SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2578

Stohastička numerička optimizacija u okruženju za evolucijsko računanje

Mirela Ćosić

Zagreb, lipanj 2012.

Sadržaj

.

Uvod

Evolucijske strategije su optimizacijske, prirodom inspirirane, strategije koje naglasak stavljaju na evoluciji i prilagodbi generacija. Evolucijske strategije se uvelike oslanjaju na Darwinovu teoriju evolucije: preživljavaju samo najbolji. Temelje se prvenstveno na selekciji i rekombinaciji, sa ciljem da se unaprjeđuju najbolje jedinke iz generacije te se na taj način pokušava stvoriti „bolja" generacija.

Postoje mnoge različite implementacije evolucijskih strategija u kojima svaka od njih ima svojih posebnosti.[1] Cilj ovog rada je bio obraditi i implementirati evolucijsku strategiju sa adaptacijom kovarijacijske matrice (eng. CMA-ES, Covariance Matrix Adaptation Evolution Strategy).

CMA-ES je baziran na multivarijatnoj normalnoj distribuciji koji uspješno pronalazi optimalno rješenje problema na način da postupno prilagođava očekivanje, standardnu devijaciju i kovarijancu, odnosno kovarijantnu matricu distribucije[4].

Temelj ovog algoritma je prvenstveno statistika, ali na način da se postepeno, znači u svakom koraku, parametri razdiobe prilagođavaju prema najboljim jedinkama izbjegavajući prividne optimume te se na taj način približavaju stvarnom optimumu funkcije.

1. Evolucijska strategija uz kovarijantnu matricu

CMA-ES se danas smatra jednim od najboljih algoritama za optimizaciju neprekinutih funkcija, čak i za veće dimenzije prostora pretraživanja. Uspješnost je uglavnom bazirana na nepromjenjivosti s obzirom na monotone i ortogonalne transformacije, odnosno da će tako transformirane funkcije uspješno minimizirati zbog postupne adaptacije kovarijantne matrice[9]. CMA-ES je predstavio Nikolaus Hansen na kongresu evolucijskog računanja 2001. godine. Poslije toga, CMA-ES je bio testiran i uspoređivan na mnogim konferencijama te zbog svoje uspješnosti na raznolikim funkcijama došao na dobar glas.[10]

Glavne karakteristike CMA-ES su:

- postepeno povećavanje uspješnosti generacije na način da se povećava izglednost odabira uspješne jedinke,
- pamćenje „evolucijskog puta" strategije.

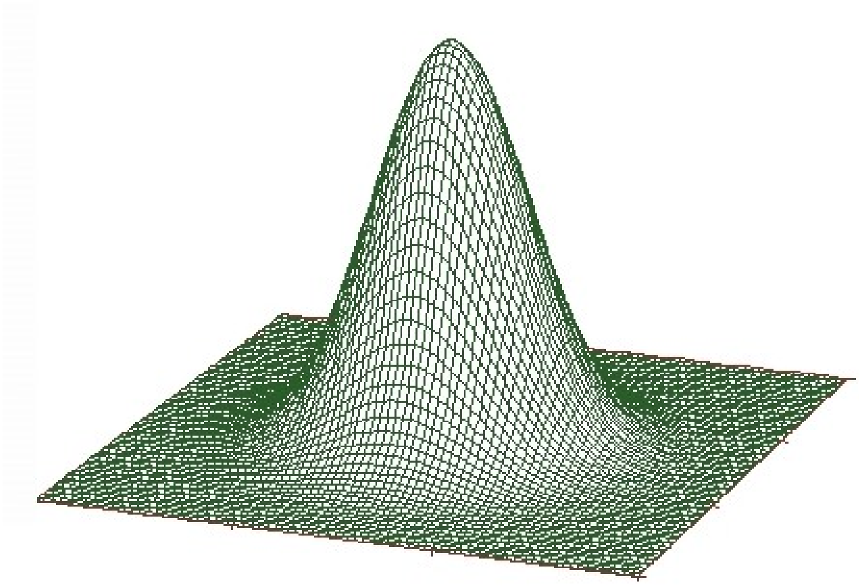
Postepeno povećavanje izglednosti da će se odabrati uspješna jedinka znači da se korak po korak populacija sve više približavati optimalnoj vrijednosti, dok će kovarijanca i disperzija biti sve manje. Takvim načinom će pretraga biti fokusirana na manje, lokalizirano područje. Pritom će pamćenje evolucijskog puta omogućavati da pretraga optimalnog rješenja ide u pravome smjeru. [2]

1.1. Matematička podloga i normalna distribucija

Kao i većina evolucijskih strategija, CMA-ES se bazira na normalnoj ili gaussovoj distribuciji (Slika 1.1). Ako se radi o višedimenzionalnom problemu, tada se poopćuje na multivarijantnu normalnu distribuciju. Ideja CMA-ES je da jedinke za novu populaciju bira po takvoj multivarijantnoj normalnoj distribuciji, gdje bi X bio vektor sa vrijednostima jedinke . Tada se vektor X može zapisati kao:

$$X \sim \mathcal{N}(\mu, C) \tag{1}$$

gdje je μ vektor očekivane vrijednosti razdiobe, a C ne-negativna, simetrična kovarijantna matrica razdiobe.



Slika 1.1 Normalna razdioba u 3D prostoru

Radi lakšeg računanja, matrica C se rastavlja na umnožak i to po formuli:

$$C = B\ D\ B^{-1} \tag{2}$$

gdje je B matrica svojstvenih vektora (jedan stupac matrice je jedan vektor), a D dijagonalna matrica sa svojstvenim vrijednostima (jedan element dijagonale je jedna svojstvena vrijednost) te je $^{-1}$ oznaka za inverznu matricu.

Uz takvu rastavu matrice i korištenjem nekoliko matematičkih teorema[5], dobivamo izraz pogodan za uzorkovanje vrijednosti po normalnoj razdiobi:

$$X = \mu + B\ D^{1/2}\ Y \tag{3}$$

gdje je Y vektor nasumičnih vrijednosti distribuiranih po normalnoj razdiobi, odnosno svaki element je odabran po normalnoj razdiobi: $y_i \sim \mathcal{N}(0, 1)$.

Vrijednosti normalne razdiobe se mogu računati na mnogo različitih načina, no u prilikom ove implementacije je odabrana *Box-Muller*-ova transformacija[6] koja iz uniformne jedinične razdiobe računa vrijednosti jedinične normalne razdiobe.

1.2. Podjela i vrste CMA-ES algoritama

Vrlo je teško naći neku konkretnu podjelu i podvrste ovog algoritma, a razlog tome je prvenstveno aktualnost CMA-ES, te se u zadnjih nekoliko godina smišlja dosta novih, naprednijih inačica.

Općenito se ES dijele na dvije glavne skupine:

- plus strategije su strategije kod kojih se selekcija za novu populaciju vrši i nad skupinom roditelja (μ) i nad skupinom njihove djece (λ). Zbog toga što se selekcija vrši i nad roditeljima, ova strategija ima svojstvo elitizma. To znači da se neće izgubiti najbolja roditeljska jedinka, već će preći u novu populaciju. Problem elitizma je to što ima tendenciju zapeti u lokalnim optimumima. Najčešći tipovi plus strategije su:
    - $(1+1)$-ES – od jednog roditelja se stvara jedno dijete te će to dijete preći u novu generaciju jedino ako je bolje od roditelja,
    - $(\mu+1)$-ES – od μ roditelja se stvara jedno dijete koje se ubacuje u skup roditelja, te se potom iz tog skupa izbacuje najlošija jedinka, a ostale postaju roditelji za sljedeću generaciju,
    - $(1+\lambda)$-ES – od jednog roditelja se stvara λ djece te se uzima samo najbolja jedinka za sljedeću generaciju,
    - $(\mu+\lambda)$-ES – od μ roditelja se stvara λ djece koji se ubacuju u skup roditelja te se bira μ najboljih jedinki za roditelje sljedeće generacije,
    - $(\mu/\rho+\lambda)$-ES – način selekcije je isti kao i kod $(\mu+\lambda)$-ES, s time da prilikom izbora roditelja se koristi parametar ρ koji može označavati, npr. broj roditelja koji se biraju prilikom stvaranja djece.
- zarez strategije su strategije kod kojih se selekcija za novu populaciju vrši samo nad skupinom djece (λ). Problem ovakve selekcije je što postoji šansa da se izgubi najbolja jedinka. To se može dogoditi ako ne bude izabrana za roditelja ili ako se prilikom mutacije ili rekombinacije izgube njena dobra svojstva. Dobro svojstvo zarez strategije je da teže zapinje u lokalnim optimumima. Najčešći tipovi zarez strategije su:
    - (μ,λ)-ES – od μ roditelja se stvara λ djece te se od djece bira μ jedinki za roditelje sljedeće generacije. Kod zarez strategija se rijetko razmatra da je μ ili λ jednak jedan jer onda cijela strategija gubi smisao,

- $(\mu/\rho, \lambda)$-ES – način selekcije je isti kao i kod (μ, λ)-ES, s time da prilikom izbora roditelja se koristi parametar ρ koji može označavati, npr. broj od koliko roditelja se stvara jedno dijete.
- $(\mu', \lambda'(\mu, \lambda)^{\gamma})$-ES – od μ' roditelja se stvara λ' djece koji se onda izoliraju na γ generacija. U svakoj toj γ-generaciji se od μ roditelja se stvara λ djece te se onda od njih bira μ najboljih za sljedeću generaciju. Nakon što prođe tih γ generacija, bira se λ' najboljih jedinki.[7]

Iako se prilikom implementacije CMA-ES može izabrati bilo koji tip selekcije, ipak je najčešći odabir $(\mu/\mu_w, \lambda)$-CMA-ES gdje μ_w označava da se prilikom izbora roditelja koristi težinska funkcija. Pritom treba napomenuti da $(\mu/\mu_w, \lambda)$-CMA-ES nije elitistički algoritam, no da se usprkos tome bira kao najčešća implementacija.

Nadalje, postoji mnogo varijanti CMA-ES u kojima se mijenjaju neki parametri, omogućava ponovo pokretanje, povećava populacija ili pak kombinira s nekim drugim idejama i strategijama. Većina od njih je u osnovi $(\mu/\mu_I, \lambda)$-CMA-ES, samo sa nekim dodatnim ili proširenim mogućnostima.

Neki od poznatijih varijanti CMA-ES su:

- MO-CMA-ES (eng. *multiobjective optimization* CMA-ES) – sastoji se od više CMA-ES algoritama (u pravilu su to (1+1) varijante), odnosno svaka jedinka MO-CMA-ES je jedan CMA-ES,
  - SS-MO-CMA-ES (eng. *the stady state variant of* MO-CMA-ES) – koristi *steady state* prilikom stvaranja nove populacije [9],
- 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 siz*e CMA-ES) – implementacija sa mogućnošću ponovog pokretanja sa povećanom populacijom,
- BIPOP-CMA-ES (eng. *bipopulation* CMA-ES) – sadrži dvije populacije: jednu veliku za globalno pretraživanje i jednu manju za lokalno pretraživanje,
- G-CMA-ES (eng. *global* CMA-ES) – kombinacija IPOP varijante, ali sa mogućnošću ponovnog pokretanja algoritma ako algoritam ne pronađe dovoljno optimalno rješenje, te se parametri prilagođavaju za globalnu pretragu
- Sep-CMA-ES (eng. separable CMA-ES)– varijanta sa dijagonalnom kovarijantnom matricom 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 lokalna pretraga,
- A-CMA-ES (eng. *active* CMA-ES) – modifikacija CMA-ES na način da ne koristi samo podatke o najboljim potomcima, već da i koristi podatke i o najlošijijm potomcima. Ideja je da osim da se razdioba ne pomiče samo prema najboljem rješenju, već i da odmiče od najlošijega. [8]
- PS-CMA-ES (eng. *particle swarm* CMA-ES) - kombinacija PSO i CMA-ES optimizacijskih algoritama

1.3. Parametri CMA-ES

Osim broja roditelja (μ) i broja potomaka (λ), koji su karakteristični parametri za ES, CMA-ES koristi puno svojih „internih" parametara koji su neovisni o optimizacijskom problemu. Većina njih nije dostupna korisniku na mijenjanje, te njihova vrijednost zapravo ovisi o implementaciji algoritma.

1.3.1. Dimenzija problema

Dimenzija problema (n ili N) je dimenzija koordinatnog sustava nad kojim se radi. Na slici (Slika 1.1) dimenzija je $n = 2$. Povećanje dimenzije donosi sa sobom povećanje kompleksnosti računa.

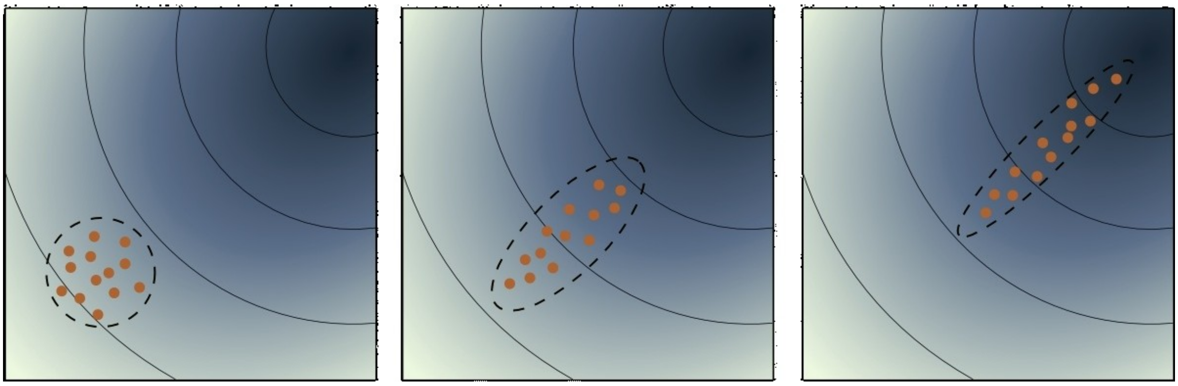
1.3.2. Očekivana vrijednost

Očekivana vrijednost (eng. *mean*, oznaka m) je parametar koji opisuje položaj gdje se u prostoru nalazi sredina oko koje se biraju nove jedinke prilikom stvaranja populacije. Ako se pogleda, npr. dvodimenzionalna normalna distribucija sa slike (Slika 1.1), njena očekivana vrijednost je vrh zvona. Treba primijetiti da je očekivana vrijednost vektor čija je veličina određena dimenzijom problema, $m \in \mathbb{R}^n$.

Prilikom izvođenja algoritma, srednja vrijednost se računa na način da se najbolje jedinke skaliraju sa težinskom funkcijom ili se pak računa njihova srednja vrijednost. Ako se algoritam uspješno izvede, očekivana vrijednost će biti jednaka ili jako približna optimalnoj vrijednosti.

1.3.3. Veličina koraka i kontrola veličine koraka

Veličina koraka (eng. *step-size*, oznaka σ) je skalarna vrijednost koja govori koliko u širinu se uzorkuje, odnosno kolika je disperzija za sve dimenzije. Veličina koraka određuje veličinu distribucijskog elipsoida (Slika 1.2).



Slika 1.2 Promjena položaja i oblika distribucijskog elipsoida kroz generacije

Veličina koraka je promjenjiva vrijednost, te se prilagođava s obzirom na uspješnost generacije. O tome koliko će se promijeniti veličina koraka ovisi samo o njegovom evolucijskom putu (eng. *evolution path*). Evolucijski put se promatra kao zbroj do sada prijeđenih koraka, gdje korak označuje razliku između očekivane vrijednosti trenutne i očekivane vrijednosti prijašnje generacije. Ako je duljina puta velika, onda i veličina koraka mora narasti da može obuhvatiti optimalno rješenje. No, ako je duljina evolucijskog puta mala tada treba smanjiti veličinu koraka da se unutar područja uzorkovanja može konvergirati k optimalnom rješenju.

1.3.4. Adaptacija kovarijantne matrice

Kovarijantna matrica (eng. *covariance matrix*, oznaka *C*) ponaša se kao parametar disperzije, ali za više dimenzija. Matematički gledano, vrijednost na indeksu (i,j) određuje kovarijaciju između i-tog i j-tog elementa slučajnog vektora.[3] Za CMA-ES, kovarijantna matrica određuje oblik distribucijskog elipsoida (Slika 1.2), odnosno za svaku dimenziju određuje kolika će biti disperzija.

Kovarijantna matrica se mijenja na način da pomiče prostor pretraživanja više prema jedinkama (vrijednostima) koje su u prošlom koraku bile uspješne. Na taj način se unutar kovarijantne matrice skuplja i pamti informacija o populacijama i njihovoj uspješnosti.

Promjenjivost kovarijantne matrice je vrlo bitna komponenta, te se može provoditi na nekoliko razina:

1. Ažuriranje matricom ranga jedan – korigiranje kovarijantne matrice ovisi samo o evolucijskom putu kovarijantne matrice sljedeći tako gradijent funkcije dobrote,
2. Ažuriranje matricom ranga μ – korigiranje kovarijantne matrice se bazira na ponderiranoj srednjoj vrijednosti boljeg dijela populacije i korisno je kod velikih populacija jer direktno uzima u obzir sve vrijednosti roditelja.
3. Ažuriranje negativnom matricom – kovarijantna matrica se korigira sa najlošijim dijelom populacije zbog toga da se zapamti u koju stranu ne treba pretraživati.

Odabir na koji način će se mijenjati kovarijantna matrica ovisi o implementaciji. Neki odaberu samo jedan način, neki pak sva tri. Pritom treba uzeti u obzir da se povećavanjem složenosti algoritma on može bitno usporiti.

1.3.5. Stope učenja

Stope učenja (eng. *learning rate*) su konstante koje određuju koliko brzo i koliko jako će se događati promjene. Intuitivno je jasno da premala stopa učenja će biti jako neefikasna, no jednako tako neefikasna će biti i prevelika stopa učenja jer će „pregaziti“ optimalno rješenje te će teško konvergirati.

CMA-ES koristi četiri stope učenja i to za:

- ažuriranje evolucijskog puta veličine koraka,
- ažuriranje evolucijskog puta kovarijantne matrice,
- ažuriranje kovarijantne matrice za matricu ranga jedan,
- ažuriranje kovarijantne matrice za matricu ranga μ.

1.4. Pseudokod

Tijek CMA-ES se može podijeliti na nekoliko koraka:

1. Unos i inicijalizacija parametara,
2. Glavna generacijska petlja (ponavlja se dovoljan broj puta):
    a. uzorkovanje vrijednosti (potomci)
    b. evaluacija i rekombinacija potomaka – izbor budućih roditelja
    c. promjena parametara s obzirom na najbolje jedinke iz skupa potomaka

3. Povrat najbolje jedinke (vrijednosti)

Svaki korak pseudokoda zapravo ovisi o inačici CMA-ES koja se implementira, no u nastavku poglavlja će biti objašnjeni koraci za $(\mu/\mu_w, \lambda)$-CMA-ES

Unos i inicijalizacija parametara

Iako većina implementacija CMA-ES ne uključuje unos parametara, *srednja vrijednosti* (m) i *veličina koraka* (σ) su jedini parametri koji bi se trebali unositi od strane korisnika, da se mogu prilagoditi funkciji koji minimiziraju. Ako se pak odluči ni te parametre ne ostavljati korisniku na mijenjanje, onda se u pravilu postavljaju na slučajne vrijednosti. Jedino ograničenje je da veličina koraka mora poprimiti pozitivnu vrijednost

Ostali parametri se inicijaliziraju na sljedeći način:

- *kovarijantna matrica* C te njene komponente (matrica svojstvenih vektora B i matrica svojstvenih vrijednosti D) se inicijaliziraju na matricu identiteta ***I*** (elementi na dijagonali imaju vrijednost jedan, a svi ostali nula)

$$C = I\,, \qquad B = I\,, \qquad D = I \tag{4}$$

- *evolucijski putovi* za kovarijantnu matricu i veličinu koraka se postave na nul-vektore veličine *n*:

$$p_c = \vec{0}\,,\ \ p_\sigma = \vec{0} \tag{5}$$

- za vektor *težine* (w) vrijede dva uvjeta:

$$w_1 \geq w_2 \geq \cdots \geq w_\mu \quad \text{i} \quad \textstyle\sum_{i=1}^{\mu} w_i = 1 \qquad (6)\ (7)$$

  Vektor težine se može inicijalizirati da bude neutralan na način da je svaki njegov element isti, $w_i = \frac{1}{\mu}$ te se onda sve odabrane jedinke smatraju jednako važnima.

- *efektivna vrijednost populacije* je μ_w i za nju vrijedi formula:

$$\mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \tag{8}$$

- za *stope učenja* se preporuča približna vrijednost, koja daje optimalne rezultate:
  - za kovarijantnu matricu: $c_C \approx \frac{4}{n}$
  - za veličinu koraka: $c_\sigma \approx \frac{4}{n}$
  - za rang-jedan matricu: $c_1 \approx \frac{2}{n^2}$
  - za rang-μ matricu: $c_\mu \approx \frac{\mu_w}{n^2}$

Prilikom implementacije stopa učenja su korištene formule preporučene od strane autora ovog algoritma. To su redom:

$$c_C = \frac{4 + \frac{\mu_w}{N}}{N + 4 + \frac{2\mu_w}{N}} \tag{9}$$

$$c_\sigma = \frac{\mu_w + 2}{N + \mu_w + 5} \tag{10}$$

$$c_1 = \frac{2}{(N + 1.3)^2 + \mu_w} \tag{11}$$

$$c_\mu = min\left(1 - c_1, 2 \cdot \frac{\mu_w - 2 + \frac{1}{\mu_w}}{(N + 2)^2 + \mu_w}\right) \tag{12}$$

Uvjeti prekida su obično najveći dopušteni broj generacija i najmanja dopuštena pogreška jedinke. Oni se mogu ili inicijalizirati ili dopustiti kao parametri koji se unose, ovisno o tome želimo li da budu neovisni o funkciji koja se minimizira ili ne. Naknadno je dodano i uvjet ponovnog pokretanja algoritma. Njegova je zadaća da detektira probijanje najveće dopuštene vrijednosti za parametar veličine koraka koja je ograničena najvećom dopuštenom vrijednosti za tip podataka *double*.

Glavna generacijska petlja

Glavna generacijska petlja se ponavlja dok god se ne dostigne neki uvjet prekida, te se može podijeliti na nekoliko cjelina:

1. Uzorkovanje vrijednosti – normalnom razdiobom se izvlače vrijednosti (potomci) iz prostora pretraživanja

$$x_i = m + \sigma\, B\, D^{1/2}\, z_i \ , \ \ i \in \{1, \dots, \lambda\} \tag{13}$$

   gdje je z_i vektor nasumičnih vrijednosti distribuiranih po jediničnoj normalnoj razdiobi, a λ veličina populacije, odnosno broj potomaka.

2. Evaluacija i rekombinacija – svaki potomak x_i se evaluira te se potom pomoću izračunate dobrote iz skupa potomaka bira njih μ najboljih. Najbolje jedinke se usrednjuju pomoću težinske funkcije te se njihova sredina postavlja kao nova srednja vrijednost:

$$m_{novi} = \sum_{i=1}^{\lambda} w_i x_{i:\lambda} \tag{14}$$

Oznaka i:λ označava da su jedinke poredane prema njihovoj dobroti od najbolje prema najlošijoj, odnosno da prilikom traženja minimuma vrijedi:

$$f(x_{1:\lambda}) \leq f(x_{2:\lambda}) \leq \cdots \leq f(x_{\lambda:\lambda}) \tag{15}$$

3. Promjena preostalih parametara – u prošlom koraku se izračunala nova srednja vrijednost, no sada treba prilagoditi i ostale parametre prema novoj generaciji.
    a. *Evolucijski put veličine koraka* se mijenja na način da se dosadašnja vrijednost skalira koristeći pritom konstantu učenja evolucijskog puta veličine koraka (c_σ) te joj se doda (opet skalirana) vrijednost promjene srednje vrijednosti:

    $$p_\sigma = (1 - c_\sigma)\, p_\sigma \; + \; \sqrt{1 - (1 - c_\sigma)^2}\, \sqrt{\mu_w}\, C^{-\frac{1}{2}}\, \frac{m_{novi} - m_{stari}}{\sigma_{stari}} \tag{16}$$

    gdje se $C^{-\frac{1}{2}}$ može rastaviti kao $B\, D^{-\frac{1}{2}}\, B^{-1}$ pošto su već te matrice izračunate, a $D^{-\frac{1}{2}}$ se računa na način da sve korjenuje svaki element dijagonale (*D* je dijagonalna matrica).
    b. *Veličina koraka* će se mijenjati samo ovisno o tome kakav je njen evolucijski put. Ako je duljina evolucijskog puta velika, povećat će se i veličina koraka, a ako je duljina mala, smanjiti će se i veličina koraka. Referentna točka prilikom kategorizacije evolucijskog puta na velike i male je duljina, odnosno norma normalne razdiobe. Promjena veličine koraka se računa prema formuli:

    $$\sigma_{novi} = \sigma_{stari} \cdot exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{E\|\mathcal{N}(0,I)\|} - 1\right)\right) \tag{17}$$

    gdje je d_σ faktor prigušivanja i vrijedi $d_\sigma \approx 1$, a očekivanje normalne razdiobe se može (dovoljno dobro) računati prema formuli:

    $$E\|\mathcal{N}(0,I)\| \approx \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right) \tag{18}$$

    c. Ideja *evolucijskog puta kovarijantne matrice* jest da pamti promjene, tj. smjer putovanja kovarijantne matrice. Njena promjena je slična promjeni evolucijskog puta veličine koraka: prijašnja vrijednost se smanji za neki faktor (stopu učenja kovarijantne matrice, c_C) te se doda promjena između trenutne i prijašnje generacije.

$$p_C = (1 - c_C)p_C + \sqrt{1 - (1 - c_C)^2}\,\sqrt{\mu_w}\,\frac{m_{novi} - m_{stari}}{\sigma_{stari}} \tag{19}$$

d. Promjena *kovarijantne matrice* se može napraviti dodavanjem triju različitih matrica: matricom ranga jedan, matricom ranga μ te negativnom matricom. Kada se koristi samo matica ranga jedan, stara vrijednost kovarijantne matrice se malo smanji, da bi se za tu smanjenu vrijednost mogla dodati nova matrica ($p_C {p_C}^T$):

$$C = (1 - c_1)\, C + c_1 p_C {p_C}^T \tag{20}$$

Iako je u pravilu dovoljno samo mijenjati kovarijantnu matricu matricom ranga jedan, kod velikih populacija se olakšava konvergencija ako se koristi i matrica ranga μ.

$$C = \left(1 - c_1 - c_\mu\right) C + c_1 p_C {p_C}^T + \, c_\mu \sum_{i=1}^{\mu} w_i\, y_{i:\lambda}\, {y_{i:\lambda}}^T \tag{21}$$

Izraz $y_{i:\lambda}$ se može izračunati po formuli:

$$y_{i:\lambda} = \frac{x_{i:\lambda} - m_{stari}}{\sigma_{stari}} \tag{22}$$

Kod negativne varijante algoritma (A-CMA-ES) se koristi i dodatno mijenjanje kovarijantne matrice sa lošim jedinkama iz populacije. Tako se formula za izračunavanje kovarijantne matrice dodatno proširuje:

$$\begin{aligned} C = {} & \left(1 - c_1 - c_\mu + \alpha\, c_{neg}\right) C \\ & + c_1 p_C {p_C}^T \\ & + \left(c_\mu - (1 - \alpha)\, c_{neg}\right) \sum_{i=1}^{\mu} w_i\, y_{i:\lambda}\, {y_{i:\lambda}}^T \\ & - c_{neg} \sum_{i=\lambda-\mu+1}^{\lambda} w_i\, z_{i:\lambda}\, {z_{i:\lambda}}^T \end{aligned} \tag{23}$$

gdje α označava faktor kojim se nadoknađuje gubitak varijance, a c_{neg} jačinu utjecaja negativne matrice na kovarijantnu matricu. Vektor z_i je definiran u formuli (13).

Nakon izračuna kovarijantne matrice, računaju se njezine komponente: matrica svojstvenih vektora B i matrica svojstvenih vrijednosti D. Pošto je takvo rastavljanje dosta složeno, u ovom radu se koriste dodatne biblioteke koje to omogućuju.

Završetak

Generiranje novih populacija se zaustavlja nakon što se dosegne neki od uvjeta prekida. Potom CMA-ES vraća najbolju jedinku ili srednju vrijednost iz zadnje generacije kao pronađeni optimum funkcije. Ako se CMA-ES prekine zato jer je evaluacija bila dovoljno dobra, onda nije velika razlika koje od toga se vraća jer je jako mala razlika među njima. No, ako dođe do završetka zbog nekog drugog uvjeta, npr. dosegnut je najveći mogući broj generacija, tada može postojati značajna razlika, te je onda reprezentativnije da se uzima najbolja jedinka zadnje populacije.

1.5. Primjer inicijalizacije i prolaska kroz generaciju

Kao funkciju evaluacije neka se koristi:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} {x_i}^2 \tag{24}$$

i to na intervalu $[-5, 5]$.

Neka se kao ulazni parametri $(\mu/\mu_w, \lambda)$-CMA-ES postave vrijednosti: $N = 7$, $\mu = 3$, $\lambda = 6$, te neka se težinska funkcija računa po formuli (24). Očekivanu vrijednost i veličinu koraka postavljamo da vrijednosti budu unutar granica.

- $m = [0.4688 \quad 4.5751 \quad 4.6489 \quad -3.4239 \quad 4.7059 \quad 4.5717 \quad -0.1465]^T$
- $\sigma = 5$

Pritom se i ostali parametri inicijaliziraju prema formulama (4), (5), (8), (9), (10), (11), (12) i u ovom primjeru iznose:

- $C = I,\ B = I,\ B = I$
- $p_c = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$, $p_\sigma = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$
- $w = [0.637 \quad 0.2846 \quad 0.0784]^T$
- $\mu_w = 2.0286$

- $c_C = 0.3705,\ c_\sigma = 0.2872, c_1 = 0.0282, c_\mu = 0.0126$

Prilikom prolaska kroz glavnu generacijsku petlju se prvo generiraju nove jedinke po pravilu normalne distribucije. To znači da se šest puta prolazi kroz formulu (13) i neka onda imamo sljedeće vrijednosti i vrijednosti njihovih dobrota:

1. $x_1 = [-5.5686 \quad 8.1613 \quad 12.8001 \quad -0.9794 \quad 9.8794 \quad 8.2061 \quad -1.6634]^T$
   $f(x_1) = 430.126$
2. $x_2 = [1.9382 \quad 0.6387 \quad 9.0909 \quad -9.1592 \quad -0.6384 \quad 0.5242 \quad -14.8677]^T$
   $f(x_2) = 392.4293$
3. $x_3 = [7.6607 \quad 6.201 \quad 0.8742 \quad 3.4276 \quad -3.8517 \quad 4.0605 \quad -1.3535]^T$
   $f(x_3) = 142.8066$
4. $x_4 = [2.0648 \quad 6.1394 \quad 0.3245 \quad -3.5741 \quad 3.8815 \quad 7.7102 \quad 5.3201]^T$
   $f(x_4) = 157.6519$
5. $x_5 = [6.0152 \quad 0.2568 \quad 5.0357 \quad -9.4945 \quad -0.8616 \quad 4.5374 \quad 7.5169]^T$
   $f(x_5) = 229.5855$
6. $x_6 = [-3.3795 \quad 6.432 \quad 3.521 \quad 2.1629 \quad -0.7394 \quad 4.7345 \quad 2.6164]^T$
   $f(x_6) = 99.6739$

Za daljnji račun treba posložiti jedinke po dobroti, od najbolje prema najlošijoj i uvrstiti u formulu (14). Tada se dobije nova vrijednost za očekivanu vrijednost:

- $m_{novi} = [0.189 \quad 6.3433 \quad 2.5172 \quad 2.0731 \quad -1.2628 \quad 4.7759 \quad 1.6986]^T$

Vidi se da su neke vrijednosti dosta promijenjene. Unutar prvih 50-ak generacija se očekivana vrijednost ne stabilizira te se to ni ne očekuje od nje.

Sljedeće na redu je ažurirati veličinu koraka i njezin evolucijski put po formulama (16) i (17) te se dobiju vrijednosti:

- $p_\sigma = [-0.0559 \quad 0.3533 \quad -0.4259 \quad 1.0982 \quad -1.1925 \quad 0.0408 \quad 0.3686]^T$
- σ_{novi} =4.6624

Potom se ista stvar radi, samo za kovarijantnu matricu. Nakon računanja vrijednosti prema formulama (19) i (21), jer se koriste ažuriranja matrice i matricom ranga jedan i matricom ranga *μ,* dobiju se vrijednosti:

- $p_C = [-0{,}0619 \quad 0{,}3914 \quad -0{,}4717 \quad 1{,}2166 \quad -1{,}321 \quad 0{,}0452 \quad 0{,}4083]^T$

- $$C = \begin{bmatrix} 0.9715 & -0.012 & -0.0019 & -0.0019 & 1.628\,10^{-4} & -6.079\,10^{-4} & -0.005 \\ -0.0012 & 0.9651 & -0.007 & 0.0183 & -0.0198 & 6.702\,10^{-4} & 0.0062 \\ -0.0019 & -0.007 & 0.9686 & -0.0218 & 0.0243 & -9.189\,10^{-4} & -0{,}0067 \\ -0.0019 & 0.0183 & -0.0218 & 1.0176 & -0.0634 & 0.0013 & 0.0177 \\ 1.628\,10^{-4} & -0.0198 & 0.0243 & -0.0634 & 1.0284 & -0.0014 & -0{,}0187 \\ -6.079\,10^{-4} & 6.702\,10^{-4} & -9.189\,10^{-4} & 0.0013 & -0.0014 & 0.9597 & 0.0014 \\ -0.005 & 0.0062 & -0{,}0067 & 0.0177 & -0{,}0187 & 0.0014 & 0.9677 \end{bmatrix}$$

Nakon izračuna kovarijantne matrice se mora napraviti njena dekompozicija na matrice B i D zbog ponovnog računanja u sljedećoj generaciji.

Po završetku jedne generacije i izračuna svih potrebnih parametara, postupak se ponavlja sve dok se ne postignu uvjeti zaustavljanja. Neki od njih mogu biti minimalna vrijednost dobrote, najveći broj generacija, broj generacija u kojem nema boljeg rješenje, preveliki raspon kovarijantne matrice (preveliki kondicijski broj), premali utjecaj parametara na očekivanu vrijednost i slično.

2. Ispitivanje

Prvi i najjednostavniji implementirani algoritam je (μ/μ, λ)-CMA-ES. Svojstvo ovog CMA-ES je da sve roditelje smatra jednako bitnima. Računanje srednje vrijednosti roditelja će biti aritmetička sredina tih jedinki. Drugim riječima, težinska funkcija ima formulu $w_i = \frac{1}{\mu}$.

Za razliku od njega, $(\mu/\mu_w, \lambda)$-CMA-ES ima definiranu težinsku funkciju koja nije ista za sve jedinke, te daje veću prednost jedinkama koje su bile uspješnije. Primjer formule za jednu takvu težinsku funkciju bi bio:

$$w_i = log(\mu + 1) - \log(i)\,, i \in \{1, \dots, \mu\} \tag{25}$$

Njihovi ulazni parametri su veličina populacije, broj roditelja, funkcija evaluacije, dimenzija i uvjeti završavanja algoritma (stagnacija, najveći dozvoljeni broj generacija i minimalno odstupanje dobrote).

Sljedeći algoritam koji se ispituje je , $(\mu/\mu_w, \lambda)$-A-CMA-ES gdje se radi još i promjena kovarijantne matrice sa negativnom matricom. Negativna matrica znači da se u obzir uzimaju i najgore jedinke populacije te se one oduzimaju od kovarijantne matrice, smanjujući pritom njihov utjecaj. Pritom će za izračun biti korištena formula (23).

Radi lakšeg ispitivanja algoritma, direktno će se mijenjat samo dimenzija problema. Svi ostali parametri će se mijenjati ovisno o dimenziji. Razlog tome je lakši prikaz funkcioniranja algoritma, jer su od strane autora već pretpostavljeni njihovi odnosi. Stoga će se parametri mijenjati na sljedeći način:

$$\lambda = 4 + 3\,\lfloor \log(N) \rfloor \tag{26}$$

$$\mu = \left\lfloor \frac{\lambda}{2} \right\rfloor \tag{27}$$

$$najveći\ broj\ generacija = \left\lceil \frac{1000\,N^2}{\lambda} \right\rceil \tag{28}$$

$$minimalno\ odstupanje = 10^{-5} \tag{29}$$

$$stagnacija = 10 + \frac{30\,N}{\lambda} \tag{30}$$

2.1. Rosenbrockova funkcija

Rosenbrockova funkcija je nekonveksna funkcija često korištena za testiranje optimizacijskih algoritama. Kao funkcija više varijabli, definirana je formulom:

$$f(x_1, x_2, \dots, x_N) = \sum_{i=1}^{N-1} [(1 - x_i)^2 + 100\,(x_{i+1} - {x_i}^2)^2\,] \tag{31}$$

Minimum te funkcije je u točki (1,1,…,1), iako ima i dosta svojih lokalnih minimuma, te se ispituje nad intervalom [-2.048, 2.048].

Funkcija je testirana nad dimenzijama problema 5,10 i 30.

Za problem dimenzije 5 nije teško predvidjeti da nema većih komplikacija. Algoritam brzo i točno pronalazi minimum. Najviše generacija treba CMA-ES koji nema implementiranu težinsku funkciju, potom onom koji ima te se kao najbolji pokazao A-CMA-ES iako češće uspije ne konvergirati prema minimumu. Po tablici (Tablica 1) se vidi da nema nekih većih razlika na ovako maloj dimenziji.

Tablica 1 Rosenbrockova funkcija, N=5, 15 pokretanja

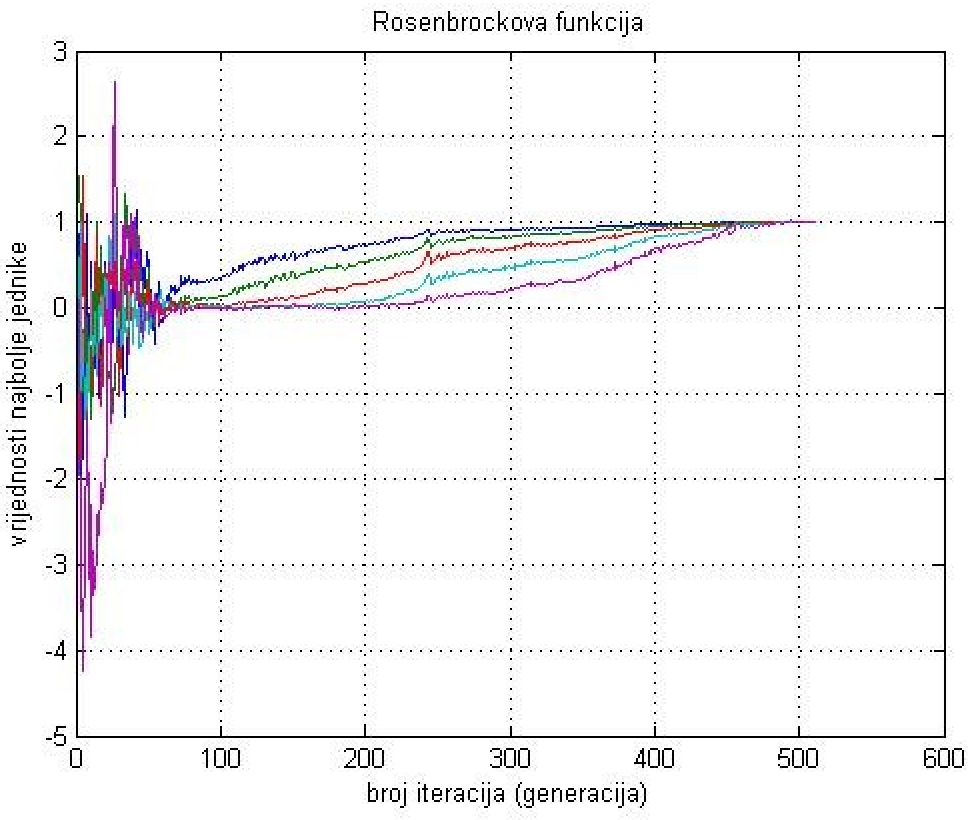
	Min	Max	Avg
Bestežinski CMA-ES	2.63819e-006	3.93084	0.26206298
Težinski CMA-ES	3.94759e-006	3.93084	0.52411832
Aktivni CMA-ES	2.64427e-006	3.93084	0.52411781

CMA-ES bez definiranih težina je samo jednom promašio i nije uspio konvergirati, dok su svim ostalim slučajevima pronalazi rješenje čija funkcija dobrote ima tek petu decimalu.

No usprkos dobrim rezultatima, ovoj inačici algoritma je trebalo u prosjeku preko 500 generacija da uspije pronaći optimum. Za razliku od toga, CMA-ES koji ima definirane težinske funkcije i to prema formuli (24), uspije u prosjeku nakon manje od 300 generacija. Po tablici se vidi da uspješnost nije puno drugačija; uspije samo dvaput konvergirati u krivo rješenje.

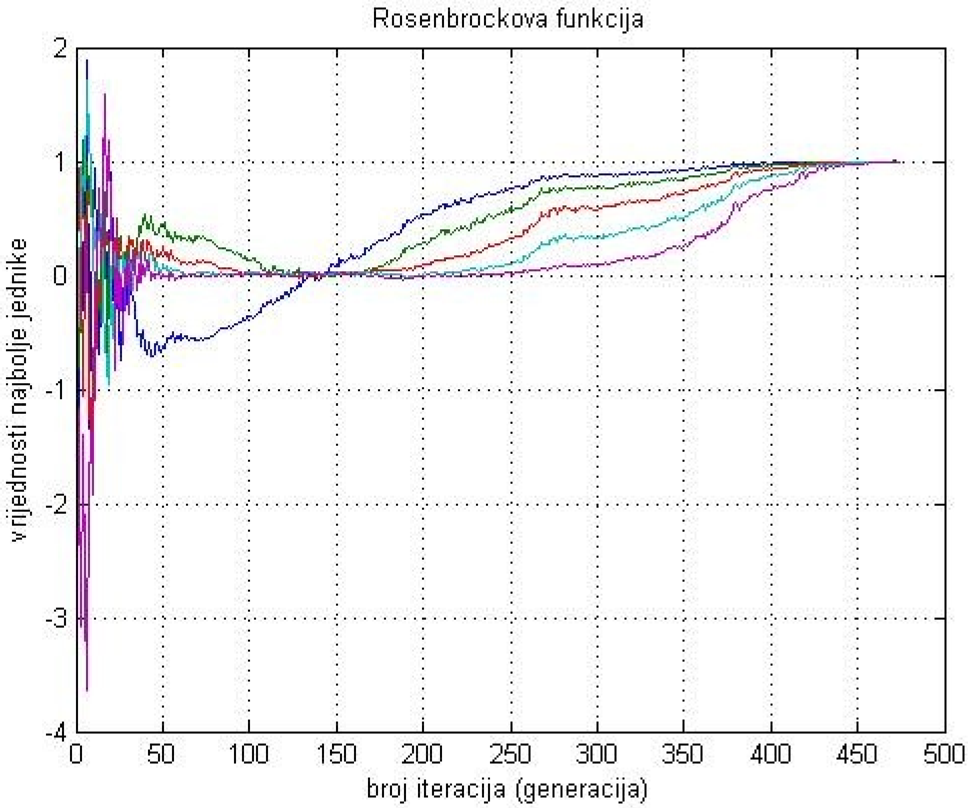
Uspješnost A-CMA-ES nije puno drugačija od težinskog CMA-ES. Otprilike im treba jednako generacija da konvergiraju, možda A-CMA-ES treba malo manje. Zgodno je primijetiti da svi „promašaji“ imaju istu vrijednost. To znači da postoji neki jaki lokalni optimum za koji su velike šanse da se zapne.

Na slikama 2.1, 2.2 i 2.3 se vidi kako se mijenja vrijednost najbolje jedinke prilikom jednog pokretanja za sva ti CMA-ES algoritma. U pravilu je način isti: unutar prvih 50 generacija se ne stabilizira, te se potom počinje polako isticati sve bolje i bolje rješenje.



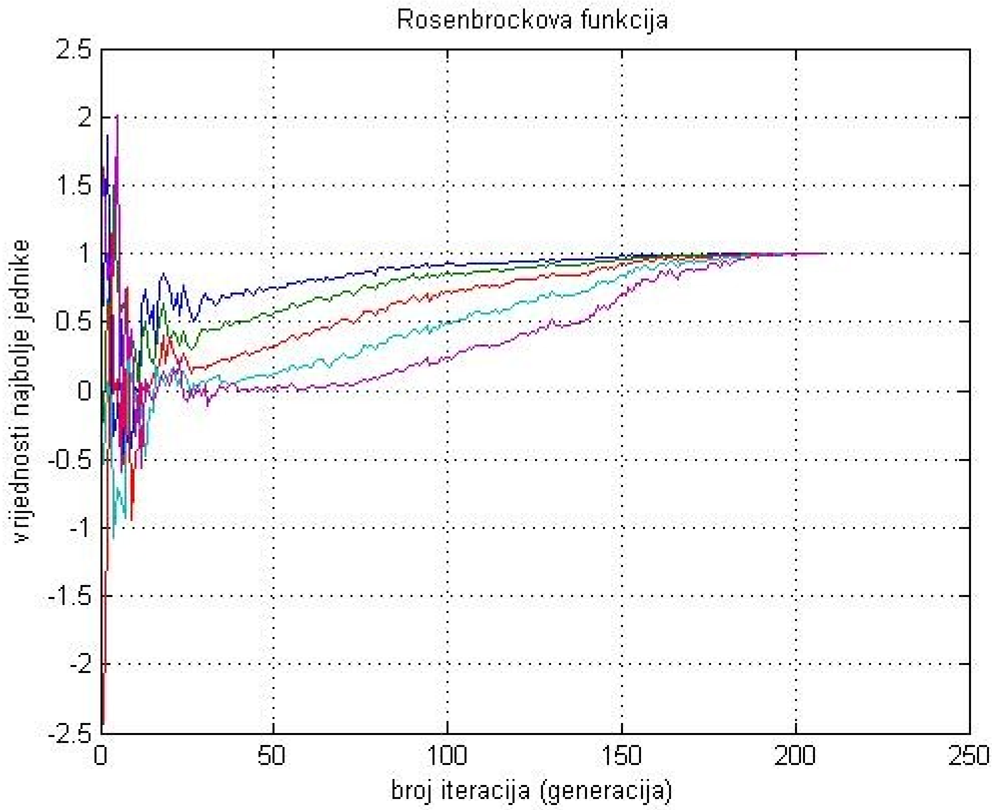
Slika 2.1 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=5, bestežinski CMA-ES

Za sve generacije je uvjet zaustavljanja je preveliki kondicijski broj matrice, što je bilo dovoljno da se nađe optimum.

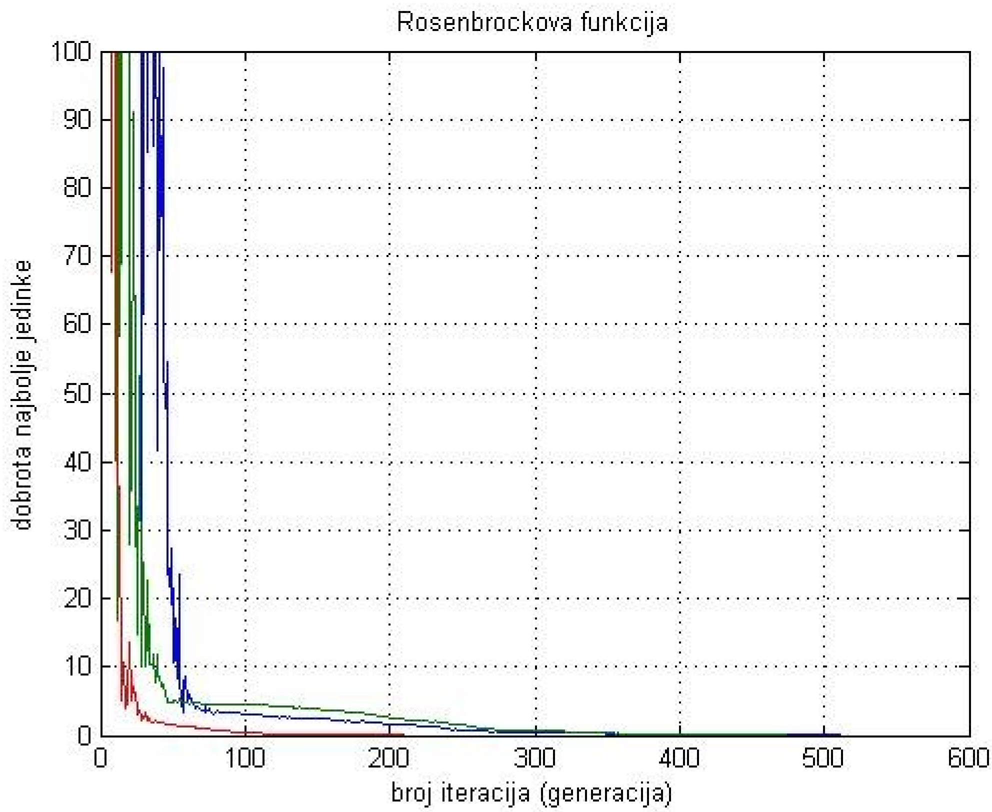


Slika 2.2 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=5, težinski CMA-ES

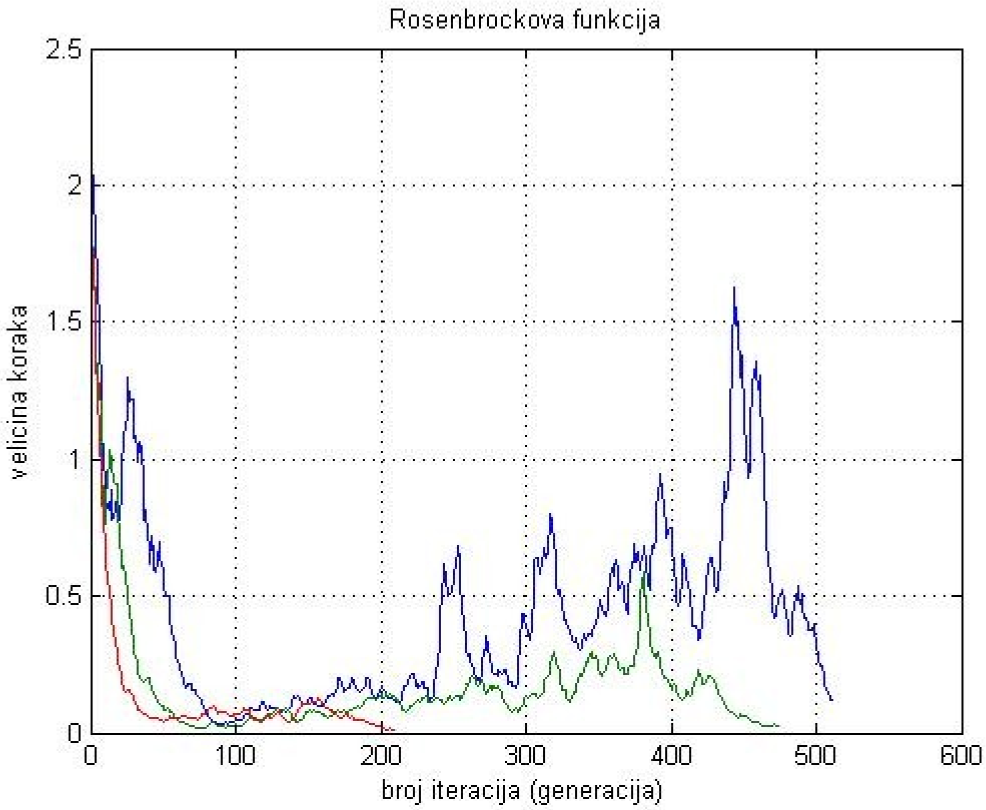
Iako zbog slike 2.2 izgleda kao da i težinskom CMA-ES treba skoro 500 generacija da konvergira, to je zbog toga što je skoro završio u lokalnom optimumu u točki (-1,1,1,1,1) zbog čega se trajanje produljilo za sigurno 100 generacija. No zbog dobre strategije pretraživanja prostora, CMA-ES se izvukao te nastavio konvergirati prema globalnom optimumu.



Slika 2.3 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=5, A-CMA-ES



Slika 2.4 Dobrota najbolje jedinke, Rosenbrockova funkcija, N=5



Slika 2.5 Veličina koraka, Rosenbrockova funkcija, N=5

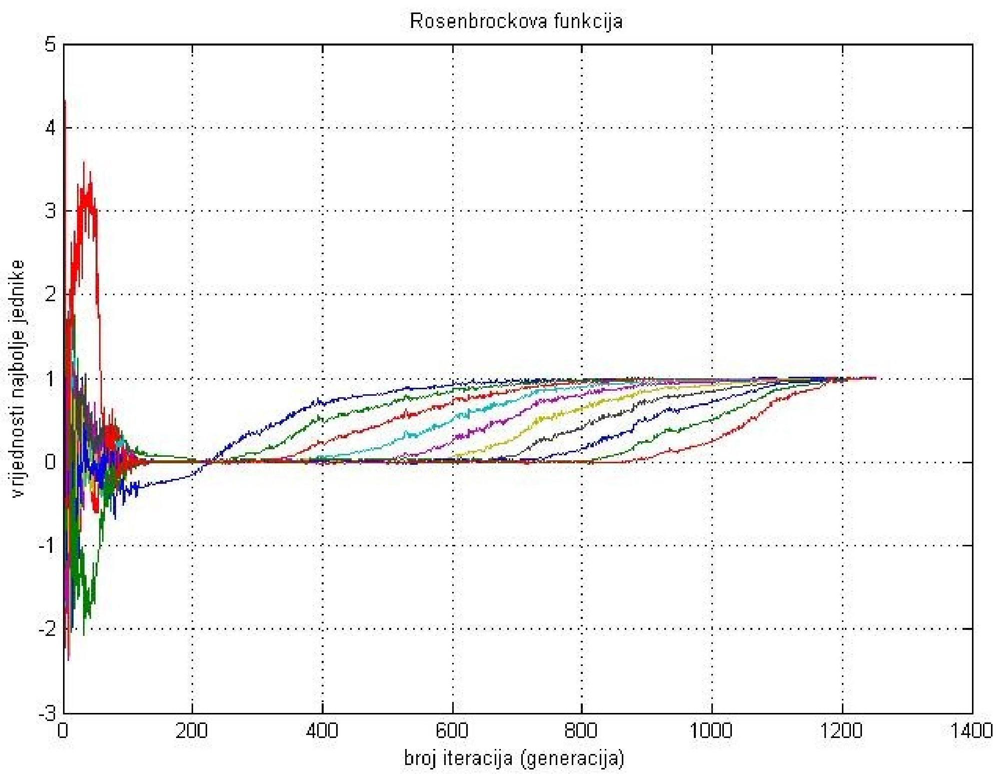
Slike 2.4 i 2.5 služe da bi se prikazalo koliko su različiti zadani CMA-ES algoritmi. Plavom, zelenom i crvenom bojom su prikazani redom bestežinski, težinski i aktivni CMA-ES. Na slikama se vidi da bestežinskom treba nešto dulje vremena da dobrota jedinke i veličina koraka padnu na neku manju vrijednost, te kasnije dobrota sporije pada prema nuli. Između 100. i 200. generacije se težinski i bestežinski CMA-ES dosta poklapaju jer se težinski pokušava izvući iz lokalnom optimuma. U usporedbi s njima, A-CMA-ES veoma brzo pada prema nuli te puno prije završava. Nagle promjene veličine koraka objašnjava to što se sve vrijednosti jedinke konstantno mijenjaju. Prilikom naglih skokova se naglo pomiče prostor pretraživanja zbog nekog boljeg rješenja koje nije u centru tog prostora. Zbog toga se nagli skokovi na slici 2.5 mogu poistovjetiti s jačim pomacima na slici 2.1.

Usporedbe radi, isti problem je pokrenut na još dva optimizacijska algoritma: *Particle Swarm Optimization* i na *steady state* genetskom algoritmu. Za PSO su korišteni unaprijed zadani, pretpostavljeni parametri te iste granice i dimenzija problema. Za takve slučajeve, PSO je imao dobrotu jedinke oko 5, a SS genetski algoritam varira između 0.2 i 43.

Za problem dimenzije 10 je bilo više poteškoća. Trajanje izvršavanja algoritama se dosta produljilo, potrebno je puno više generacija proći, a i češće se događaju lošiji rezultati. Za bestežinski CMA-ES je u prosjeku trebalo 1072 generacije da dođe do rješenja, težinskome 775, a aktivnom 667. No cijena aktivnog CMA-ES je ta što je čak četiri od petnaest puta zapeo u lokalnom optimumu i nije došao do globalnog, dok se težinskom i bestežinskom to dogodilo samo jednom. Ta je razlika vidljiva u tablici (Tablica 2) gdje je prosječna vrijednost A-CMA-ES dosta velika.

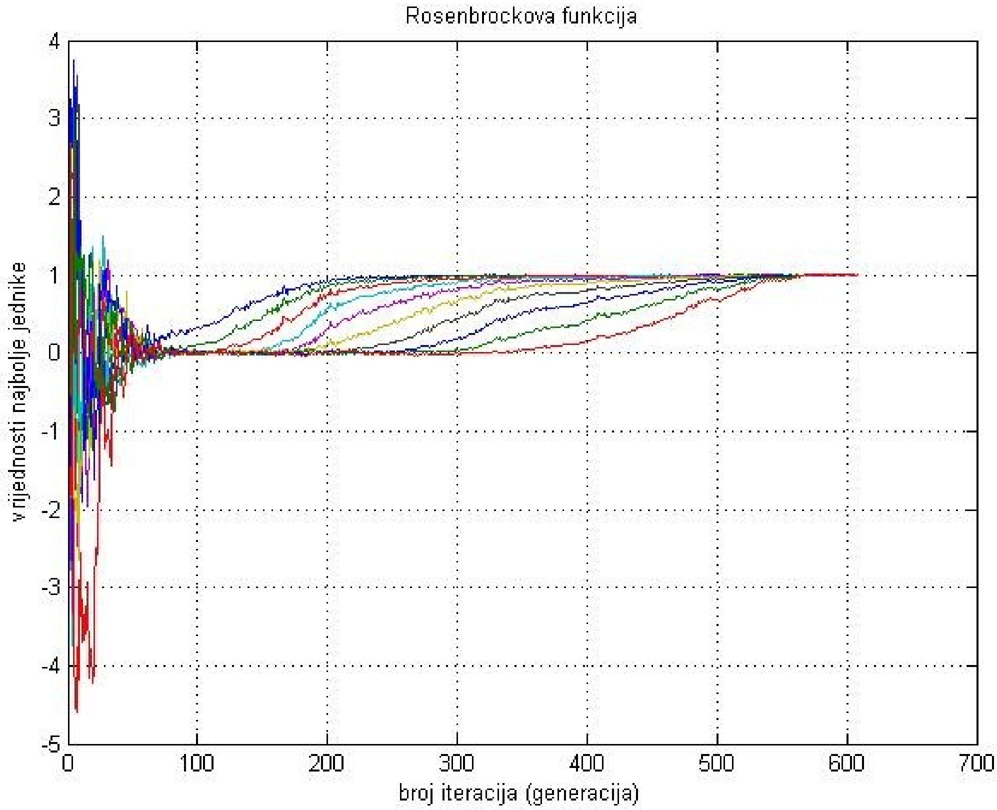
Tablica 2 Rosenbrockova funkcija, N=10, 15 pokretanja

	Min	Max	Avg
Bestežinski CMA-ES	3.03544e-006	3.98658	0.53155069
Težinski CMA-ES	4.96287e-006	3.98658	0.26577954
Aktivni CMA-ES	4.51206e-006	3.98658	1.06309339

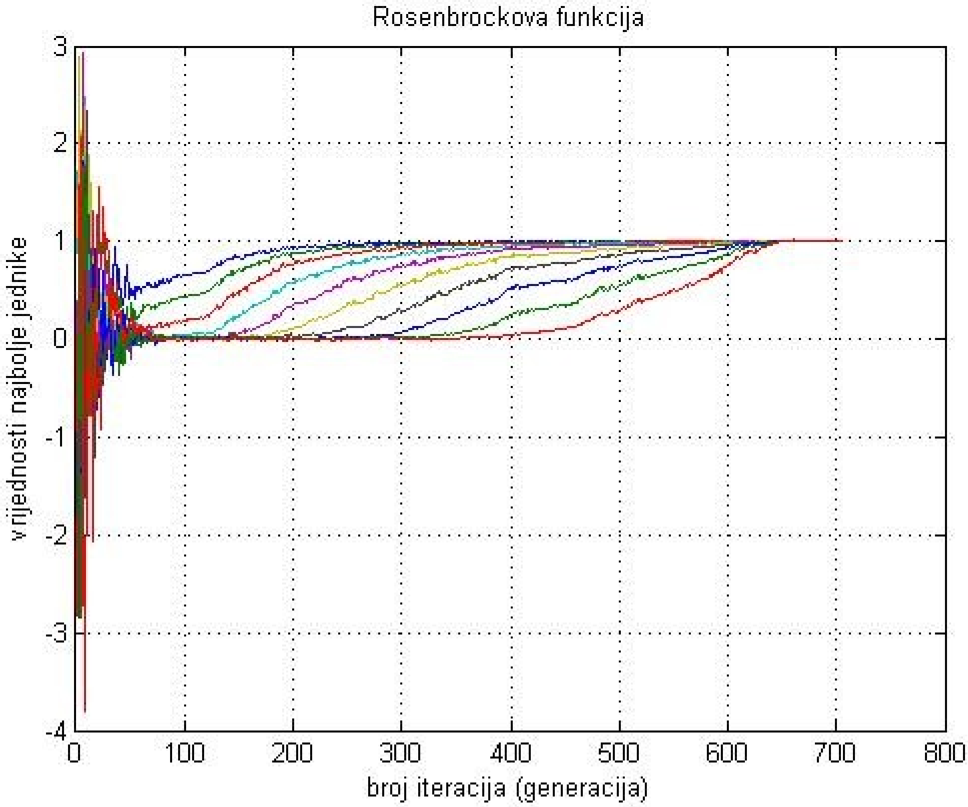


Slika 2.6 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=10, bestežinski CMA-ES

Na slici 2.6 se opet vidi kako se produljilo izvođenje algoritma zbog bježanja u lokalne optimume, ali se poslije 200. generacije počinju formirati rješenja koja konvergiraju prema globalnom optimumu. Ista stvar se vidi i na slikama 2.7 i 2.8.Razlika je u tome da je cijeli proces brži, te umjesto 1200 generacija, treba oko 600,700.

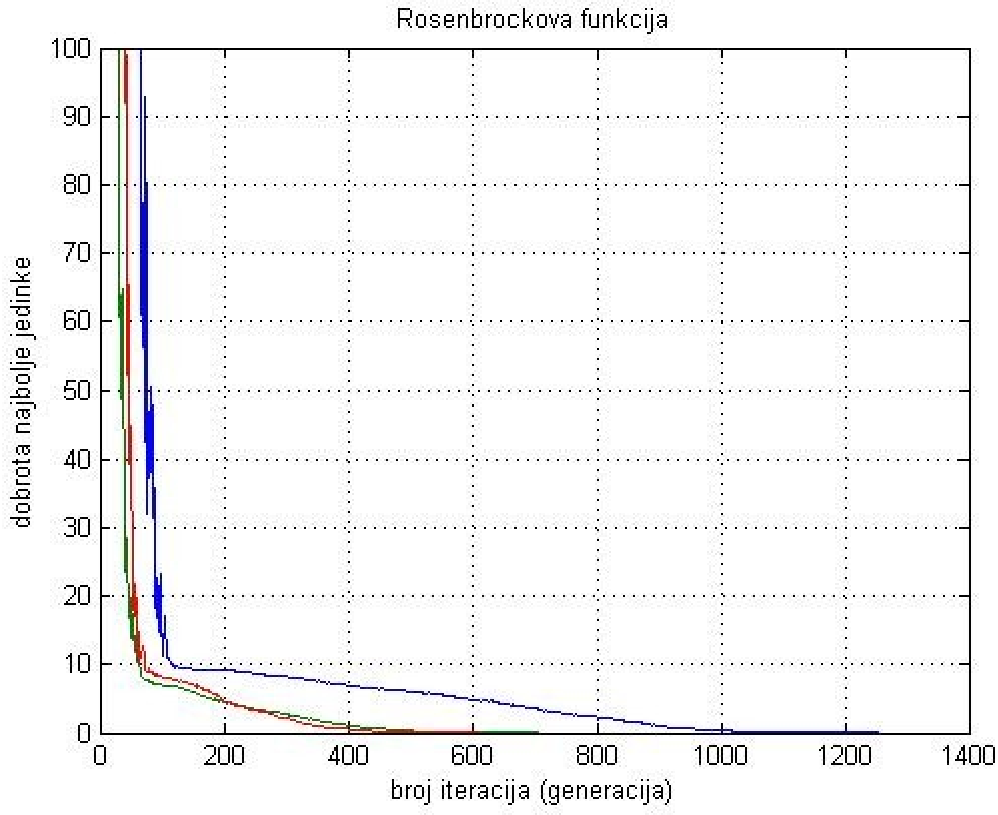


Slika 2.7 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=10, težinski CMA-ES



Slika 2.8 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=10, A- CMA-ES

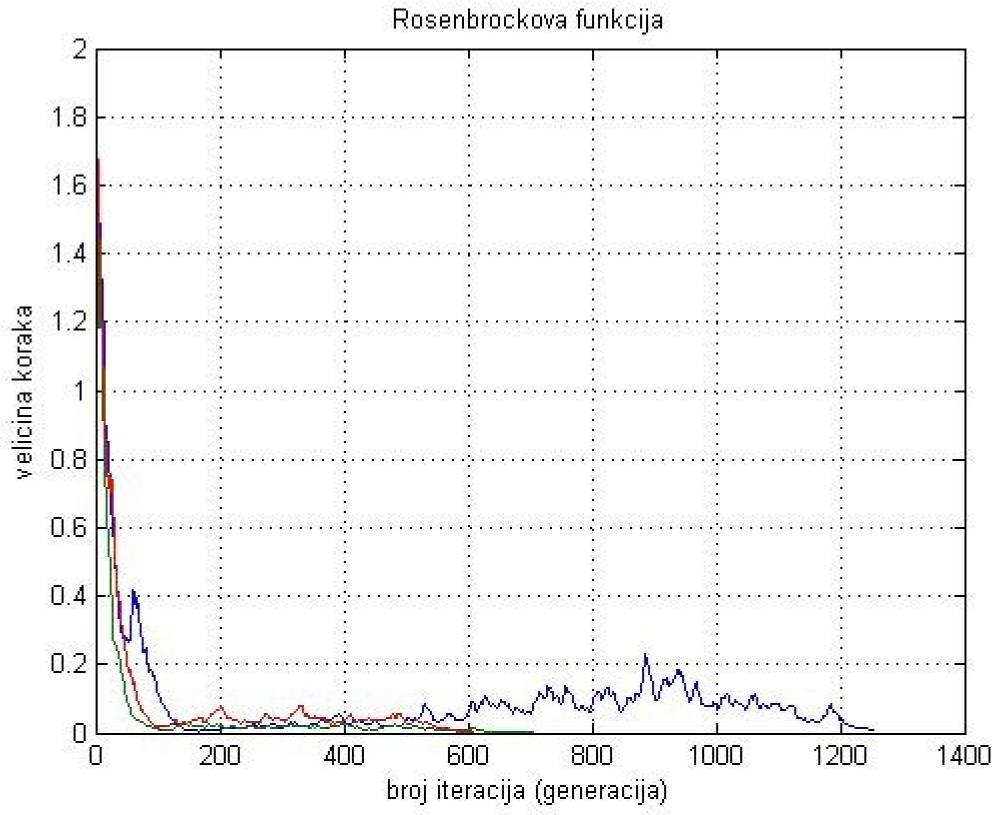
Ono što je zanimljivo vidjeti je slika 2.9 gdje se bolje vide razlike između algoritama. Bestežinski CMA-ES (plavo obojana funkcija na grafu) je osjetno lošiji. Osim što mu treba proći više generacija da dobrota jedinke padne ispod deset, nakon toga i dalje veoma sporo pada, kao da se teško prilagođava. Za razliku od njega, težinski i aktivni CMA-ES imaju puno strmije padanje vrijednosti dobrote. Njihova vrijednost dobrote u 400. generaciji odgovara vrijednosti dobrote bestežinskog CMA-ES u otprilike 1000. generaciji.



Slika 2.9 Dobrota najbolje jedinke, Rosenbrockova funkcija, N=10

Za razliku od grafa dobrote najbolje jedinke za N=5, kod dimenzije 10 je potrebno dvostruko više generacija da se prođe početni dio u kojem vrijednost dobrote naglo pada. Čak i vrijednost do koje naglo pada je dvostruko veća za veću dimenziju.

Veličina koraka se slično ponaša za obadvije dimenzije, jedino za veću dimenziju ne „divlja“ toliko kasnije. Razlog tome je vjerojatno manje promjene između generacija, odnosno najboljih jedinki unutar generacija.



Slika 2.10 Veličina koraka, Rosenbrockova funkcija, N=10

Slike 2.9 i 2.10 lijepo ilustriraju kako A-CMA-ES na većim dimenzijama nije znatno bolji. Malo bolji i brži je, ali kada se uzme u obzir i da češće zapne u lokalnim optimumima, možda i nije tako dobra opcija prilikom potrage za globalnim optimumom.

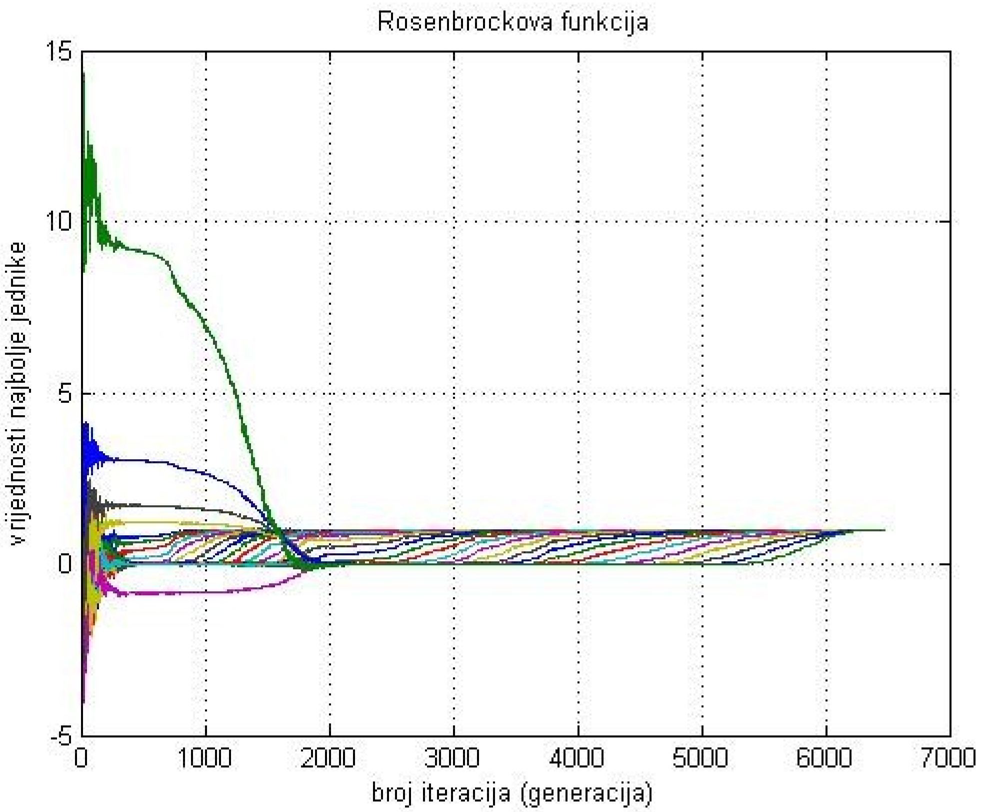
U usporedbi sa PSO i SS genetskim algoritmom, CMA-ES uspijeva pronaći minimum funkcije. Za PSO je najbolja vrijednost dobrote nešto malo manja od 30, a za SS je varira od 0.5 do 990, s time da je prosječna oko 180, te se CMA-ES opet pokazuje kao dobar izbor algoritma.

Sljedeća dimenzija nad kojom su testirani algoritmi je veličine 30. Ispitivanje nad tolikom dimenzijom je potrajalo jer osim što je narasla dimenzija problema, narastao je i broj generacija potrebnih za pronalazak dovoljno dobrog rješenja.

Tablica 3 Rosenbrockova funkcija, N=30, 15 pokretanja

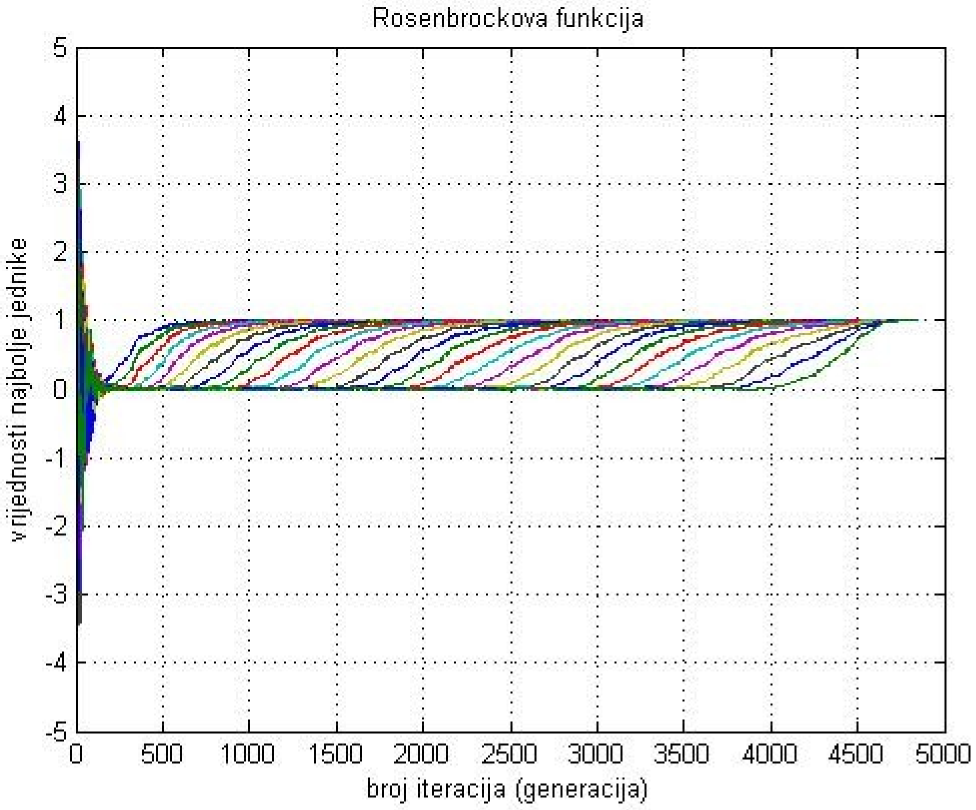
	Min	Max	Avg
Bestežinski CMA-ES	9.03181e-006	3.98662	0.53155767
Težinski CMA-ES	9.03693e-006	3.98662	0.53155769
Aktivni CMA-ES	7.73321e-006	9.94808e-006	9.28984e-006

Za bestežinskom CMA-ES je potrebno u prosjeku oko 7500 generacija, težinskom oko 5080, a aktivnom za tisuću generacija manje, prosječno oko 4080 generacija. Tu se A-CMA-ES opet pokazao najbržim i ovaj put i najtočnijim! Naime, u petnaest pokretanja algoritma, niti jednom nije zapeo u lokalnom optimumu. To se najbolje vidi u najvećoj vrijednosti u tablici (Tablica 3).

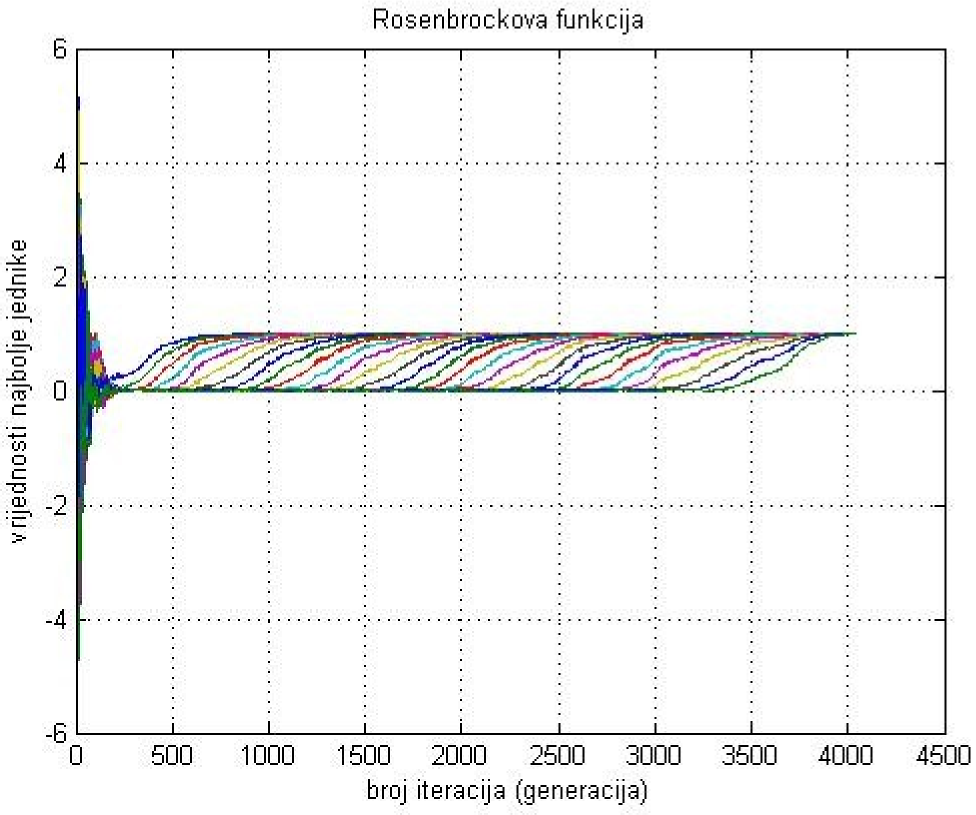


Slika 2.11 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=30, bestežinski CMA-ES

Na slici 2.11 se vidi kako zapravo bestežinskom CMA-ES algoritmu treba stvarno dosta generacija prije nego završi, s time da je ovaj primjer još iznadprosječno dobar. Razlog zbog kojeg zadnja tri elementa poprimaju veće vrijednosti nije poznat. Vjerojatno je to samo do tog pokretanja algoritma. Težinski i aktivni CMA-ES normalno funkcioniraju (slika 2.12 i 2.13): kroz prvih 100-ak generacija vrijednosti dosta skaču, te se poslije toga smiruju i postupno kreću prema jedan.

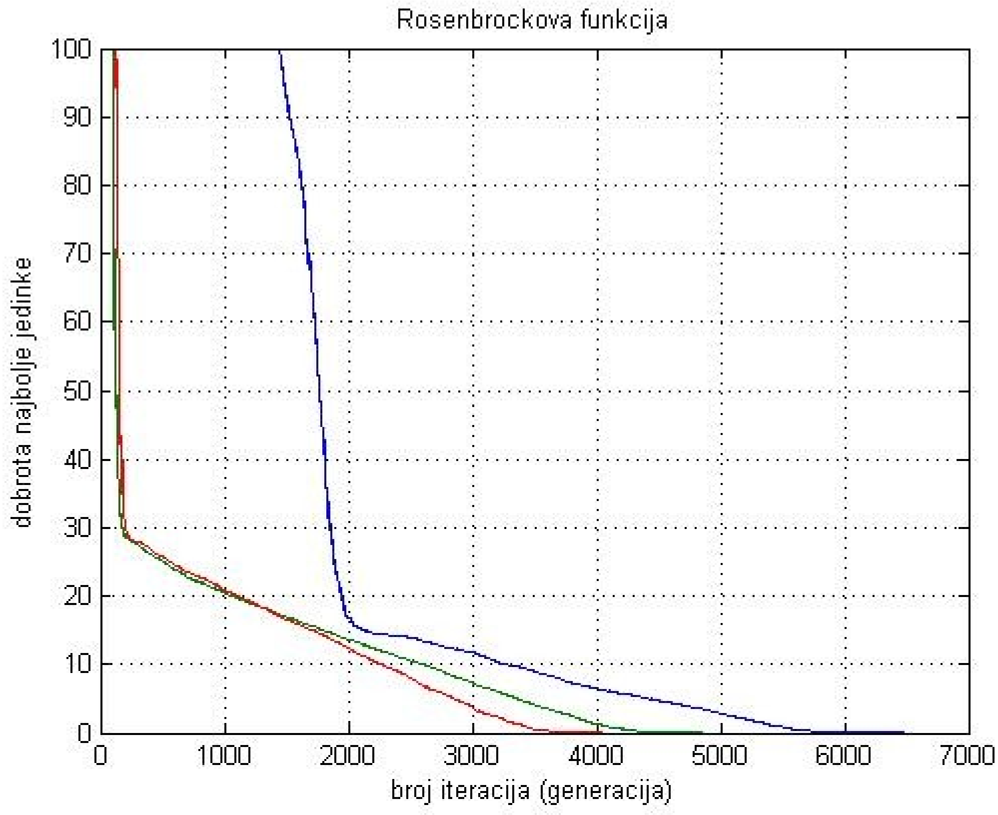


Slika 2.12 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=30, težinski CMA-ES



Slika 2.13 Vrijednost najbolje jedinke, Rosenbrockova funkcija, N=30, A-CMA-ES

Na slici 2.14 se jasno vidi koliko je zapravo bestežinski CMA-ES lošiji od druga dva. Vrijednost dobrote kroz generacije se puno lošije smanjuje, dulje vremena mu treba da počne naglo padati, a i to naglo nije jednako naglo kao i kod druga dva algoritma. Istina je da i bestežinski CMA-ES uspije naći optimalno rješenje, no za to mu treba gotovo dvaput više generacija.



Slika 2.14 Vrijednost dobrote najbolje jedinke, Rosenbrockova funkcija, N=30

CMA-ES se pokazao uspješnim prilikom rješavanja ovog optimizacijskog problema, čak i na većim dimenzijama. Naravno da to ne garantira da će biti jednako uspješan i prilikom rješavanja drugih optimizacijskih problema, no već postoje različite inačice CMA-ES koje su vrlo uspješne prilikom rješavanja skupine različitih problema.

Zaključak

CMA-ES se pokazao kao jako dobrim sredstvom prilikom rješavanja kontinuiranih optimizacijskih problema. Zbog svojih svojstava i načina traženja minimuma, veoma je otporan na funkcije sa puno lokalnih optimuma jer prilagođuje prostor pretraživanja ovisno o svim jedinkama iz populacije. Ako su parametri dobro podešeni, konvergencija neće biti ni prebrza ni prespora. Prebrza konvergencija znači da će „glavom bez obzira" krenuti prema najboljem rješenju, prestajući pritom razmatrati ostale vrijednosti te će vrlo vjerojatno zaglaviti u nekom lokalnom ekstremu. Ako je pak konvergencija prespora, neće nikada naći optimum jer neće dovoljno smanjivati prostor pretrage. Većina tih parametara je odabrana i isprobana te postoje vrijednosti koje se preporučuju za optimalno izvođenje algoritma.

Uspješnost također ovisi i o varijanti CMA-ES algoritma. Na mnogim ispitivanjima se može vidjeti da nije svejedno koja je inačica odabrana, neće niti otprilike biti jednako uspješni.

CMA-ES je kao algoritam za rješavanje optimizacijskih problema sasvim dobar, no svi uvijek imaju svoje prednosti i mane, te tako ni CMA-ES nije savršen. Iza uspješnosti ovog algoritma stoji preko 10 godina prilagođavanja parametara i strategija.

Literatura

Svaki autor piše popis literature na kraju svog poglavlja. Popis literature se piše stilom literatura.

[1] SCHOLARPEDIA, Evolution Strategies: http://www.scholarpedia.org/article/Evolution_Strategies, 20.5.2012.

[2] WIKIPEDIA, CMA-ES: http://en.wikipedia.org/wiki/CMA-ES, 20.5.2012.

[3] WIKIPEDIA, Covariance matrix: http://en.wikipedia.org/wiki/Covariance_matrix, 20.5.2012.

[4] AUGER A., HANSEN N.: *Tutorial: CMA-ES - Evolution Strategies and Covariance Matrix Adaptation*, GECCO'11, lipanj 2011. http://www.lri.fr/~hansen/gecco2011-CMA-ES-tutorial.pdf

[5] WIKIPEDIA, Multivariate normal distribution, http://en.wikipedia.org/wiki/Multivariate_normal_distribution#Drawing_values_from_the_distribution, 20.5.2012.

[6] WIKIPEDIA, Box-Muller transform, http://en.wikipedia.org/wiki/Box_Muller_transform , 20.5.2012.

[7] DODLEK Z., *Statistička usporedba rezultata algoritama zasnovanih na evolucijskom računanju*. Fakultet elektrotehnike i računarstva, svibanj 2009.

[8] JASTREBSKI G., ARNOLD D.: *Improving Evolution Strategies through Active Covariance Matrix Adaptation*, Evolutionary Computation, 2006

[9] LOSHCHILOV I., SCHOENAUER M., SEBAG M.: *Not all Parents are equal for MO-CMA-ES*, Evolutionary Multi-Criterion Optimization, 2011

[10] LRI: Homepage of Nikolaus Hansen, http://www.lri.fr/~hansen/, 20.5.2012.

Sažetak

CMA-ES je u zadnje vrijeme dosta popularna evolucijska strategija. Jedan od razloga je njegova uspješnost na „teškim“ funkcijama, odnosno nekoveksnim, nederivabilnim funkcijama sa puno lokalnih optimuma. Kada se tome doda još i veća dimenzija prostora, stvar se poprilično zakomplicira. Drugi razlog je mogućnost prilagođavanja parametara, jer ih ima dosta, i biranje što uopće mogu biti jedinke unutar jedne populacije.

Sama populacija se bira po normalnoj distribuciji. Parametri te distribucije se mijenjaju iz generacije u generaciju tako da se svaki put biraju sve bolje i bolje jedinke, sve dok se ne dođe do optimuma funkcije. Nad nekoliko testiranih primjera se CMA-ES pokazala jako uspješna.

Ključne riječi: evolucijska strategija, populacija, normalna distribucija, kovarijantna matrica, evolucijski put

Summary

Recently, CMA-ES is quite popular evolutionary strategy. One of the reasons for that is its success with non-convex, non-smooth and noisy functions with many local optima. When you add large dimensionality, the thing can be quite complicated. Another reason is the possibility of adjusting the parameters, because there are plenty of them, and choose what can individuals within a population be.

The population is elected by a normal distribution. The parameters of these distributions change from generation to generation so that every time is choosen better and better individuals, until it reaches the global optimum. CMA-ES is tested with several examples and proved very successful.

Keywords: evolutionary strategies, population, normal distribution, covariance matrix, evolution paths

Skraćenice

CMA-ES	*Covariance Matrix Adaptation Evolution Strategy*	evolucijska strategija sa kovarijantnom matricom
ES	*Evolution Strategy*	evolcijska strategija
MO-CMA-ES	M*ultiobjective optimization CMA-ES*	CMA-ES sa višeciljnom optimizacijom
LR-CMA-ES	*Local Restart CMA-ES*	CMA-ES sa ponovnim pokretanjem
IPOP-CMA-ES	*Increasing population size CMA-ES*	CMA-ES sa povećavanjem populacije
BIPOP-CMA-ES	*Bipopulation CMA-ES*	CMA-ES sa dvije populacije
G-CMA-ES	*Global CMA-ES*	Globalni CMA-ES
L-CMA-ES	*Local CMA-ES*	Lokalni CMA-ES
A-CMA-ES	*Active CMA-ES*	Aktivni CMA-ES
PS-CMA-ES	*Particle swarm* CMA-ES	CMA-ES sa rojem čestica