Gena Formelsammlung v.62

Extremwertbestimmung

<u>*Extremwertbestimmung einer Variablen:*</u>

$y=f(x)$ 2-mal diffbar

Notwendiges Kr. $f'(x_0)=0$ $\begin{Bmatrix} f''(x_0)<0\ Max\ in\ x_0 \\ f''(x_0)>0\ Max\ in\ x_0 \end{Bmatrix}$

Hinreichendes K. $f''(x_0)\neq 0$

<u>*Extremwertbestimmung 2er Variablen:*</u>

$y=f(x,y)$ $f(x,y)$ besitzt part. Ableitungen 2. Ordnungen

Notw. Kr. $\begin{Bmatrix} f_x(x_0,y_0)=0 \\ f_y(x_0,y_0)=0 \end{Bmatrix}$

Hinr. Kr. $\Delta(x_0,y_0)=f_{xx}(x_0,y_0)*fyy(x_0,y_0)-f_{xy}^2(x_0,y_0)>0 \begin{Bmatrix} f_{xx}(x_0,y_0)<0\ Max\ in(x_0,y_0) \\ f_{xx}(x_0,y_0)>0:\ Min\ in(x_0,y_0) \end{Bmatrix}$

<u>*Gradientenmethode:*</u>

$grad\ f=\begin{pmatrix} f_x \\ f_y \end{pmatrix}$

Bilde: $\vec{x}_{i+1}=\vec{x}_i+h*grad\ f$

$x_i \to Max.$

$h=$ Schrittweite

Mutations Selektion Verfahren

1) Wähle Anfangschromosom $Z_0 \in S$
2) Verändere $Z_0 \to Z_1$ (Mutation)
3) Falls Z_1 den Ristriktionen (falls vorhanden) nicht genügt weiter bei 2)
4) $F(Z_1)>F(Z_0) ==> Z_0 \leftarrow Z_1$
5) Abbruch falls Abbruchkriterium erfüllt ist
6) Weiter bei 2)

Bem.: Ist ein minimum gesucht so ersetze $f(z)=-f(z)$

Lösung von Gleichungssystem

Bsp:

$\begin{Bmatrix} x^2+y^2=13 \\ x^3-y^2=-19 \end{Bmatrix} \begin{Bmatrix} x^2+y^2-13=0 \\ x^3-y^2+19=0 \end{Bmatrix}$

$F(x,y)=-((x^2+y^2-13)^2+(x^3-y^2+19)^2)\geq 0$

<u>*Algorithmus:*</u>

(1) Wähle $Z_0=(x_0,y_0)$ beliebig

(2) $x_1=x_0+\delta_1(\delta_1\ Zufallszahl -0{,}5\leq\delta_1\leq 0{,}5)$

$y_1=y_0+\delta_1(\delta_1\ Zufallszahl -0{,}5\leq\delta_1\leq 0{,}5)$

(3) $F(x_1,y_1)>F(x_0,y_0) ==> \begin{Bmatrix} x_0\leftarrow x_1 \\ y_0\leftarrow y_1 \end{Bmatrix}$

(4) weiter bei (2)

Nullstellenberechnung

Man Löse:

$y=x^4-7x^3+8x^2-2x+1=0$

$F(x)=-y^2\leq 0$

<u>*Algorithmus:*</u>

(1) Wähle Startwert x_0

(2) Mutation $x_1=x_0+\delta(\delta\ Zufallszahl\ mit -0{,}5\leq\delta_1\leq 0{,}5)$

(3) $F(x_1)>F(x_0) ==> x_0\leftarrow x_1$

(4) Fahre fort bei (2)

Lösung von Differntialgleichungen

$\begin{Bmatrix} (y-x)*y''+\sin^2(x)=0 \\ y(0)=0\ ,\ y(1)=1{,}841147 \end{Bmatrix}$ RWP $0\leq x\leq 1$

<u>Lösung</u>: Polynomansatz:

$y=a_0+a_1x+...+a_nx^n$

Ansatz: $y(0)=0=a_0$

$n=4$: $y=ax+bx^2+cx^3+dx^4$

$y(1)=1{,}84117=a+b+c+d$

$a=1{,}84117-b-c-d$

Ansatz: $y=(1{,}84117-b-c-d)x+bx^2+cx^3+dx^4$

geg.: $h=0{,}1$

Fitnessfunktion:

$$F(b,c,d)=-\sum_{j=1}^{10}[(y(jh)-jh)*y''(jh)+\sin^2(jh)]^2\leq 0$$

mit $y''(x)=2b+6cx+12dx^2$

$y=ax+bx^2+cx^3+dx^4$ Lsg. von oben $\to F(b,c,d)=0$

<u>*Algorithmis:*</u>

(1) Wähle $b_0, c_0 d_0$ beliebig

(2) Mutation: $\{b_0\to b_1; c_0\to c_1; d_0\to d_1\}$

(3) Falls $F(b_1,c_1,d_1)>F(b_0,c_0,d_0) ==> \begin{Bmatrix} b_0\leftarrow b_1 \\ c_0\leftarrow c_1 \\ d_0\leftarrow d_1 \end{Bmatrix}$

(4) Fahre fort mit (2)

Lösungen von Gleichungssystemen

Bspl.: lin. Gleichungssystem

$$\begin{pmatrix} M_{1,1} & ... & ... & M_{1,40} \\ \vdots & \vdots & \vdots & \vdots \\ M_{40,1} & ... & ... & M_{40,40} \end{pmatrix} * \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{40} m_{1,}j*x_j-r_1 \\ \vdots \\ \sum_{j=1}^{40} m_{40,}j*x_j-r_j \end{pmatrix}$$

$$Fitness: F(\vec{x})=-\frac{1}{40}\left\{\sum_{i=1}^{40}\left|\sum_{j=1}^{40} m_i,j*x_j-r_i\right|\right\}\leq 0$$

Es gilt: $F(\vec{x})=0 <==> w\ \vec{x}$ Lsg. von oben

<u>*Algorithmus:*</u>

(1) $\vec{x}_0$ Anfangsvektor

(2) Mutation $\vec{x}_1=x_0+\delta$ δ Zufallszahl

(3) $F(\vec{x}_1)>F(\vec{x}_0) ==> \vec{x}_0\leftarrow\vec{x}_1$

(4) Weiter bei (2)

Lineare Optimierung

Maschine produziert:

A: 2000 Stück/h
B: 1500 Stück/h
C: 1000 Stück/h
D: 1200 Stück/h

Gesamtzahl soll Maximum sein.

align ***Randbedingungen***:

(1) Maschine läuft 16 Std/Tag
(2) Typ A höchstens 700/Tag
(3) Typ B mind. 4000 Stück/Tag
(4) Anzahl $D\geq$ Anzahl b

Tagesproduktion:

$a=$ Zeit der Anlage für Typ A/Tag
$b=$ Zeit der Anlage für Typ B/Tag
$c=$ Zeit der Anlage für Typ C/Tag
$d=$ Zeit der Anlage für Typ D/Tag

Nebenbedingung:

(1) $a+b+c+d\leq 16$
(2) $2000*a<7000\to a\leq 3{,}5$
(3) $1500*b\geq 4000\to b\geq 2{,}66$
(4) $1200*d\geq 1500*b\to d\geq\frac{1{,}5}{1{,}2}b$
(5) $a\geq 0; a\geq 0; b\geq 0; c\geq 0; d\geq 0$

<u>***Algorithmus***</u>:

(1) Wähle Anfangschromosom (a,b,c,d) (erfüllt die Nebenbediengung)
(2) Mutation $(a,b,c,d\to a',b',c',d')$
(3) Falls $F(a',b',c',d)$ Nebenbed. nicht erfüllt zurück nach (2)
(4) $F(a',b',c',d')>F(a,b,c,d) ==> \{a\leftarrow a'; ...; l\ d\leftarrow d'\}$
(5) weiter bei (2)

<u>nichtlineare Optimierung:</u> Algorithmus siehe oben!

TSP

Fitness:

$Z=F(a,b,c,d)=2000*a+1500*b+1000*c+1200*d$

TSP: Bspl.: Orte (1,2,3,4,5) durch 5 Punkte (x,y) gegeben

Chromosom: $S=5-$ Permutation (z.B $S=2,3,5,1,4$)

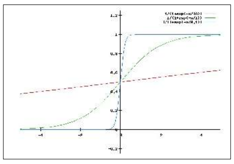
Mutation: zufällige Vertauschung zweier Zahlen $((2,3,5,1,4)\to(2,1,5,3,4))$

Fitness: $f(S)=\sum_{i=1}^{n}(x_i-x_i+1)^2+(y_i-y_i+1)^2$

gesucht: Chromosom S mit $F(S)=-f(s)=Max.$

Simulated Annealing

(1) Wähle Anfangschromosom $X\in M$ und $T>0$
(2) Mutation $x\to x'$
(3) $r=F(x')-F(x)$, F Fitness
(4) Berechne $p(r)=\frac{1}{1+e^{-\frac{r}{T}}}$
(5) wähle Zufallszahl $0\leq Z\leq 1$: $Z\leq p(r) ==> x\leftarrow x'$
(6) verkleinere T (z.B. $T=T*0{,}99$)
(7) weiter bei (2)



Treshold Accepting

(1) Wähle Anfangschromosom $X\in M$ und Threshold $T>0$
(2) Mutation $x\to x'$
(3) $r=F(x)-T$
(4) $F(x')\geq r ==> x\leftarrow x'$
(5) verkleinere T (z.B. $T=T*0{,}99$)
(6) weiter bei (2)

Sinflut Methode

(1) Wähle Anfangschromosom $X\in M$ und eine Schranke
(2) Mutation $x\to x'$
(3) $F(x')>T ==> x\leftarrow x'$
(4) vergrößere T um ein Inkrement (z.B. $T=T+0{,}001$)
(5) weiter bei (2)

Evolution:

(1) Gegeben Ökologie mit Lebewesen
(2) Mutation der DNS
(3) Rekombination = Kreuzung = Reproduktion
(4) Fitness: Überlebensfähigkeit
(5) Selektion: die Besseren überleben
(6) weiter bei (2)

Der Genetische Algorithmus Unterschied zum Mutations S. V.

	<u>Mutation Selektion Verfahren</u>	<u>Genetischer Algorithmus</u>
(1)	Chromosom X	Menge von Chromosomen bzw. Induvidien (Population)
(2)	Mutation $x\leftarrow x'$	Mutation
(3)	$F(x')>F(x) ==> x\leftarrow x'$	Rekombination
(4)	weiter bei (2)	Selektion
(5)	-	weiter bei (2)
		Mutation, Rekombination, Selektion genetische Operationen

<u>Der Genetische Algorithmus</u>

(1) Wähle Anfangspopulation P mit N Induvidien (= Bitstrin mit gleicher Länge)
(2) Berechne $F(x)$ für alle $x\in P$, $F=$ Fitnessfunktion
(3) Führe eine der folgenden Operationen durch:
Rekombination (mit Wahrscheinlichk. $p(c)$) ($c=$ crossover)
Mutation (mit Wahrscheinlichk. $p(M)$)
Selektion (mit Wahrscheinlichk. $p(S)$)
(4) neue Individium $\to P'$
(5) weiter bei (3) bis $card(P')=card(P)=N$ ($card(M)=$ Anz. der Elemente von M)
(6) $P\leftarrow P'$ und $P'=\emptyset$
(7) weiter bei (2), falls A
(8) Ermittle Individium mit optimaler Fitness als Lsg.

Erfahrungswerte:

zu(1) $50\leq N\leq 500$

zu(2) $p(M)\leq\frac{1}{N}$, $p(c)\approx 0{,}6$

$p(S)=1-p(M)-p(c)$

Auswahlverfahren / Roulett

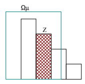
$f = Fitness: F(K) = f(1) + f(2) + ... + f(k)\ 1 \le k \le N \quad (f(1) = Fitness\ vom\ 1.\ Individium)$
Wähle Zufallszahl $Z \in \mathbb{N}$ mit $0 \le Z \le F(N)$
Wenn $F(K-1) \le Z < F(K) ==>$ Individium K gilt als ausgewählt

Auswahlverfahren / Turniermethode

(1) Wähle zufällig 2 Individuem aus
(2) Wähle das Individium mit der höchsten Fitness

Auswahlverfahren / (N,my) Selektion

(1) Sortiere Individuen nach absteigender Fitnesswerten
(2) Wähle eine Zufallszahl $Z \in \mathbb{N}$ mit $1 \le Z \le \mu \quad (\mu \le N)$
(3) Indivdium Z gilt als ausgewählt (d.h. von den μ besten per Zufall ausgewählt)



Auswahlverfahren / Vergleichende Aussagen

(N,μ) Selektion:	schwache Fitness Individium keine Chance → harte Selektionsbed.
Roulettverfahren:	schwache Fitness Indivium geringe Chance → weiche Selektionsbed.
hart:	schnelle Konvergenz, i.a. nur lokales Extremum
weiche:	langsame Konvergenz, globales Extremum möglich

Ein Punkt Rekombination

per Zufall

$$\left.\begin{matrix} 0\ 1\ 1\ 0\,|\,\underbrace{1\ 1\ 0}_{} \\ 1\ 1\ 0\ 0\,|\,\underbrace{0\ 1\ 1}_{\text{vertauscht}} \end{matrix}\right\} \Rightarrow \begin{matrix} 0\ 1\ 1\ 0\,|\,0\ 1\ 1 \\ 1\ 1\ 0\ 0\,|\,1\ 1\ 0 \end{matrix}$$

Eltern Nachkommen

Zwei Punkt Rekombination

per Zufall

$$\left.\begin{matrix} 0\ 1\ 1\,|\,0\ 1\ 0\ 1\,|\,1\ 0\ 1\ 1 \\ 1\ 0\ 0\,|\,\underbrace{0\ 0\ 1\ 1}_{\text{vertauscht}}\,|\,1\ 1\ 1\ 0 \end{matrix}\right\} \Rightarrow \begin{matrix} 0\ 1\ 1\,|\,0\ 0\ 1\ 1\,|\,1\ 0\ 1\ 1 \\ 1\ 0\ 0\,|\,0\ 1\ 0\ 1\,|\,1\ 1\ 1\ 0 \end{matrix}$$

Eltern Nachkommen

Gleichmäßige Rekombination

0 1 1 0 1 1 0 0 0 1 0 1
1 1 0 0 1 1 0 1 1 0 1 0 Eltern
0 1 0 1 1 0 0 1 0 1 0 0 Template durch Zufall generiert

1 1 0 0 1 1 0 0 1 1 1 0
0 1 1 0 1 1 0 1 0 0 0 1 Nachkommen

Vertausche Bit, wenn Template Vektorkomponente = 0
(Vertausche Bit = Elternteil <-> Elternteil)

Intermiediare Rekombination

0,6	4,3	2,4	−1,8	5,6
2,7	9,3	−6,4	1,8	0,6
1,7	6,8	2,0	0	3,4 ← arithm. Mittel

PMX Rekombination

Ein Element darf nicht 2−mal vorkommen

$$\begin{array}{lccc|ccc|ccc}
\text{Eltern:} & (1 & 4 & 2 & 3 & 7 & 6 & 9 & 5 & 8\ ) \\
 & (3 & 7 & -8 & 6 & 1 & 9 & 2 & 4 & 5\ ) \\
\hline
 & (\underline{1} & 4 & 2 & 6 & \underline{1} & \underline{9} & 2 & 5 & 8\ ) \\
 & (\underline{3} & \underline{7} & 8 & \underline{3} & \underline{7} & 6 & 2 & 4 & 5\ )
\end{array}$$

bleiben unverändert

Nachkommen:
(7 4 2 6 1 9 3 5)
(9 1 8 3 7 6 4 5)

Mutation

Population:
1. 1011011
2. 0110111
3. 1000110
⋮ …
N 1001011

Algorithmus:
(1) Bestimme Zufallszahl $1 \le k \le N$ und $1 \le n \le M$
(2) Vertausche Bit n in Individium k

Bsp.: per Zufall
100 |0110| 01011
100 |$\underbrace{1001}_{\text{Umkehrung}}$| 01011

Bemerkungen:
(1) Mutationen führen aus lokalen Extrema weg, können sinnvolle Info zerstören.
(2) Erfahrung: Mutationsrate $p(M) \le \frac{1}{N}$

Dekodierung

Lösung eines Gleichungssystems mittels GA

Lösung: $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad x_i \in \mathbb{R}$

GA: $\underbrace{\boxed{100100}}_{x_1}\underbrace{\boxed{101110}}_{x_2}\underbrace{\boxed{00111}}_{x_3}$

Umwandlung eines Bitstringes in eine reele Zahl aus [a,b]

$10010 \to e \in [a,b] \subset \mathbb{R}$
(1) Gegeben $b_1, b_{2,..}, b_n$ ein Bitstring mit $b_i \in \{0,1\}$
(2) Dualwert $d = b_1 2^{n-1} + b_2 2^{n-2} + ... + b_n 2^0$
(3) Reele Zahl aus $[a,b]$ $\boxed{r = a + (b-a) * \frac{d}{2^n - 1}}$

$11011 \to [1,-1]$
(1) $11011\ n = 5$
(2) $d = 2^4 + 2^3 + 2^1 + 2^0 = 27$
(3) $r = -1 + \frac{27}{31} * 2 = 0{,}741925$

Anwendungen

Pipline Systeme
Angiografie

Angigrafie:
Problem: Patient bewegt sich zwischen den beiden Röntgenbildern.
Verfahren: (x,y) Pixel von Bild 1; (x',y') Pixel von Bild 2
Setze: $x'' = a + bx + cy + dxy$
$y'' = c + fx + gy + hxy$
Mittels GA $a, b, c, .., h$, so bestimmen das Bilddifferenz $(x',y') - (x'',y'')$ minimal.

<u>Evolutionsstrategien:</u>

Das Verfahren:
Grundverfahren wie bei genetischen Algorithmen
Population
Mutationen
Rekombinationen
Selektionen
Fitness

<u>Abweichung:</u> Population Chromosom reelwertige Vektoren
<u>Hauptgwicht:</u> Mutationen

Normalverteilte Zufallszahlen

<u>Normalverteilung $N(0,\sigma)$</u>
Erwartungswert $\mu = 0$
Standardabweichung σ
<u>Umkehrfunktion:</u>
$x = \pm\sigma\sqrt{-\ln(\sigma * y * \sqrt{2\pi})} = f^{-1}(y)$

rnd1 = RANDOM [0,1]

<u>Erzeugung normalverteilter Zufallszahlen:</u>
(1) Wähle (gleichverteilte) Zufallszahl $0 \le y \le \frac{1}{\sigma\sqrt{2\pi}}$
(2) Bestimme $x = \pm\sigma\sqrt{-\ln(\sigma * y * \sqrt{2\pi})}$
(3) Bestimme gleichverteilte Zufallszahl $Z \in [-x, x]$
zu (3) $Z = a + rnd1 * (b-a) = -x + rnd1 * (x - (-x)) =$
$= -x + rnd1 * 2x$
einfacher: $Z = ((2 * rnd1) - 1) * \sigma * \sqrt{(-\ln(rnd1))}$

Mutation bei Evolutionsstrategien

$x = (x_{1,...}, x_n),\ x_i \in \mathbb{R}^n$
<u>Mutation:</u>
$x_i^{neu} = x_i^{alt} + Z_i$, wobei $Z_i\ N(0,\sigma)$ − verteilt
$\sigma_i^{neu} = \sigma_i^{alt} * e_i^Z$, wobei $Z_i\ N(0,\tau)$ − verteilte Zufallszahl mit $\tau \approx \frac{1}{\sqrt{2n}}$
n = Populationsgröße

Chromosom

$(x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n)$

Rekombination

Rekombination = arithmetisches Mittel
Bsp.: $(1,2,3,4) + (5,7,9,8) \to (3,4.5,6,6)$

Selektion

(μ, Γ) − Konzept: μ Eltern produzieren Γ Nachkommmen. Von den nachkommen überleben die μ besten.
$(\mu + \Gamma)$ − Konzept: μ Eltern produzieren Γ Nachkommen von allen überleben die μ besten.

Anwendungen

- Optimierung optischer Linsen
- Optimierung sozio−ökonomischer Systeme
- Regressionsanalyse
- Konstruktion von Tragwerken

<u>Genetische Programmierung Beispiele</u>

Ausgang: $y_1 = 1 + 2x - 4x^2$
$y_2 = 2 - x$
Rekombinattion: $y_1 = 1 + 2\boldsymbol{x} - 4x^2$
$y_2 = \boldsymbol{2 - x}$
Nachkommen: $y_3 = 1 + 2(2-x) - 4x^2$
Analog: $y_4 = 2 - 4x^2$
$y_5 = 1 + 2x - 4 + 2^2$

Ausdruck der $\sin(2x)$ gut erfasst

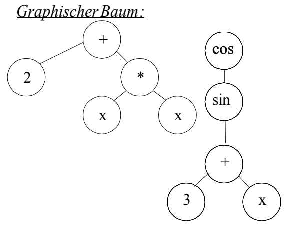
Anfangspopulation:	2. Generation: erzeugt durch Rekombinattion:
1	$\sin(x^2) * 2$
2	$\sin(x^2) * \cos(x)$
$1 * 2$	$\sin(\cos(x))$
$\sin(x)$	⋮
$\cos(x)$	⋮
$\sin(x^2)$	⋮
$\cos(x^2)$	⋮

Arithmetische Ausrücke

Präfixdarstellung:

$x + 2$	→ $+(x,2)$
$3 * 6$	→ $*(3,6)$
$3 * (x+2)$	→ $*(3, +(x+2))$
$((1+2)+3)$	→ $+(+(1,2),3)$
$(x + \sin(x^2) - 7)$	→ $-(+(x, \sin(*(x,x))),7)$
$\log(\sin(x-3)+1)$	→ $\log(+(\sin(-(x,3)),1))$

<u>Graphischer Baum:</u>



If Abfragen

Bsp: $if\ x>1\ then\ s:=\mathrm{sqrt(x)}\ else\ s:=1$

$if\ (x, 1, \mathrm{sqrt(x)}, 1)$

alg.: $if\ (a_1, a_2, a_3, a_4) = if\ a_1 > a_2\ then\ a_3\ else\ a_4$

Beispiel:

$if\ x^2>2\ then\ y:=3x-2\ else\ y:=0$

$z:=y+1$

$w:=4*z$

Präfixdarstellung:

$w:=*(+(if\ (*(x,x),2,-(*(3,x),2),0),1),4)$

Rekursionen

Bsp.: for i:=1 to 4 do

$s:=s+2$

Präfix: $rek(i,1,4s,+(s,2))$

allgemeine Rekursion: for a := b to c do

$d:=c$

$rek(a,b,c,d)$

Ähnlich Codierbar

- *Boolesche Ausdrücke*
- *Graphische Operationen*
- Vektor - Matrix Operationen
- *Komplexe Operationen*

Fitness

i	a	b	$f(i)$		Ausdruck	Präfix
1	0	0	1	$E[1]$	$a \wedge b$	$+(a,b)$
2	1	0	0	$E[2]$	$(a \wedge b) \vee a$	$-(+(a,b),a)$
3	0	1	1	$E[3]$	$\neg(a)$	$\wedge a$
4	1	1	1	$E[4]$	$\neg(a) \wedge (a \vee b)$	$+(\wedge a, -(a,b))$

Fitness:

$$r(j)=\sum_{j=1}^{4} |E[j]-f(i)|\ minimal$$

Genetische Operationen

Rekombination
Selektion
Mutatuion

Rekombination

Verfahren zwei Teilausdrücker werden vertauscht.

Bsp: $+(\boldsymbol{a},-(5,b))$

$-(+(3,c)-(\mathbf{2},\boldsymbol{a}))$ *Eltern*

$+(-(2,a),-(5,b))$

$-(+(3,c),a)$ *Nachkommen*

Selektion

Ziel: *Individuen mit hoher Fitness* $\rightarrow$ *Population*

Individuen mit niedriger Fitness $\rightarrow$ *löschen*

Turniermethode: *Suche 2 beliebige Individuen und wähle Individuen mit höherer Fitness aus*

Mutation

Beispiel: $+(-(w,x),*(+(1,2),c))$

Suche per Zufall den Anfang eines Ausruckes.

Ersetze Ausdruch durch einen Ausdruck, der per Zufall generiert wurde.

z.Bsp.:

$+(-(w,x),*(a,c))$

Implementierung

(1) *Wähle Anfangspopulation mit N Individuen*
(2) $p*N$ *Individuen: Rekombination*
(3) $q*N$ *Individuen: Mutation*
(4) $r*N$ *Individuen: Selektion*
(5) *erhalte neue Population*
(6) *weiter bei* 2

$p+q+r=1$

Bemerkungen: *zu* 1. *Anfangspopulation*

a) Festlegung: O = *Operationen*

T = *Terminale Symbole*

Bsp.: $O=\{+,i,*,/\}$

$T=\{1,a,x,-4\}$

Zufallszahl

gesucht Zufallszahl $Z \in [a,b]$

gegeben Zufallszahl

Vorgehensweise:

Zufallszahl dividieren durch größt möglichste Zahl

Zufallszahl nun im *Intervall* $[0,1]$

$\boldsymbol{Z=a+u*(b-a)}$ *Zufallszahl nun* im *gewünschten Intervall* $[a,b]$

Beispiel:

Zufallszahl = 32561 *soll* im *Intervall* $[0,60]$ *liegen*

$$t=0+\frac{32561}{99999}*60$$

$\boxed{Z=a+u*(b-a)}$