

# Evolucijske strategije i CMA-ES

15. prosinca 2022.

## 1 Evolucijska strategija

Evolucijska strategija je izvorno predstavljena kao jednostavna metaheuristika za rješavanje proizvoljnih optimizacijskih problema. Kao i većina ostalih evolucijskih algoritama, temelji se na skupu potencijalnih rješenja optimizacijskog problema. Skup rješenja se obično naziva populacija, a elementi skupa jedinke.

U svakom koraku algoritma, iz trenutne populacije rješenja (roditelji) nekim postupkom generira se određeni broj novih jedinki (djeca). Potom se iz dobivenog skupa jedinki odabiru one koje će „preživjeti” i predstavljati roditeljske jedinke u sljedećoj iteraciji (nova generacija), a odabir je obično deterministički i temelji se na boljoj vrijednosti funkcije cilja. Veličina (roditeljske) populacije se obično označava s  $\mu$ , a broj novostvorenih jedinki djece s  $\lambda$ . Prilikom odabira jedinki za novu generaciju, razlikuju se dva pristupa: u **plus strategiji**, selekcija za novu populaciju obavlja se i nad skupinom roditelja ( $\mu$ ) i nad skupinom njihove djece ( $\lambda$ ), pa ova strategija ima svojstvo *elitizma* (očuvanje trenutno najboljeg rješenja). Ovakve strategije označavaju se kao  $(\mu + \lambda)$ -ES. U **zarezu strategiji**, jedinke za novu generaciju biraju se samo iz skupa djece, što podrazumijeva da mora vrijediti  $\lambda \geq \mu$ ; oznaka ove strategije je  $(\mu, \lambda)$ -ES. U takvom načinu odabira, može se dogoditi i gubitak trenutno najboljeg rješenja, no moguće je postići manju vjerojatnost zapinjanja u lokalnom optimumu. Tijek rada općenite inačice algoritma opisan je kao algoritam 1.

S obzirom na odnos parametara i strategiju odabira, neke inačice algoritma mogu biti:

---

**Algoritam 1** Osnovna inačica evolucijske strategije

---

```
generiraj populaciju rješenja ( $\mu$ )
ponavljaj
    stvori  $\lambda$  jedinki djece
    odaberi  $\mu$  najboljih jedinki za novu generaciju
do uvjeta zaustavljanja
```

---

- $(1 + 1)$ -ES: od jednog roditelja stvara se jedno dijete te od obje jedinke zadržava bolja;
- $(\mu + 1)$ -ES: od  $\mu$  roditelja stvara se jedno dijete koje se dodaje u skup roditelja, te se potom iz tog skupa izbacuje najlošija jedinka;
- $(1 + \lambda)$ -ES: od jednog roditelja stvara se  $\lambda$  djece te uzima samo najbolja jedinka za sljedeću generaciju;
- $(\mu + \lambda)$ -ES: od  $\mu$  roditelja stvara se  $\lambda$  djece koji se ubacuju u skup roditelja te se bira  $\mu$  najboljih jedinki za sljedeću generaciju.

U slučaju zarez strategija, parametar  $\lambda$  svakako treba biti strogo veći od jedan te veći od  $\mu$ .

U osnovnoj inačici algoritma nije definirano kako se stvara skup jedinki djece, no podrazumijeva se da njihova svojstva ovise o svojstvima roditeljske populacije. Uobičajen pristup je odabir slučajnog roditelja te primjena nekog operatora perturbacije, mutacije ili lokalne promjene.

**Primjer.** Ovako opisan jednostavan algoritam može se primijeniti u optimizaciji zadane funkcije cilja tako da svaka jedinka predstavlja jednu točku prostora pretraživanja. U postupku stvaranja novih jedinki, uzima se slučajno odabrana jedinka iz roditeljske populacije te joj se dodaje slučajno generiran pomak, najčešće po normalnoj raspodjeli s proizvoljno zadanom standardnom devijacijom,  $\mathcal{N}(0, \sigma)$ . Iz dobivenog skupa rješenja odabire se  $\mu$  najboljih jedinki za novu generaciju.

## 2 Evolucijska strategija s optimizacijom parametara raspodjele

Ako se ograničimo na kontinuiranu optimizaciju, ES može se modelirati tako da optimira parametre *raspodjele* po kojoj se ravna funkcija cilja. Umjesto optimiranja položaja minimuma kao točke u  $n$ -dimenzijskom prostoru, u ovoj inačici cilj je optimirati parametre raspodjele koja bi trebala dobro opisivati položaj optimuma funkcije cilja.

Formalno, tražimo raspodjelu  $p_{\theta}(\mathbf{x})$  vektora slučajnih varijabli  $\mathbf{x}$  koja predstavlja rješenje funkcije cilja  $f(\mathbf{x})$ , a parametri raspodjele su označeni s  $\theta$ . U najjednostavnijem pristupu,  $p_{\theta}(\mathbf{x})$  je  $n$ -dimenzijska Gaussova raspodjela *neovisnih* varijabli, u kojoj  $\theta$  predstavlja samo vektor aritmetičkih sredina  $\mathbf{m}$  i vektor standardnih devijacija  $\sigma$  za svaku varijablu.

Ako pretpostavljamo  $n$  neovisnih slučajnih varijabli koje se ravnaju po Gaussovoj raspodjeli sa srednjom vrijednošću nula i devijacijom iznosa jedan, njihova raspodjela opisana je izrazom  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ , gdje je  $\mathbf{I}$  jedinična matrica. Promjena parametara raspodjele pojedine varijable tada se jednostavno postiže množenjem jedinične raspodjele vektorom standardnih devijacija te dodavanjem vektora srednjih vrijednosti pojedinih varijabli:

$$\theta = (\mathbf{m}, \sigma), p_{\theta}(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{I}) = \mathbf{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{I})$$

U ovoj inačici algoritma, u svakoj iteraciji novi skup jedinki veličine  $\lambda$  uzorkuje se iz trenutnih parametara raspodjele te se računa vrijednost funkcije cilja za svaku točku. Kako bismo odredili nove parametre raspodjele, promatra se samo  $\mu$  *najboljih jedinki* na temelju kojih se procjenjuju novi parametri. (Za razliku od prethodne inačice, u ovom algoritmu ukupan broj točaka je  $\lambda$ , od kojih se uzima podskup najboljih  $\mu$  iz trenutnog skupa rješenja.) Nova srednja vrijednost procjenjuje se za svaku varijablu na temelju aritmetičke sredine  $\mu$  najboljih jedinki:

$$\mathbf{m}^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{x}_i^{(g+1)}, \quad (1)$$

gdje se oznakom  $(g)$  označavaju trenutne vrijednosti parametara raspodjele, a oznakom  $(g+1)$  one koje će biti korištene u idućoj iteraciji. S druge strane, varijanca (kvadratna vrijednost devijacije  $\sigma$ ) procjenjuje se koristeći udaljenost  $\mu$  najboljih jedinki od srednje vrijednosti jedinki u *trenutnom* skupu ( $\mathbf{m}^{(g)}$ ):

$$\sigma^{(g+1)^2} = \frac{1}{\mu} \sum_{i=1}^{\mu} (\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)})^2 \quad (2)$$

Na taj način simulira se "pomak" devijacije i povećavanje područja uzorkovanja, što može biti korisno ukoliko je minimum udaljen od trenutnog položaja. Opisani postupak se ponavlja dok nije zadovoljen neki uvjet zaustavljanja, a kao konačno rješenje uzima se najbolja pronađena točka  $\mathbf{x}$  tijekom cijelog postupka.

**Primjer.** U slučaju dvodimenzijske funkcije cilja s varijablama  $x$  i  $y$ , parametri raspodjele su samo srednje vrijednosti  $m_x$  i  $m_y$  te njihove devijacije  $\sigma_x$  i  $\sigma_y$ . Cilj optimizacije je naći optimalne vrijednosti tih parametara, što se svodi na srednju vrijednost što bliže optimumu zadane funkcije cilja.

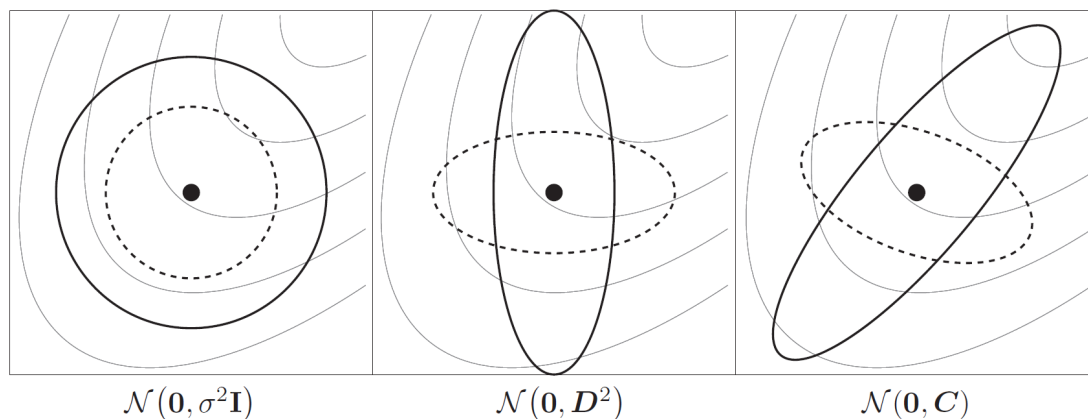
## 3 Jednostavna inačica algoritma CMA-ES

### 3.1 Motivacija

U prethodnoj inačici algoritma očit su neki nedostaci: većina optimizacijskih problema nije *separabilna*, već postoje ovisnosti između različitih varijabli. Također, neovisne raspodjele varijabli mogu modelirati samo one pomake u prostoru rješenja koji su paralelni sa smjerovima koordinatnih osi, što može biti neučinkovito za zadanu funkciju cilja.

Zbog navedenoga, za modeliranje raspodjele varijabli bolje je koristiti multivarijantnu normalnu raspodjelu koja dopušta *ovisnosti* među pojedinim slučajnim varijablama. Takva raspodjela označava se kao  $\mathcal{N}(\mathbf{m}, \mathbf{C})$ , gdje je  $\mathbf{C}$  *kovarijacijska matrica*. U slučaju neovisnih slučajnih varijabli, kovarijacijska matrica svodi se na dijagonalnu matricu, gdje su elementi na dijagonali jednaki varijanci (kvadratu standardne devijacije) pojedine varijable. Ako varijable nisu neovisne, i ostali elementi kovarijacijske matrice će biti proizvoljne vrijednosti (različiti od nule). Od uporabe kovarijacijske matrice dolazi i naziv algoritma *covariance matrix adaptation-ES*.

Utjecaj kovarijacijske matrice normalne raspodjele moguće je prikazati i grafički: ako bismo sve točke uzorkovali na jednakoj udaljenosti od srednje vrijednosti, sve točke nalazile bi se na površini  $n$ -dimenzijskog elipsoida definiranog s  $\mathbf{x}^\top \mathbf{C}^{-1} \mathbf{x} = 1$ . Primjeri utjecaja različitih kovarijacijskih matrica mogu se vidjeti na slici 3.1.



Slika 1: Oblik raspodjele za različite primjere kovarijacijske matrice (iz [1])

Grafički prikaz kovarijacijske matrice prikazuje nam sljedeće podatke: smjerovi osi dobivenih elipsoida zapravo predstavljaju *svojstvene vektore* matrice, dok kvadrati duljina osi predstavljaju *svojstvene vrijednosti*. Može se dokazati da svaka kovarijacijska matrica normalne raspodjele ima sljedeća svojstva:

- matrica je uvijek simetrična i *pozitivno definitna*;
- sve svojstvene vrijednosti su realne i pozitivne;
- svi svojstveni vektori matrice su *ortogonalni* (tj. linearno nezavisni).

Drugim riječima, kovarijacijska matrica se može predstaviti uz pomoć matrice svojstvenih vektora  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$ , te uz pripadne svojstvene vrijednosti  $\lambda_1^2, \dots, \lambda_n^2$  i dijagonalne matrice  $\mathbf{D} = \text{diag}(\lambda_1, \dots, \lambda_n)$  kao:

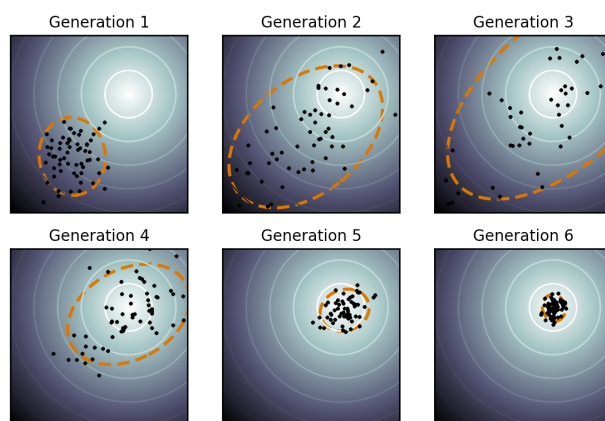
$$\mathbf{C} = \mathbf{B}^\top \mathbf{D}^2 \mathbf{B} = \begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1^2 & 0 & \dots & 0 \\ 0 & \lambda_2^2 & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & \dots & 0 & \lambda_n^2 \end{bmatrix} \begin{bmatrix} - & \mathbf{b}_1 & - \\ - & \mathbf{b}_2 & - \\ \dots & \dots & \dots \\ - & \mathbf{b}_n & - \end{bmatrix}$$

Prema tome, ovakav model raspodjele može se puno bolje prilagoditi zadanoj funkciji cilja u odnosu na skup neovisnih slučajnih varijabli. Ako pretpostavimo kvadratnu funkciju cilja, za nju vrijedi da Hesseova matrica drugih derivacija  $\mathbf{H}$  ima konstantne vrijednosti. Ukoliko za kovarijacijsku matricu postavimo da vrijedi  $\mathbf{C} = \mathbf{H}^{-1}$ , elipsoid koji opisuje raspodjelu poklapat će se s konturama (krivuljama jednake vrijednosti) zadane funkcije cilja. Na ovaj način zapravo kovarijacijskom matricom pokušavamo opisati

ponašanje Hesseove matrice funkcije cilja, uz odgovarajući položaj srednje vrijednosti (slika 3.1). U tom će slučaju uzorkovanje trenutne raspodjele odgovarati obliku funkcije cilja i omogućiti pronalaženje boljih rješenja. Primjerice, očekivano ponašanje algoritma može se ilustrirati prilagodbom na slici 2.

Uz ove pretpostavke, optimizacijski problem prelazi u sljedeći oblik:

$$\theta = (\mathbf{m}, \mathbf{C}), \quad p_{\theta}(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}, \mathbf{C}) \sim \mathbf{m} + \mathcal{N}(\mathbf{0}, \mathbf{C}).$$



Slika 2: Primjer rada algoritma na jednostavnom problemu (Wikipedia)

### 3.2 Izvedba algoritma

Tijek algoritma jednak je onome u prethodnoj inačici: uzorkovanje točaka u  $n$ -dimenzijskom prostoru uz pomoć trenutnih parametara raspodjele, ažuriranje parametara na temelju podskupa najboljih jedinki te ponavljanje prethodna dva koraka.

Za uzorkovanje iz zadane raspodjele  $\mathcal{N}(\mathbf{m}, \mathbf{C})$  potrebno je prvo odrediti svojstvene vektore i svojstvene vrijednosti kovarijacijske matrice, odnosno odrediti matrice  $\mathbf{B}$  i  $\mathbf{D}$ . Potom se zadana raspodjela može dobiti korištenjem sljedeće ekvivalencije:

$$\mathcal{N}(\mathbf{m}, \mathbf{C}) \sim \mathbf{m} + \mathbf{B}\mathbf{D}\mathcal{N}(\mathbf{0}, \mathbf{I}),$$

gdje je  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  normalna jedinična raspodjela neovisnih varijabli.

Nakon uzorkovanja, ocjenjuje se funkcija cilja u svim točkama i određuje podskup od  $\mu$  najboljih točaka. Ažuriranje srednje vrijednosti svodi se na računanje prosjeka položaja podskupa najboljih točaka (1), kao i u prethodnoj inačici algoritma.

Ažuriranje kovarijacijske matrice može se ilustrirati na primjeru dvodimenzijске raspodjele; ako su točke u dvodimenzijском prostoru označene kao  $(x_i, y_i)$ , a srednje vrijednosti raspodjele kao  $m_x$  i  $m_y$ , elementi kovarijacijske matrice  $2 \times 2$  mogu se odrediti kao

$$c_{1,1}^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} (x_i^{(g+1)} - m_x^{(g)})^2$$

$$c_{2,2}^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} (y_i^{(g+1)} - m_y^{(g)})^2$$

$$c_{1,2}^{(g+1)} = c_{2,1}^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} (x_i^{(g+1)} - m_x^{(g)})(y_i^{(g+1)} - m_y^{(g)})$$

U općenitom slučaju, nova kovarijacijska matrica proizvoljnih dimenzija računa se kao

$$\mathbf{C}^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} (\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)})(\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)})^\top \quad (3)$$

Važno svojstvo gornjeg načina ažuriranja kovarijacijske matrice je da procjenjuje *po-mak* postupka pretrage u odnosu na srednju vrijednost iz prethodnog koraka; ova ideja zajednička je svim oblicima algoritma CMA-ES. Nakon ažuriranja svih parametara, prelazi se na sljedeći korak koji započinje uzorkovanjem novog skupa točaka.

## 4 Potpuna inačica algoritma CMA-ES

Iako dobro opisuje osnovnu ideju algoritma, prethodna inačica ima nekoliko nedostataka koje je čine manje učinkovitom u primjeni. Prvi problem je ažuriranje kovarijacijske matrice uz pomoć izraza (3); opisani način je pouzdan jedino uz veliki broj točaka u svakoj iteraciji algoritma, što bi uzrokovalo velik broj potrebnih evaluacija funkcije cilja. Drugi problem je ignoriranje informacije iz prethodnih koraka, što u slučaju udaljenog optimuma rezultira velikim brojem potrebnih iteracija. Treći problem očituje se u tome da kovarijacijska matrica određuje *oblik* raspodjele, dok ima relativno mali utjecaj na *veličinu* raspodjele, koja može biti važna uz udaljeni optimum ili pred kraj postupka, kada se približimo lokalnom optimumu. Zbog toga se uvodi i treći parametar raspodjele, *globalni korak*  $\sigma$ , koji skalira kovarijacijsku matricu:

$$\theta = (\mathbf{m}, \sigma, \mathbf{C}), \quad p_\theta(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{C}) \sim \mathbf{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{C}),$$

Zbog navedenoga, algoritam se nadograđuje robusnijim mehanizmima ažuriranja parametara koji su opisani u nastavku.

### 4.1 Uzorkovanje novih točaka

Proces uzorkovanja novih točaka za korak  $(g+1)$  temeljem trenutnih parametara raspodjele  $(g)$  ne razlikuje se od osnovne inačice i svodi se na

$$\mathbf{x}_k^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathbf{BDN}(\mathbf{0}, \mathbf{I}), k = 1, \dots, \lambda$$

### 4.2 Ažuriranje srednje vrijednosti

Nakon uzorkovanja, određuje se vrijednost funkcije cilja u svakoj točki i promatra se podskup od  $\mu$  najboljih točaka. Ažuriranje srednje vrijednosti obavlja se korištenjem samo tih točki na sljedeći način:

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + \sum_{i=1}^{\mu} w_i (\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)}), \quad (4)$$

gdje su  $w_i$  težinski koeficijenti pojedinih točaka. Za težinske koeficijente vrijedi

$$\sum_{\mu} w_i = 1, \quad w_1 \geq w_2 \geq \dots \geq w_{\mu} > 0$$

Najčešći obabir vrijednosti koeficijenata je takav da svi imaju jednaku vrijednost  $1/\mu$ , no moguće je odabrati i linearno ili logaritamski padajuć niz vrijednosti. Na temelju odabranih vrijednosti koeficijenata, definira se efektivna vrijednost odabira:

$$\mu_{eff} = \frac{1}{\sum w_i^2}$$

Ukoliko se za koeficijente odabere jednaka vrijednost, tada je  $\mu_{eff} = \mu$  i izraz (6) svodi se na

$$\mathbf{m}^{(g+1)} = \frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{x}_i^{(g+1)},$$

### 4.3 Ažuriranje kovarijacijske matrice

Uloga kovarijacijske matrice je opisivanje *oblika* raspodjele (elipsoid u  $n$ -dimenzijskom prostoru) koji najbolje odgovara zadanoj funkciji cilja. Budući da se iz malog broja točaka ne može dobiti pouzdana procjena oblika, logično je iskoristiti vrijednosti iz prethodnih koraka i provoditi postupnu promjenu. Ako se za početnu vrijednost matrice definira  $\mathbf{C}^{(0)} = \mathbf{I}$ , tada se promjena može definirati kao

$$\mathbf{C}^{(g+1)} = (1 - c_{\mu})\mathbf{C}^{(g)} + c_{\mu}\mathbf{C}_{\mu}^{(g+1)} = (1 - c_{\mu})\mathbf{C}^{(g)} + c_{\mu}\frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{y}_i^{(g+1)} \mathbf{y}_i^{(g+1)\top}, \quad (5)$$

gdje je  $\mathbf{y}_i^{(g+1)} = (\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)})$ . Parametar  $c_{\mu} \leq 1$  je stopa učenja ranga  $\mu$ , a preporučena vrijednost se kreće oko  $\mu_{eff}/n^2$ . Budući da je zbroj vektorskih produkata u gornjem izrazu ranga najmanje  $\mu$  ili  $n$ , ova komponenta ažuriranja naziva se i **rang- $\mu$** .

(Postoji i inačica ažuriranja koja u obzir uzima sve jedinke iz skupa ( $\lambda$ ), ali tako da određeni dio najlošijih jedinki ima *negativne* težinske koeficijente, uz zbroj svih koeficijenata  $\sum_{\lambda} w_i = 0$ . Takva inačica naziva se i *aktivna selekcija*, eng. *active* ili A-CMA-ES.)

U gornjem izrazu, vektorski produkt s desne strane ignorira *predznak* vektora  $\mathbf{y}_i$ , budući da je  $\mathbf{y}\mathbf{y}^{\top} = -\mathbf{y}(-\mathbf{y})^{\top}$ , pa na taj način zapravo gubimo informaciju o *smjeru* pomaka. Iz tog razloga promatra se *evolucijski put* koji opisuje pomake srednjih vrijednosti raspodjela iz uzastopnih iteracija postupka, tj.

$$\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} + \frac{\mathbf{m}^{(g)} - \mathbf{m}^{(g-1)}}{\sigma^{(g-1)}} + \dots \quad (6)$$

Na temelju ovih pomaka, definira se parametar *evolucijski put kovarijacijske matrice* kojega označavamo s  $\mathbf{p}_c \in \mathbb{R}^n$  i započinje s nul-vektorom, a računa se kao

$$\begin{aligned}\mathbf{p}_c^{(g+1)} &= (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{1 - (1 - c_c)^2} \sqrt{\mu} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \\ &= (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\end{aligned}\quad (7)$$

gdje je  $c_c \leq 1$ . u gornjem izrazu, član  $\sqrt{c_c(2 - c_c)\mu}$  odabran je kako bi se iznos pomaka *normalizirao* tako da vrijedi  $\mathbf{p}_c^{g+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ , jer vrijedi  $(1 - c_c)^2 + \sqrt{c_c(2 - c_c)}^2 = 1$ . Na temelju ove vrijednosti definira se ažuriranje kovarijacijske matrice **ranga jedan** kao

$$\mathbf{C}^{(g+1)} = (1 - c_1)\mathbf{C}^{(g)} + c_1\mathbf{p}_c^{(g+1)}\mathbf{p}_c^{(g+1)\top} \quad (8)$$

gdje se za parametar  $c_1$  preporuča  $c_1 \approx 2/n^2$ . Ovaj način ažuriranja dolazi do većeg izražaja uz manji broj točaka i manji  $\mu_{eff}$ .

Konačno, oba načina ažuriranja (rang- $\mu$  (5) i rang-1 (8)) se kombiniraju u jedan zajednički izraz kao

$$\mathbf{C}^{(g+1)} = \underbrace{(1 - c_1 - c_\mu)}_{\text{može biti približno 1}} \mathbf{C}^{(g)} + c_\mu \frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{y}_i^{(g+1)} \mathbf{y}_i^{(g+1)\top} + c_1 \mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)\top}. \quad (9)$$

#### 4.4 Ažuriranje veličine koraka

Prethodno opisani izrazi utječu na oblik raspodjele, odnosno ekvivalentnog elipsoida u  $n$ -dimenzijskom prostoru. Za potrebe ažuriranja globalne veličine koraka  $\sigma$ , koji utječe na veličinu raspodjele, definira se pomoćna vrijednost *evolucijski put veličine koraka*  $\mathbf{p}_\sigma$ . Evolucijski put prati pomake srednjih vrijednosti (kao u izrazu (6)), a ideja promjene globalnog koraka ilustrirana je na slici 3.

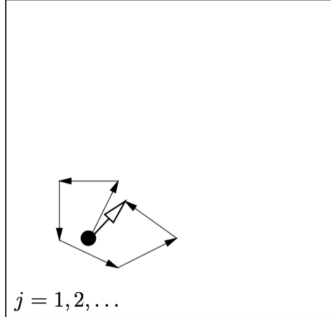
Ukoliko se uzastopni pomaci međusobno poništavaju (slika lijevo), evolucijski put je kratak i globalni korak bi se trebao  *smanjiti*. S druge strane, ukoliko su uzastopni pomaci korelirani i pokazuju u istom smjeru (slika desno), globalni korak treba *povećati*. U "idealnom" slučaju, uzastopni pomaci su približno okomiti i nekorelirani (sredina). Na temelju ovoga, akumilirana duljina evolucijskog puta računa se kao

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}} \mathbf{C}^{(g)-\frac{1}{2}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \quad (10)$$

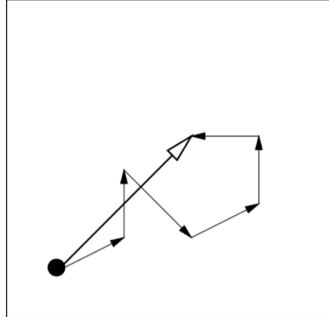
U računanju ovog parametra javlja se vrijednost  $\mathbf{C}^{-\frac{1}{2}} = \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^\top$ ; ovaj član je uvršten kako bi se iznos pomaka učinio neovisnim o *smjeru*, budući da razlika  $\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}$  ovisi o trenutnoj orijentaciji kovarijacijske matrice. Dodatno, član  $\sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}}$  uvršten je zbog normalizacije, kao i kod računanja evolucijskog puta kovarijacijske matrice. Ovako izračunati evolucijski put uspoređuje se s "očekivanom" duljinom puta, koja uz pretpostavku nekoreliranih pomaka iznosi  $\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| \approx \sqrt{n}(1 - \frac{1}{4n} + \frac{1}{21n^2})$ . Konačno, ažuriranje globalne veličine koraka obavlja se kao



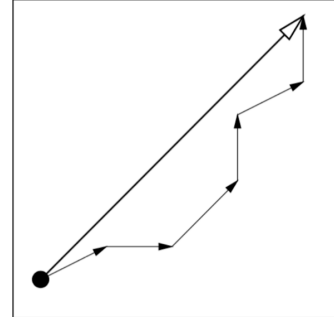
Single steps cancel each other off and thus evolution path is short.  
→ **Decrease  $\sigma$**



Ideal case: single steps are uncorrelated.



Single steps point to the same direction and thus evolution path is long.  
→ **Increase  $\sigma$**



Slika 3: Procjena evolucijskog puta na temelju uzastopnih pomaka (iz [2])

$$\sigma^{(g+1)} = \sigma^{(g)} \exp \left( c_{\sigma} \left( \frac{\|\mathbf{p}_{\sigma}^{(g+1)}\|}{\mathbb{E}\|\mathcal{N}(0, \mathbf{I})\|} - 1 \right) \right). \quad (11)$$

#### 4.5 Preporučene vrijednosti parametara

Budući da algoritam posjeduje nekoliko parametara, autor preporuča sljedeće vrijednosti na temelju velikog broja ispitivanja u primjeni:

$$\begin{aligned} c_c &= \frac{4 + \mu_{eff}/n}{n + 4 + 2\mu_{eff}/n} \approx \frac{4}{n} \\ c_{\mu} &= \min \left( 1 - c_1, 2 \frac{\mu_{eff} - 2 + 1/\mu_{eff}}{(n + 2)^2 + \mu_{eff}} \right) \approx \frac{\mu_{eff}}{n^2} \\ c_1 &= \frac{2}{(n + 1.3)^2 + \mu_{eff}} \approx \frac{2}{n^2} \\ c_{\sigma} &= \frac{\mu_{eff} + 2}{n + \mu_{eff} + 5} \approx \frac{4}{n} \end{aligned}$$

#### 4.6 Inačice algoritma CMA-ES

Osim opisanog načina izvedbe, ovaj algoritam doživio je mnogo preinaka i poboljšanja; neke od popularnijih inačica su sljedeće:

- MO-CMA-ES (eng. *multiobjective optimization* CMA-ES) – sastoji se od više CMA-ES algoritama (u pravilu su to (1+1) varijante), tj. svaka jedinka MO-CMA-ES je jedan CMA-ES;

- SS-MO-CMA-ES (eng. *the steady state variant* MO-CMA-ES) – koristi *steady state* selekciju prilikom stvaranja nove populacije;
- LR-CMA-ES (eng. *local restart* CMA-ES) – omogućava se ponovo pokretanje ako se uoči da ne uspijeva dovoljno brzo konvergirati;
- IPOP-CMA-ES (eng. *increasing population size* CMA-ES) – implementacija s mogućnošću ponovnog pokretanja s povećanom populacijom;
- BIPOP-CMA-ES (eng. *bipopulation* CMA-ES) – sadrži dvije populacije: jednu veliku za globalno pretraživanje i jednu manju za lokalno pretraživanje;
- Sep-CMA-ES (eng. *separable* CMA-ES) – varijanta s dijagonalnom kovarijantnom matricom (neovisne slučajne varijable) te na taj način i sa smanjenom vremenskom kompleksnosti,
- L-CMA-ES (eng. *local* CMA-ES) – veličina koraka i veličina populacije su veoma mali, time se oponaša više lokalnih pretraga;
- A-CMA-ES (eng. *active* CMA-ES) – modifikacija CMA-ES na način da ne koristi samo podatke o najboljim jedinkama, već i podatke o najlošijim jedinkama;
- PS-CMA-ES (eng. *particle swarm* CMA-ES) – kombinacija PSO i CMA-ES optimizacijskih algoritama.

## Literatura

- [1] Nikolaus Hansen. "The CMA Evolution Strategy: A Tutorial", arXiv preprint <https://arxiv.org/abs/1604.00772> (2016).
- [2] Lilian Weng. "Evolution Strategies", <https://lilianweng.github.io/lil-log/2019/09/05/evolution-strategies.html>
- [3] Mirela Ćosić. "Stohastička numerička optimizacija u okruženju za evolucijsko računanje", <https://www.bib.irb.hr/586610>
- [4] David Ha. "A Visual Guide to Evolution Strategies", <https://blog.otoro.net/2017/10/29/visual-evolution-strategies/>
- [5] Anne Auger, Nikolaus Hansen. "Tutorial: CMA-ES — Evolution Strategies and Covariance Matrix Adaptation", <http://www.cmap.polytechnique.fr/~nikolaus.hansen/gecco2011-CMA-ES-tutorial.pdf>