5.5407 \\ 0.7433 & 2.0534 & 17.1379 & -6.2748 \\ 0.9397 & 9.6485 & 25.4957 & -9.2966 \\ 0.1238 & 5.2634 & 16.0083 & -4.8229 \\ 0.1052 & 8.4158 & 86.8562 & -5.7611 \\ 0.3272 & 7.1355 & 57.6776 & -1.2812 \\ 0.7582 & 8.3660 & 69.9791 & -6.1094 \\ 0.9352 & 6.0560 & 36.2084 & -4.6050 \\ 0.4551 & 4.0169 & 24.0440 & -9.1691 \end{bmatrix}$$

invfit - doplnok k účelovej funkcii

Charakteristika:

Funkcia vypočíta doplnok k vektoru hodnôt účelovej funkcie podľa vzťahu

$$\text{Newobj} = (\max(\text{Oldobj}) - \text{Oldobj}) + \min(\text{Oldobj})$$

Takto je možné z maximalizačnej úlohy vytvoriť minimalizačnú úlohu pre potreby tohoto toolboxu.

Syntax:

$$\text{Newobj} = \text{invfit}(\text{Oldobj})$$

Newobj - (výstupný) vektor hodnôt doplnkovej účelovej funkcie

Oldobj - (vstupný) vektor hodnôt pôvodnej účelovej funkcie

Príklad:

```
old=[1 2 3 10 0.1 100]
```

```
new=invfit(old)
```

```
new=[ 99.1000  98.1000  97.1000  90.1000 100.0000  0.1000 ]
```

muta - aditívna mutácia

Charakteristika:

Funkcia zmutuje populáciu reťazcov s intenzitou úmernou parametru *rate* (z rozsahu od 0 do 1). Mutovaných je len niekoľko génov v rámci celej populácie. Mutácie vzniknú pripočítaním alebo odpočítaním náhodných čísel ohraničených veľkostí k pôvodným hodnotám náhodne vybraných génov celej populácie. Absolútne hodnoty prípustných veľkostí aditívnych mutácií sú ohraničené hodnotami vektora *Amp*. Po tejto operácii sú ešte výsledné hodnoty génov ohraničené (saturované) na hodnoty prvkov matice *Space*. Prvý riadok matice určuje dolné ohraničenia a druhý riadok horné ohraničenia jednotlivých génov.

Syntax:

Newpop=muta(Oldpop,rate,Amp,Space)

Newpop - nová, zmutovaná populácia

Oldpop - stará populácia

Amp - vektor ohraničení prípustných aditívnych hodnôt mutácií

Space - matica obmedzení, ktorej 1. riadok je vektor minimálnych a 2. riadok je vektor maximálnych prípustných mutovaných hodnôt

rate - miera početnosti mutovania génov v populácii (od 0 do 1)

Príklad:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$
$$Amp = [1 \quad 1 \quad 1 \quad 1 \quad 1]$$
$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

B=muta(A, 0.1 , Amp , S)

$$B = \begin{bmatrix} 1.000 & 1.000 & 1.000 & 0.832 & 1.000 \\ 2.000 & 2.000 & 2.000 & 2.518 & 2.000 \\ 3.000 & 3.000 & 3.000 & 3.000 & 3.000 \\ 4.000 & 4.000 & 4.000 & 4.000 & 4.000 \\ 5.000 & 5.000 & 5.000 & 5.000 & 5.000 \\ 6.000 & 5.494 & 6.000 & 6.000 & 6.000 \end{bmatrix}$$

mutm - multiplikatívna mutácia

Charakteristika:

Funkcia zmutuje populáciu reťazcov s intenzitou úmernou parametru *rate* (z rozsahu od 0 do 1). Mutovaných je len niekoľko génov v rámci celej populácie. Mutácie vzniknú vynásobením pôvodných hodnôt náhodne vybraných génov náhodnými číslami ohraňených veľkostí. Prípustné veľkosti multiplikatívnych hodnôt sú ohraňené hodnotami dvojriadkovej matice *Amps*. Po tejto operácii sú ešte výsledné hodnoty génov ohraňené prvkami matice *Space*. Prvý riadok oboch matíc určuje dolné ohraňenia a druhý riadok horné ohraňenia jednotlivých génov.

Syntax:

Newpop=mutm(Oldpop,rate,Amps,Space)

Newpop - nová, zmutovaná populácia

Oldpop - stará populácia

Amps - matica obmedzení multiplikatívnych hodnôt, ktorej 1.riadok je vektor maximálnych a 2. riadok je vektor minimálnych prípustných multiplikatívnych hodnôt

Space - matica obmedzení, ktorej 1.riadok je vektor minimálnych a 2. riadok je vektor maximálnych prípustných mutovaných hodnôt

rate - miera početnosti mutovania génov v populácii (od 0 do 1)

Príklad:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$
$$Amp = \begin{bmatrix} 0.5 & 0.5 & 0.9 & 0.9 & 0.9 \\ 1.0 & 1.0 & 1.5 & 2.0 & 2.0 \end{bmatrix}$$
$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

B=mutm(A,0.1,Amp,S)

$$B = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 2.0000 & 1.1665 & 2.0000 & 2.0000 & 2.0000 \\ 3.0000 & 3.0000 & 3.0000 & 3.0000 & 5.7004 \\ 3.2633 & 4.0000 & 4.0000 & 4.0000 & 4.0000 \\ 5.0000 & 5.0000 & 5.2133 & 5.0000 & 5.0000 \\ 6.0000 & 4.3092 & 6.0000 & 10.0000 & 6.0000 \end{bmatrix}$$

mutx - obyčajná mutácia

Charakteristika:

Funkcia zmutuje populáciu reťazcov s intenzitou úmernou parametru rate (z rozsahu od 0 do 1). Mutovaných je len niekoľko génov v rámci celej populácie. Mutované hodnoty sú zmenené na náhodné hodnoty z priestoru definovaného ohraničeniami pomocou dvojriadkovej matice. Prvý riadok matice určuje dolné ohraničenia jednotlivých génov reťazcov a druhý riadok ich horné ohraničenia.

Syntax:

Newpop=mutx(Oldpop,rate,Space)

Newpop - nová, zmutovaná populácia

Oldpop - stará populácia

Space - matica obmedzení, ktorej 1.riadok je vektor minimálnych a 2. riadok je vektor maximálnych prípustných mutovaných hodnôt

rate - miera početnosti mutovania génov v populácii (od 0 do 1)

Príklad:

```
A = [ 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 ]
```

```
S = [ 0  0  0  0  0
      10 10 10 10 10 ]
```

B=mutx(A,0.1,S)

```
B = [1.0000  1.0000  1.0000  1.0000  1.0000
      2.0000  6.7135  2.0000  2.0000  2.0000
      2.9209  3.0000  3.0000  3.0000  3.0000
      4.0000  4.0000  4.0000  4.0000  4.0000
      5.0000  5.0000  5.0000  5.8097  5.0000
      6.0000  6.0000  6.0000  6.0000  6.0000 ]
```

selbest - výber najlepších jedincov

Charakteristika:

Funkcia skopíruje zo vstupnej populácie do výstupnej populácie určené počty reťazcov v závislosti od ich hodnôt účelovej funkcie. O tom, ktoré reťazce budú kopírované, rozhoduje vektor *Nums*, ktorého prvky určujú počty vybraných reťazcov nasledovne: prvá hodnota určuje koľkokrát sa skopíruje najúspešnejší reťazec do výstupnej populácie, druhá hodnota určuje koľkokrát sa skopíruje 2. najúspešnejší reťazec do výstupnej populácie atď. Najúspešnejšími reťazcami sú chápané jedince s najmenšími hodnotami účelovej funkcie.

Syntax:

Newpop=selbest(Oldpop,Objpop,Nums);

Newpop - nová (výstupná) populácia

Oldpop - stará (vstupná) populácia

Objpop - vektor hodnôt účelovej funkcie starej populácie

Nums - vektor, ktorého prvky určujú, koľkokrát sa reťazec na príslušnom poradí úspešnosti skopíruje do cieľovej populácie

Príklad:

A = [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]

uf=[1 2 3 4 5 6]

vyber=[2 0 1 1]

B=selbest(A,uf,vyber)

B = [1 1 1 1 1
 1 1 1 1 1
 3 3 3 3 3
 4 4 4 4 4]

seldiv - výber na základe maximálnej miery diverzity

Charakteristika:

Funkcia vyberie zo vstupnej populácie do výstupnej populácie určené počty čo najodlišnejších reťazcov. Mierou odlišnosti (diverzity) pritom je euklidovská vzdialenosť jednotlivých génov daného reťazca od referenčného reťazca. Referenčným reťazcom môže byť buď najúspešnejší jedinec populácie (s najmenšou hodnotou účelovej funkcie), alebo reťazec priemerných hodnôt génov populácie. O tom, ktoré reťazce budú kopírované, rozhoduje vektor *Nums*, ktorého prvky určujú počty vybraných reťazcov nasledovne: prvá hodnota určuje koľkokrát sa skopíruje najviac odlišný reťazec do cieľovej populácie, druhá hodnota určuje koľkokrát sa skopíruje 2. najodlišnejší reťazec v poradí do cieľovej populácie atď.

Syntax:

Newpop=seldiv(Oldpop,Objpop,Nums,sw)

Newpop - nová (výstupná) populácia

Oldpop - stará (vstupná) populácia

Objpop - vektor hodnôt účelovej funkcie starej populácie

Nums - vektor, ktorého prvky určujú, koľkokrát sa reťazec na príslušnom poradí podľa miery diverzity skopíruje do novej populácie

sw - prepínač: 0 - vzťažný reťazec pri určovaní miery diverzity tvorí aritmetický priemer génov všetkých reťazcov

1 - vzťažný reťazec pri určovaní diverzity je najlepši jedinec

Príklad:

A = [2 2 2 2 2
4 2 1 3 3
2 2 2 3 2
0 2 0 3 2
8 2 5 2 7
2 2 9 2 2
3 3 3 3 3
0 2 2 2 1]

uf=[1 5 2 4 3 7 6 10]

nums=[2 1 1]

B=seldiv(A,uf,nums,1)

Vektor miery diverzity jednotlivých reťazcov voči najlepšiemu reťazcu

[2 2 2 2 2] je:

d = [0 2.646 1 1.732 8.367 7 2.236 2.236]

B = [8 2 5 2 7
8 2 5 2 7
2 2 9 2 2
4 2 1 3 3]

selrand - náhodný výber jedincov

Charakteristika:

Funkcia náhodne vyberie zo vstupnej populácie do výstupnej populácie určený počet reťazcov.

Syntax:

Newpop=selrand(Oldpop,Objpop, Num);

Newpop - nová (výstupná) populácia

Oldpop - stará (vstupná) populácia

Objpop - vektor hodnôt účelovej funkcie starej populácie

Num – počet náhodne vybraných reťazcov

Príklad:

```
A = [ 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 ]
```

B=selrand(A,5)

```
B = [ 2   2   2   2   2
      6   6   6   6   6
      5   5   5   5   5
      2   2   2   2   2
      3   3   3   3   3 ]
```

selsort - výber a zoradenie jedincov podľa úspešnosti

Charakteristika:

Funkcia vyberie zo vstupnej populácie do výstupnej populácie určený počet najúspešnejších reťazcov, ktoré navyše zotriedi podľa hodnoty účelovej funkcie od najúspešnejšieho po najhoršieho. Najúspešnejšími reťazcami sú chápané jedince s najmenšími hodnotami účelovej funkcie.

Syntax:

Newpop=selsort(Oldpop,Objpop,Num);

Newpop - nová (výstupná) populácia

Oldpop - stará (vstupná) populácia

Objpop - vektor hodnôt účelovej funkcie starej populácie

Num – počet vybraných najúspešnejších reťazcov

Príklad:

```
A = [ 4  4  4  4  4
      2  2  2  2  2
      6  6  6  6  6
      3  3  3  3  3
      1  1  1  1  1
      5  5  5  5  5 ]
```

uf=[4 2 6 3 1 5]

pocet=4

B=selsort(A,uf,pocet)

```
B = [ 1  1  1  1  1
      2  2  2  2  2
      3  3  3  3  3
      4  4  4  4  4 ]
```


selsus - výber jedincov pomocou váhovaného ruletového kolesa

Charakteristika:

Funkcia vyberie zo vstupnej populácie do výstupnej populácie určený počet reťazcov pomocou výberu, ktorého ekvivalentom je otáčanie váhovaným ruletovým kolesom. Váhované ruletové koleso je rozdelené na kruhové výseky, z ktorých každý je priradený jednému reťazcu a jeho veľkosť je nepriamo úmerná hodnote jeho účelovej funkcie (resp. priamo úmerná jeho úspešnosti). Úspešnejšie reťazce majú väčšiu šancu byť vybrané.

Syntax:

Newpop=selsus(Oldpop,Objpop,Num);

Newpop - nová (výstupná) populácia

Oldpop - stará (vstupná) populácia

Objpop - vektor hodnôt účelovej funkcie starej populácie

Num – počet vybraných najúspešnejších reťazcov

Príklad:

```
A = [ 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 ]
```

```
uf=[1 2 3 4 5 6]
```

```
B=selsus(A,uf,4)
```

```
B = [ 5   5   5   5   5
      1   1   1   1   1
      3   3   3   3   3
      2   2   2   2   2
      2   2   2   2   2 ]
```

shake - náhodné premiešanie poradia reťazcov v populácii

Charakteristika:

Funkcia vráti novú populáciu, v ktorej budú mať reťazce pôvodnej populácie náhodne zmenené poradia. Intenzita premiešania môže byť nastavená od hodnoty 0 - žiadne premiešanie až po hodnotu 1 - maximálne premiešanie.

Syntax:

Newpop=shake(Oldpop,rate);

Newpop - matica populácie so zmeneným poradím reťazcov

Oldpop - matica populácie s pôvodným poradím reťazcov

rate - intenzita premiešania (od 0 do 1)

Príklad:

```
A = [ 1  1  1  1
      2  2  2  2
      3  3  3  3
      4  4  4  4
      5  5  5  5
      6  6  6  6 ]
```

B=shake(A, 0.9)

```
B = [ 2  2  2  2
      6  6  6  6
      3  3  3  3
      1  1  1  1
      5  5  5  5
      4  4  4  4 ]
```

swapgen - mutácia poradia génov v reťazci (permutačné kódovanie)

Charakteristika:

Funkcia zmutuje poradie génov v náhodne vybraných reťazcoch populácie. Početnosť mutácií v populácii je úmerná parametru *rate* (z rozsahu od 0 do 1). Pri mutácii reťazca je vymenené poradie dvoch náhodne vybraných génov.

Syntax:

Newpop=swapgen(Oldpop,rate)

Newpop - nová, zmutovaná populácia

Oldpop - stará populácia

rate - určuje početnosť výskytu mutácií v populácii

Príklad:

```
A=[ 1 2 3 4 5 6 7
    7 6 5 4 3 2 1
    1 1 1 2 2 2 2
    1 2 3 4 5 6 7
    7 7 7 7 7 5 5]
```

B=swapgen(A, 0.5)

```
B=[ 1 2 3 4 5 6 7
    7 2 5 4 3 6 1
    1 1 1 2 2 2 2
    1 2 3 4 5 6 7
    7 7 7 7 5 7 5]
```

swappart - vzájomná výmena poradia dvoch častí reťazcov **(permutačné kódovanie)**

Charakteristika:

Funkcia náhodne vyberie niekoľko reťazcov, u každého reťazca vymení poradie jeho dvoch častí, ktoré vzniknú rozdelením reťazca v náhodnej pozícii. Početnosť takto modifikovaných reťazcov v populácii je úmerná parametru *rate* (z rozsahu od 0 do 1).

Syntax:

Newpop=swappart(Oldpop,rate)

Newpop - nová, zmenená populácia

Oldpop - stará populácia

rate - určuje početnosť výskytu tejto modifikácie v populácii

Príklad:

```
A=[ 1 2 3 4 5 6 7
    7 6 5 4 3 2 1
    1 1 1 2 2 2 2
    1 2 3 4 5 6 7
    7 7 7 7 7 5 5]
```

B=swappart(A, 0.5)

```
B=[ 6 7 1 2 3 4 5
    7 6 5 4 3 2 1
    2 2 2 1 1 1 2
    1 2 3 4 5 6 7
    7 7 7 7 7 5 5]
```