

Strategii evolutive

- Specific
- Structura generală
- Operatori de recombinaire
- Operatori de mutație
- Variante de selecție a supraviețuitorilor
- Adaptarea și auto-adaptarea parametrilor

Specific

Strategii evolutive: metode evolutive de rezolvare a problemelor de optimizare în domenii continue

Istoric: prima strategie a fost dezvoltată în 1964 de către Bienert, Rechenberg și Schwefel (studenți la Politehnica din Berlin) cu scopul de a optimiza structura unei conducte flexibile

Codificarea datelor: reală (elementele populației sunt vectori cu elemente reale)

Operator principal: mutație

Particularitate: auto-adaptarea parametrilor de control a mutației

Structura generală

Clasa de probleme:

Se caută x^* în $D \subset R^n$ cu
proprietatea că

$f(x^*) < f(x)$ pentru orice x din D

Populația constă din elemente
din D (vectori cu componente
reale)

Obs. O configurație este cu atât
mai bună cu cât valoarea lui f
este mai mică

Structura algoritmului:

Inițializarea populației

Evaluare populație inițială

REPEAT

generare urmași prin recombinare

modificarea urmașilor prin mutație

evaluarea urmașilor

selecția supraviețuitorilor

UNTIL <e satisfăcută o condiție de
oprire>

Criterii referitoare la
resurse
(ex: număr generații)

Criterii referitoare la
convergență
(ex: valoare f sau
măsură a diversității)

Operatori de recombinare

Specific: constă în generarea unui urmaș pornind de la un set de părinți

$$y = \sum_{i=1}^{\rho} c_i x^i, \quad 0 < c_i < 1, \quad \sum_{i=1}^{\rho} c_i = 1$$

Recombinare intermediară

(convexă): urmașul este o combinație liniară a părinților

$$y_j = \begin{cases} x_j^1 & \text{cu probabilitatea } p_1 \\ x_j^2 & \text{cu probabilitatea } p_2 \\ \vdots & \\ x_j^{\rho} & \text{cu probabilitatea } p_{\rho} \end{cases},$$

$$0 < p_i < 1, \quad \sum_{i=1}^{\rho} p_i = 1$$

Recombinare discretă: urmașul este constituit din componente selectate aleator de la părinți

Operatori de recombinare

Recombinare geometrică:

$$y_j = (x_j^1)^{c_1} (x_j^2)^{c_2} \dots (x_j^\rho)^{c_\rho}, \quad 0 < c_i < 1, \quad \sum_{i=1}^{\rho} c_i = 1$$

Observație: a fost introdusă de Michalewicz pentru probleme de optimizare cu restricții

Recombinare euristică:

$y = x^i + u(x^i - x^k)$ cu x^i element cel puțin la fel de bun ca și x^k

u – valoare aleatoare din $(0,1)$

Operatori de recombinare

Recombinare de tip SBX (Simulated Binary Crossover) [Deb, Agrawal, 1995]:

Idee: reproducerea proprietăților încrucișării cu un punct de tăietură

Observații:

- Încrucișarea cu un punct de tăietură conservă suma valorilor elementelor: $p_1 + p_2 = c_1 + c_2$
- Diferența dintre urmași poate fi mai mică (efect de contracție) sau mai mare (efect de extensie)
- Deb și Agrawal au studiat distribuția raportului $|c_1 - c_2| / |p_1 - p_2|$ și au propus o regulă de calcul specifică

Operatori de recombinare

Recombinare de tip SBX (Simulated Binary Crossover) [Deb, Agrawal, 1995]:

Mod de calcul:

$$c_1 = \bar{p} - \frac{\beta}{2}(p_2 - p_1) \quad c_2 = \bar{p} + \frac{\beta}{2}(p_2 - p_1)$$
$$\bar{p} = \frac{p_1 + p_2}{2}$$
$$\beta = \begin{cases} (2u)^{1/(d+1)} & \text{daca } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{1/(d+1)} & \text{daca } u > 0.5 \end{cases} \quad u = rand(0,1)$$

Parametrul d: permite controlul distantei dintre părinți și urmași

- Valori mari ale lui d conduc la urmași apropiați de părinți
- Valori mici ale lui d conduc la urmași mai îndepărtați de părinți

Operatori de mutație

Principiu general: mutația constă în perturbarea a elementelor populației prin adăugarea unei variabile aleatoare

$$x' = x + z$$

$$z = (z_1, \dots, z_n)$$

vector aleator cu medie 0 și

matrice de covarianța $C = (c_{ij})_{i,j=1,n}$

Specific: mutația prin perturbarea cu un vector aleator de medie 0 favorizează modificările mici ale configurației curente pe când mutația specifică algoritmilor genetici nu face distincție între perturbații mici și perturbații mari.

Operatori de mutație

Variante:

V1. Componentele vectorului de perturbare sunt variabile aleatoare independente ($E(z_i z_j) = E(z_i)E(z_j) = 0$) cu aceeași repartiție.

Exemple:

a) fiecare componentă a vectorului perturbație are repartiția uniformă în $[-s, s]$

b) fiecare componentă a vectorului perturbație are repartiția $N(0, s)$

Obs. Matricea de covarianță va fi o matrice diagonală de forma $C = \text{diag}(s^2, s^2, \dots, s^2)$ iar valoarea s este singurul parametru al mutației

Operatori de mutație

Variante:

V2. Componentele vectorului de perturbare sunt variabile aleatoare independente ($E(z_i z_j) = E(z_i)E(z_j) = 0$) însă cu repartiții având parametri diferiți.

Exemple:

- a) componenta z_i a vectorului perturbație are repartiția uniformă în $[-s_i, s_i]$
- b) fiecare componentă a vectorului perturbație are repartiția $N(0, s_i)$

Obs. Matricea de covarianță va fi o matrice diagonală de forma $C = \text{diag}(s_1^2, s_2^2, \dots, s_n^2)$ iar parametrii mutației sunt $s_1, s_2, \dots, s_n$

Operatori de mutație

Variante:

V3. Componentele vectorului de perturbare sunt variabile aleatoare corelate

Exemplu:

a) vectorul z are repartiția $N(0, C)$

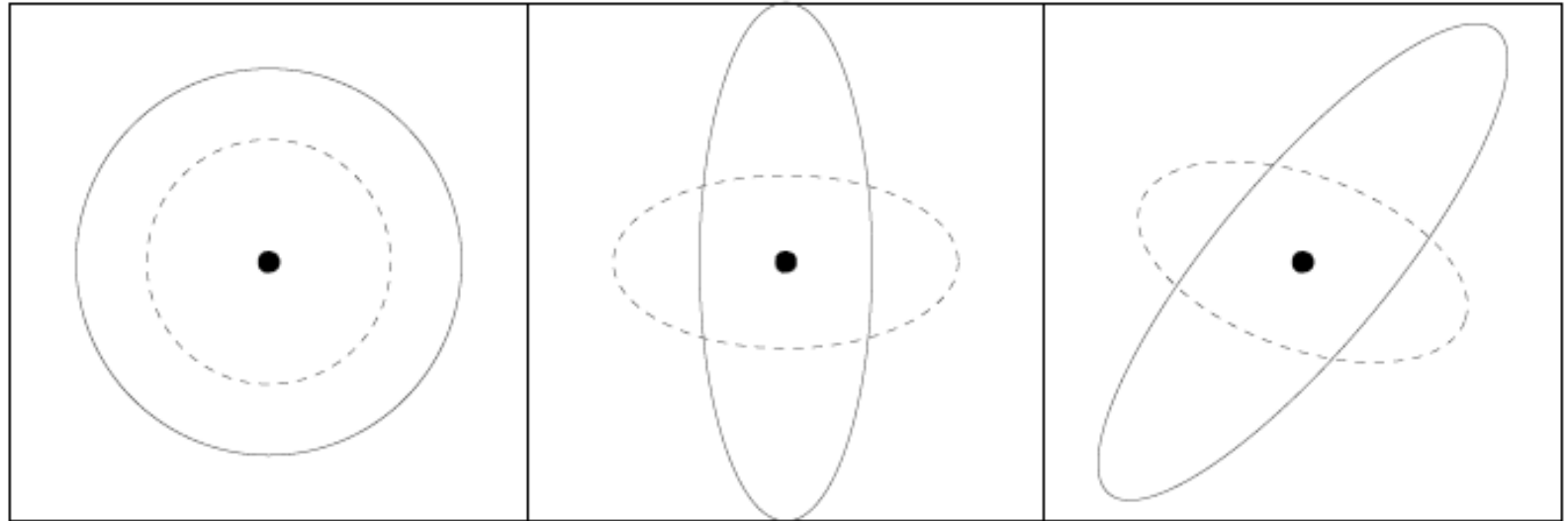
Obs. Matricea de covarianță va avea elemente nenule și în afara diagonalei principale astfel că vor fi $n(n+1)/2$ parametri de control ai mutației:

$s_1, s_2, \dots, s_n$ - pași de mutație

$a_1, a_2, \dots, a_k$ - unghiuri de rotație ($k=n(n-1)/2$)

$$c_{ij} = \frac{1}{2} \cdot (s_i^2 - s_j^2) \cdot \tan(2 a_{ij})$$

Operatori de mutație



$$\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{I}) \sim \mathbf{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \mathcal{N}(\mathbf{m}, \mathbf{D}^2) \sim \mathbf{m} + \mathbf{D} \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \mathcal{N}(\mathbf{m}, \mathbf{C}) \sim \mathbf{m} + \mathbf{C}^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Variante de mutatie

[Hansen, PPSN 2006]

Operatori de mutație

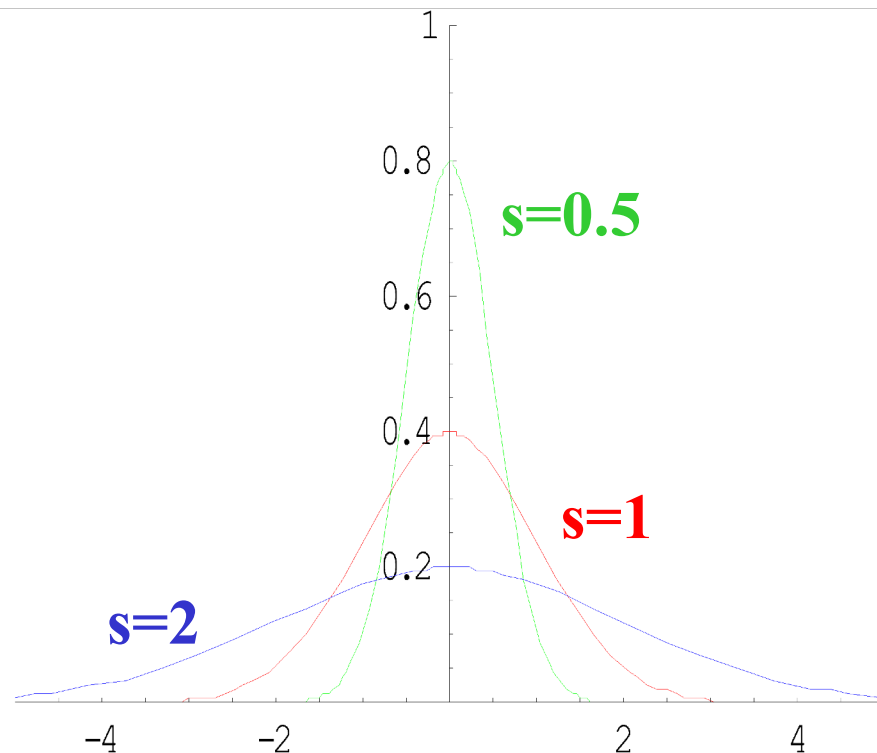
Problema: alegerea parametrilor de control ai mutației

Exemplu: perturbație de tip $N(0,s)$

- s mare $\rightarrow$ perturbație mare
- s mic $\rightarrow$ perturbație mică

Soluții:

- Metode euristice de adaptare (exemplu: regula 1/5)
- Autoadaptare (modificarea parametrilor prin încrucișare și mutație)



Operatori de mutație

Regula 1/5.

Regulă euristică dezvoltată pentru SE având perturbații independente caracterizate printr-un singur parametru s

Idee: s se ajustează în funcție de rata de succes a mutației

Rata (probabilitatea) de succes

$p_s = \frac{\text{nr. de mutații care conduc la îmbunătățiri ale configurației}}{\text{nr total de mutații}}$

- Obs.**
1. Rata de succes se estimează după cel puțin n mutații
 2. A fost dedusă pentru cazul unui singur element în populație

Operatori de mutatie

Regula 1/5.

$$s' = \begin{cases} s / c & \text{daca } p_s > 1/5 \\ cs & \text{daca } p_s < 1/5 \\ s & \text{daca } p_s = 1/5 \end{cases}$$

Studii teoretice asupra unor cazuri particulare de funcții obiectiv (ex: funcția sferă) au condus la concluzia că valori adecvate pentru c sunt $0.8 \leq c < 1$ (ex: $c=0.817$)

Operatori de mutație

Autoadaptare.

Idee:

- se extind elementele populației cu valorile parametrilor de control
- se aplică recombinare și mutație specifică și pentru parametrii de control
- vor fi favorizate valorile parametrilor de control care conduc la indivizi competitivi

Extinderea elementelor
populației în funcție de
tipul de perturbație

$$\bar{x} = (x_1, \dots, x_n, s)$$

$$\bar{x} = (x_1, \dots, x_n, s_1, \dots, s_n)$$

$$\bar{x} = (x_1, \dots, x_n, s_1, \dots, s_n, a_1, \dots, a_{n(n-1)/2})$$

Operatori de mutație

Etape:

- se modifică componentele corespunzătoare parametrilor de control
- se modifică componentele corespunzătoare variabilelor de decizie

Exemplu: perturbare bazată pe variabile aleatoare independente

$$\bar{x} = (x_1, \dots, x_n, s_1, \dots, s_n)$$

$$s'_i = s_i \exp(r) \exp(r_i),$$

Variabile cu repartiție lognormală

- asigură pozitivitatea lui s
- simetrică în jurul lui 1

$$r \in N(0, 1/\sqrt{2n}), r_i \in N(0, 1/\sqrt{2\sqrt{n}})$$

$$x'_i = x_i + s'_i z \quad \text{cu } z \in N(0, 1)$$

Operatori de mutație

Obs: Există și variante de mutație care nu folosesc perturbare bazată pe distribuție normală

Varianta propusă de Michalewicz (1996):

$$x'_i(t) = \begin{cases} x_i(t) + \Delta(t, b_i - x_i(t)) & \text{daca } u < 0.5 \\ x_i(t) - \Delta(t, x_i(t) - a_i) & \text{daca } u \geq 0.5 \end{cases}$$

$$\Delta(t, y) = y \cdot u \cdot (1 - t / T)^p, \quad p > 0$$

- a_i și b_i sunt limitele domeniului de valori corespunzătoare componentei x_i
- u este o valoare generată aleator în $(0,1)$
- t este indicatorul de iterație
- T este numărul maxim de iterații

Operatori de mutație

CMA – ES (Covariance Matrix Adaptation –ES) [Hansen, 1996]

Initialize $\mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} = \mathbf{I}$, and $\mathbf{p}_c = \mathbf{0}$, $\mathbf{p}_\sigma = \mathbf{0}$,
 set $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_{cov} \approx \mu_{eff}/n^2$, $\mu_{cov} = \mu_{eff}$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_{eff}}{n}}$,
 λ , and $w_i, i = 1, \dots, \mu$ such that $\mu_{eff} \approx 0.3 \lambda$, where $\mu_{eff} = \frac{1}{\sum_{i=1}^{\mu} w_i^2}$

While not terminate

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{z}_i, \quad \mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}), \quad \text{sampling}$$

$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \langle \mathbf{z} \rangle_{sel} \quad \text{where } \langle \mathbf{z} \rangle_{sel} = \sum_{i=1}^{\mu} w_i \mathbf{z}_{i:\lambda} \quad \text{update mean}$$

$$\mathbf{p}_c \leftarrow (1 - c_c) \mathbf{p}_c + \mathbb{1}_{\{\|\mathbf{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_{eff}} \langle \mathbf{z} \rangle_{sel} \quad \text{cumulation for } \mathbf{C}$$

$$\begin{aligned} \mathbf{C} \leftarrow & (1 - c_{cov}) \mathbf{C} + c_{cov} \frac{1}{\mu_{cov}} \mathbf{p}_c \mathbf{p}_c^T \quad \text{update } \mathbf{C} \\ & + c_{cov} \left(1 - \frac{1}{\mu_{cov}}\right) \mathbf{Z} \quad \text{where } \mathbf{Z} = \sum_{i=1}^{\mu} w_i \mathbf{z}_{i:\lambda} \mathbf{z}_{i:\lambda}^T \end{aligned}$$

$$\mathbf{p}_\sigma \leftarrow (1 - c_\sigma) \mathbf{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_{eff}} \mathbf{C}^{-\frac{1}{2}} \langle \mathbf{z} \rangle_{sel} \quad \text{cumulation for } \sigma$$

$$\sigma \leftarrow \sigma \times \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right) \right) \quad \text{update of } \sigma$$

Selecția supraviețuitorilor

Variante:

(μ, λ)

Din μ părinți se generează $\lambda > \mu$ urmași iar dintre aceștia se selectează μ supraviețuitori (obs: este important ca numărul urmașilor să fie mai mare decât cel al părinților)

$(\mu + \lambda)$

Din μ părinți se generează λ urmași iar din populația reunită a părinților și urmașilor se selectează μ supraviețuitori (selecție prin trunchiere). Este o selecție elitistă (cel mai bun element al populației nu se pierde)

Observație: dacă se utilizează recombinare atunci și numărul părinților (ρ) se poate specifica:

$(\mu / \rho + \lambda)$

$(\mu / \rho, \lambda)$

Selecția supraviețuitorilor

Cazuri particulare:

$(1+1)$ – dintr-un singur părinte se generează un urmaș care se acceptă doar dacă este mai bun decât părintele (similar cu algoritmi aleatori de tip Matyas sau Solis-Wets)

$(1,/+λ)$ – dintr-un singur părinte se generează mai mulți urmași prin mutație; cel mai bun dintre urmași este selectat (părintele poate participa sau nu la selecție)

$(μ+1)$ – dintr-o populație de $μ$ părinți se generează un fiu iar dacă acesta este mai bun decât cel mai slab element al populației atunci este selectat

Obs: varianta $(μ + λ)$ este mai orientată înspre exploatare decât $(μ, λ)$

Selecția supraviețuitorilor

Varianta ($\mu+1$) corespunde strategiei de tip “steady state” (sau asincronă)

Strategie generațională:

- La fiecare iterație se construiește o nouă populație de urmași
- Procesul de selecție este aplicat asupra populației de urmași sau asupra populațiilor reunite
- Prelucrare **sincronă**

Strategie steady state:

- La fiecare iterație se construiește un singur element care este asimilat în populația curentă dacă este suficient de bun
- Prelucrare **asincronă**

Variante ale SE

Strategii de tipul (μ, k, λ, ρ)

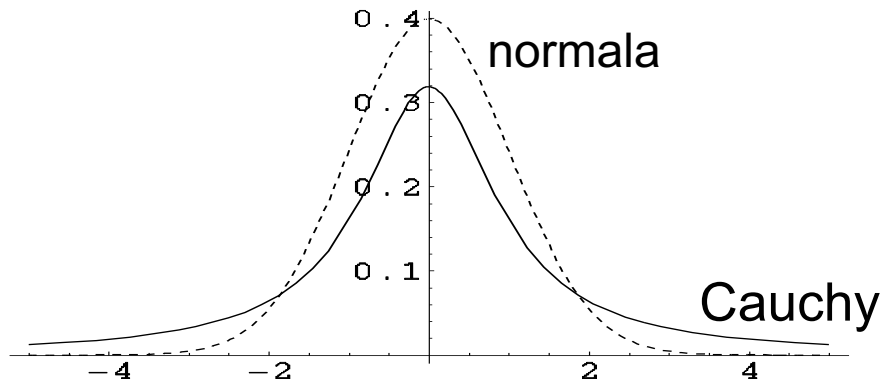
Durata de viață a elementelor este limitată la k generații (elementele care au supraviețuit de-a lungul a k generații sunt eliminate – cu excepția celui mai bun element al populației)

Folosește recombinare cu părinți

Strategii evolutive rapide:

Perturbarea este bazată pe repartiția Cauchy

$$\varphi(x) = \frac{s}{\pi(x^2 + s^2)}$$



Analiza comportării strategiilor evolutive

Criterii de evaluare:

Eficacitate:

- Valoarea funcției obiectiv după un anumit număr prestabilit de evaluări (nfe = number of function evaluation)

Eficiența:

- Numărul de evaluări de funcții necesare pentru ca funcția obiectiv să atingă o anumită valoare (aproximare cu o anumită acuratețe)

Rata de succes

- Raportul dintre numărul de rulări în care algoritmul atinge valoarea dorită și numărul total de evaluări

Sumar SE

Reprezentare	Vectori cu componente reale
Recombinare	Discretă sau intermediară
Mutație	Perturbare aleatoare (uniformă, normală Cauchy)
Selecție părinți	Uniform aleatoare sau sistematică
Selecție supraviețuitori	(μ, λ) sau $(\mu + \lambda)$
Particularitate	Auto-adaptare a parametrilor de control a mutației