INTERNACIONALNI NAUČNO-STRUČNI SKUP
GRAĐEVINARSTVO - NAUKA I PRAKSA

ŽABLJAK, 03-07. MARTA 2008.

Živojin Praščević[1], *Nataša Praščević*[2]

METODA OPTIMIZACIJE ČLANOVA ROJEVA

Rezime

U ovome radu je prikazana jedan noviji postupak za optimizaciju kontinualnih funkcija sa i bez ograničenja koji spada u stohastičke heurističke metode, odnosno u tzv. metode inteligencije rojeva. Postupak je matematički i algoritamski veoma jednostavan i može se uspešno primeniti za rešavanje velikog broja problema nelinearne optimizacije sa i bez ograničenja. U radu je predložen postupak za tretman ograničenja i napisan odgovarajući kompjuterski program. Na kraju je na jednom primeru optimizacije armiranobetonskog (AB) nosača prikazana primena ove metode i izvršeno upoređenje sa rezultatima dobijenim primenom gentskog algoritma.

Ključne riječi

nelinearna optimizacija, heurističke metode, inteligencija rojeva

PARTICLE SWARM OPTIMIZATION

Summary

In this work is presented one newer method for optimization of continuos functions which belongs to the heuristic stochastic methods, ie. to the methods of so called swarm intelligence. This method is mathematically and algorithmicaly very simple and might be applied successfully for solving a large number of nonlinear optimization problems with and without constraints. One proposal for tretmant of constraints in this method is proposed in the paper and a computer program is written according to describd procedure. The method is ilustrated by one example of RC beam optimization and results obtained by this method are compared with the results received by the genetic algorithm.

Key words

nonlinear optimization, heuristic methods, swarm inteligency

[1] *Prof. dr Živojin Praščević, dipl.grad.inž. Građevinski fakultet, Beograd, e-mail: zika@grf.bg.ac.yu*

[2] *Doc. dr Nataša Praščević,dipl.grad.inž., Građevinski fakultet, Beograd, e-mail: natasa@grf.bg.ac.yu*

1. UVOD

Metoda optimizacije članova (čestica) rojeva (engl. *Particle swarm optimization*) spada u evolucione kompjuterske metode zasnovane na populaciji za optimizaciju kontinualnih funkcija sa i bez ograničenja. Ona je jedna od metoda tzv. *inteligencije rojeva* u koje se ubraja *metoda optimizacije kolonije mrava* (engl. *Ant colony optimization*) koja se koristi za rešavanje problema kombinatorne optimizacije. Metodu su predložili socijalni psiholog James Kennedy (Džejms Kenedy) i elektroinženjer Russell C. Eberhart (Rasel C. Eberhart) 1995. god. [1]. Iako je to relativno nova metoda, ona je od svog nastanka imala nekoliko izmena i dopuna i o metodi je napisan veći broj radova, knjiga ili poglavlja u knjigama.

Metoda je inspirisana socijalno-psihološkim ponašanjem ljudi i konsultovanjem sa drugim ljudima prilikom donošenja različitih odluka, kao i ponašanjem u prirodi rojeva čestica koje mogu biti ptice, ribe, pčele i sl. prilikom njihovog kretanja u jatima ili rojevima radi pronalaženja hrane, cvetnog nektara za stvaranje meda ili selidbe na druge lokacije. Određene pravilnosti i ponašanje u kretanjima ptica i riba u jatima još ranije su zainteresovale zoologe, kako ističu J. Kennedy i R. C. Eberhart [1], i bile su predmet njihovog proučavanja.

Prilikom donošenja odluka, donosioc odluke razgovara sa drugim ljudima o tom problemu, prikuplja razne informacije, mišljenja, savete i iskustva drugih ljudi, upoređuje ih sa svojim iskustvom, saznanjem i shvatanjem i na osnovu toga donosi odluku. Odluka se sastoji od niza konkretnih i abstraktnih činjenica i zaključaka i cilj je da se dođe do optimalnog ili najprihvatljivijeg rešenja u datoj situaciji i okolnostima. Čini se da i ptice, ribe, pčele ili druga živa bića koja se kreću u jatima ili rojevima, a koja imaju u odnosu na čoveka veoma male saznajne i kreativne mogućnosti, podešavaju instiktivno svoj položaj u odnosu na svoja prethodna stanja i položaje ostalih članova roja ili jata, koristeći svoja čula i biološka svojstva koja poseduju radi održavanja života. Pri tome kretanju nailaze na razne prepreke i opasnosti od kojih neke mogu biti fatalne.

Ova metoda za rešavanje problema nelinearnog programiranja, odnosno optimizacije, kako će se videti iz daljeg izlaganja, je matematički i algoritamski veoma jednostavna i daje u mnogim slučajevima rezultate visoke tačnosti. Zbog toga ima određene prednosti za rešavanje nekih problema nad nekim drugim složenijim heurističkim metodama kao što su genetski algoritmi, evolutivno programirnje i dr., kao i na "klasičnim" numeričkim metodama zasnovanim na teoriji nelinearnog programiranja i uslovima Korusha-Kuhna-Tuckera.

Pošto se u ovoj metodi simuliraju početna rešenja primenom Monte Carlo simulacija i ta rešenja poboljšavaju u iteracijama dok se ne dobije optimalno rešenje, ona spada u stohastičke i heurističke metode.

2. ZADATAK OPTIMIZACIJE – NELINEARNI PROGRAM

Treba odrediti vrednost funkcije cilja

$$z = \min f(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \ldots, x_n], \tag{1}$$

sa uslovima ograničenja

$$g_i(\mathrm{x}) \leq 0, \quad i = 1,2,...,m. \tag{2}$$

Ovim uslovima ograničenja treba dodati još i uslove

$$\min x_j \leq x_j \leq \max \leq x_j, \quad j = 1,2,...,n. \tag{3}$$

Vektor x je tačka u n-dimenzionalnom prostoru R^n a ograničenja (2) i (3) određuju skup dopustivih rešenja D.

3. ALGORITAM OPTIMIZACIJE

U prvoj iteraciji u oblasti dopustivih rešenja D simuliraju se metodom Monte Carlo vektori x_p (p=1,2,...,n_p), koji prestavljaju u prvoj iteraciji vektore položaja članova (čestica) roja u skupu D u kojem se kreću, čije su komponente

$$\begin{aligned} &x_{j,p}^{(1)} = \min x_{1,p} + (\max x_{1,p} - \min x_{j,p})(rand), \\ &j = 1,2,...,n; p = 1,2,...,n_p; \quad 0 \leq (rand) \leq 1; \end{aligned} \tag{4}$$

gde je $(rand)$ slučajni broj uniformne raspodele, n broj nepoznatih, a n_p je broj članova roja.

Za svaki od ovih vektora $\mathrm{x}_p = [x_{1,p}, x_2, \ldots, x_{n,p}]$ određuje se funkcija cilja

$$f_p^{(1)} = f(\mathrm{x}_p^{(1)}) \quad p=1,2,\ldots,n_p. \tag{5}$$

Između simuliranih vektora $\mathrm{x}_p^{(1)}$ bira se onaj za koji funkcija cilja $f_p^{(1)}$ ima najmanju vrednost i taj vektor predstavlja najbolje globalno rešenje $\mathrm{x}_g^{(1)}$ u prvoj iteraciji, za koji važi

$$\min f_p^{(1)} = z^{(1)} = f(\mathrm{x}_g^{(1)}). \tag{6}$$

U nekoj sledećoj iteraciji k = 2,3,4,. . . . za poznato $\mathrm{x}_p^{(k-1)}$ određuje se vektor položaja člana (čestice) roja $\mathrm{x}_p^{(k)}$ prema izrazu

$$\mathrm{x}_p^{(k)} = \mathrm{x}_p^{(k-1)} + \mathrm{v}_p^{(k-1)}; \; k = 2,3,...; \; p = 1,2,...,n_p. \tag{7}$$

gde je $\mathrm{v}_p^{(k-1)} = [v_{1,p}^{(k-1)}, v_{2,p}^{(k-1)}, ..., v_{n,p}^{(k-1)}]$ promena vektora položaja člana (čestice) roja p posle iteracije k-1. Ovaj vektor se naziva i *vektor brzine* člana (čestice) roja p i sračunava se prema obrascu

$$\mathbf{v}_p^{(k-1)} = \omega \mathbf{v}_p^{(k-2)} + \varphi_1[\mathbf{x}_{l,p}^{(k-1)} - \mathbf{x}_p^{(k-1)}](rand) + \\ + \varphi_2[(\mathbf{x}_g^{(k-1)} - \mathbf{x}_p^{(k-1)}](rand). \tag{8}$$

$\mathbf{x}_{l,p}^{(k-1)}$ je najbolji položaj člana (čestice) roja p kojem odgovara najmanja vrednost funkcije cilja u prethodnim iteracijama $1,2,\ldots k$-1. Ovaj vektor, odnosno komponenta naziva se komponenta *saznanja* i njome se unosi u račun saznanje o najboljem položaju svakog člana roja u prošlosti, odnosno prethodnim iteracijama.

$\mathbf{x}_g^{(k-1)}$ je vektor položaja onog člana (čestice) roja za koji funkcija cilja u iteraciji k-1 ima najmanju vrednost $z^{(k-1)}$. Ovaj vektor naziva se i *socijalni* ili *socijalna komponenta*, kojom se uzima u obzir najbolji trenutni položaj pomenutog člana roja.

ω je *faktor inercije*, a uveli su ga Y. Shi i R. C. Eberhart [2] i on je $\omega \leq 1$, često se uzima ω=0.9 i on utiče na smanjenje vektora promene (brzine) u narednim iteracijama.

φ_1 i φ_2 su *faktori učenja* (*learning factors*) i njima se određuje relativni uticaj komponente saznanja i socijalne komponente na položaj člana (čestice) roja u sledećoj iteraciji k. Često se ovi koeficijenti uzimaju da su $\varphi_1 \approx 2$, $\varphi_2 \approx 2$.

(*rand*) je, kako je već rečeno, slučajni broj uniformne raspodele i svaki put se simulira u gornjem izrazu drugi broj.

M. Clerc (Klerk) [3] je umesto faktora inercije uveo *faktor suženja* (*constriction factor*) K koji se sračunava prema izrazu

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\,|}, \quad \varphi = \varphi_1 + \varphi_2 > 4. \tag{9}$$

Ovim faktorom se utiče na smanjenje promene položaja (brzine) člana (čestice) roja u narednim iteracijama, tako da se ona računa prema izrazu

$$\mathbf{v}_p^{(k-1)} = K\{\mathbf{v}_p^{(k-2)} + \varphi_1[\mathbf{x}_{l,p}^{(k-1)} - \mathbf{x}_p^{(k-1)}](rand) + \\ + \varphi_2[(\mathbf{x}_g^{(k-1)} - \mathbf{x}_p^{(k-1)}](rand)\}. \tag{10}$$

Komponente $x_{j,p}$ (j=1,2,...,n) vektora $\mathbf{x}_p$ i komponente $v_{j,p}$ (j=1,2,...,n) brzina $\mathbf{v}_p$ sračunavaju se prema izrazima (7) i (8), a zatim vrednosti funkcija cilja

$$f_p^{(k)} = f_p^{(k)}(\mathbf{x}_p^{(k)}), \quad p = 1,2,\ldots,n_p, \tag{11}$$

za svaki član (česticu) roja i u svakoj iteraciji k = 1,2,3,...

Posle toga se u svakoj iteraciji između ovih vrednosti određuje ona koja je minimalna

$$z^{(k)} = \min f(\mathbf{x}_p^{(k)}) = f(\mathbf{x}_g^{(k)}), \; k = 1,2,\ldots. \tag{12}$$

i vektor $\mathbf{x}_g^{(k)}$ koji odgovara toj iteraciji.

Vektor $\mathrm{x}_{l,p}^{(k)}$, kojem odgovara "nabolja" minimalna vrednost funkcije cilja za član (česticu) roja p u svim prethodnim iteracijama, uključujući i iteraciju k, sračunava se upoređivanjem funkcija cilja za taj član u dve susedne iteracije k-1 i k na sledeći način

$$\mathrm{x}_{l,p}^{(k)} = \mathrm{x}_{l,p}^{(k-1)} \ za\ f_p^{(k)} > f_p^{(k-1)},\ \mathrm{x}_{l,p}^{(k)} = \mathrm{x}_{l,p}^{(k)} \ za\ f_p^{(k)} < f_p^{(k-1)}, \tag{13}$$

$p = 1,2,\ldots,n_p,\ k = 2,3,4,\ldots$

U prvoj iteraciji $k = 1$ su $\mathrm{x}_{l,p}^{(1)} = \mathrm{x}_g^{(1)}$ i $\mathrm{v}_p^{(1)} = \mathbf{0},\ p = 1,2,\ldots,n_p.$ (14)

Postupak se ponavlja sve dok apsolutna vrednost razlike minimalnih vrednosti funkcije cila u dve uzastopne iteracije ne postane zanemarljiva

$$| z^{(k)} - z^{(k-1)} | \le \delta,$$

gde je δ neki unapred izabrani mali broj.

Pošto u zadatku optimizacije postoje ograničenja (2) i (3), to i ona moraju biti uzeta u obzir. Postoji nekoliko predloga u literaturi kako postupati sa ograničenjima. Jedan od načina je da se primeni *Metoda kaznenih funkcija* (*Penalty function method*) i da se zadatak sa ograničenjim prevede u zadatak bez ograničenja. Ovde se predlaže znatno jednostavniji postupak, koji se sastoji u sledećem

U svakoj iteraciji, pošto se odrede komponente vektora $\mathrm{x}_p^{(k)}$ i odgovarajuće funkcije cilja $f_p^{(k)}$, za svaki član (česticu) p roja proverava se da li dobijene komponente ovog vektora, koje prestavljaju tražene varijable, zadovoljavaju uslove ograničenja (2) i (3). Za onaj član roja p za koje uslovi ograničenja nisu uspunjeni stavlja se da je funkcija cilja neki veoma veliki broj. Na taj način ovaj položaj člana $\mathrm{x}_p^{(k)}$ ne može biti relevantan i zbog velike vrednosti njegove funkcije cilja on se praktično privremeno isključuje iz roja. On se može eventualno u kasnijim iteracijama ponovo vratiti ako zadovolji uslove ograničenja. Tačnost rešenja zavisi od dužine intervala u kojem se simuliraju vrednosti nepoznatih, kao i od broja simulacija. Pošto se ova procedura može realizovati samo uz primenu elektronskog računara koji sve računske operacije izvršava ogromnom brzinom, to nema problema da se uzimaju i veći intervali i veći broj simulacija. Ako je forma funkcije cilja takva da ima više lokalnih minimima, onda treba uzeti duže intervale unutar skupa ograničenja i veći broj n_p članova (čestica) roja.

Na osnovu ovde prikazanog algoritma autori ovog rada su napisali kompjuterski program u programskom sistemu MATLAB.

4. BROJČANI PRIMER

Autori su, koristeći pomenuti kompjuterski program, uradili više karakterističnih primera, između ostalih i primer optimizacije mrežnog plana, odnosno roka građenja sa funkcijom cilja koja predstavlja ukupne troškove i upoređivali sa rezultatima dobijenim primenom drugih metoda, pre svega genetskih algoritama. Dobijeni rezultati pokazuju da je ova metoda kraća, jednostavnija, lakša za programiranje i daje veću tačnost u mnogim slučajevima od drugih metoda, iako pomenute metode daju takođe rezultate velike tačnosti.

Radi ilustracije ovde će ukratko biti prikazan jedan zadatak optimizacije dimenzija poprečnog preseka armirno betonskog nosača sistema proste grede, koju je izvršio ranije Ž. Praščević primenom teorije Korusha-Kuhna-Tuckera (TKKT) u knjizi [4] i radu [5] primenom binarnog genetskog algoritma. Ovde se zbog ograničenog prostora daju konačne formulacije funkcije cilja koja predstavlja relativne ukupne troškove betona, armature i oplate za dužni metar grede i uslova ograničenja koji su proizašli iz odredaba naših Propisa za beton i armirani beton BAB 87, kao i uobičajenog postupka dimenzionisanja, koji su detaljno prikazani u pomenutim radovima.

Funkcija cilja je

$z = x_1x_2 + 11.3636x_1 + 22.7272x_2 + 64\,940.9x_3$

Uslovi ograničenja su:

$-0.81x_1x_2^2 + 13.0169x_1x_2 + 79\,403 \leq 0,$

$-19.272x_2x_3 + 5x_1x_2 + 30\,500 \leq 0,$

$-x_1 + 40 \leq 0,\ 40 \leq x_1 \leq 43,\ 50 \leq x_2 \leq 61,\ 30 \leq x_3 \leq 41.$

Ovde su: x_1 – širina poprečnog preseka (b), x_2 – debljina poprečnog preseka (d) i x_3 površina poprečnog preseka armature (A_a).

Računato je sa brojem članova (čestica) roja n_p=40, i koeficijentima φ_1=2.50, φ_2=2.50, uz primenu koeficijenta K=2.61803 prema obrascu (9). Sa ovim podacima primenom pomenutog kompjuterskog programa dobijeno je posle 8 iteracija rešenje x_1=40.00 cm, x_2=58.19 cm, x_3=37.58 cm^2, z=6544.72.

Ovi rezultati su identični sa tačnim rezultatima dobijenim primenom TKKT. Primenom genetskog algoritma u radu [5] dobijeno je x_1=40.00cm, x_2=58.98cm, x_3=37.25cm^2 z=6573.2, što znači da je ova cena veća za 0.4% od cene dobijene optimizacijom članova rojeva. U praksi bi se dobijene teorijske vrednosti za x_2 zaokružile na cele santimetre, a usvojena površina armature bi se podesila broju i površini poprečnog preseka izabranih profila armature. Cena dužnog metra nosača dobila bi se, prema [5] kada se vrednost funkcije cilja z pomnoži sa cenom kubnog metra betona.

LITERATURA

[1] J. Kennedy, R. C. Eberhart, "Particle Sworm Optimization", Proc. IEEE Int. Conf. on Neural Networks (Perth, Australia) IEEE Service Center, Piscataway, NJ, IV, 1995, p. 1942 – 1948

[2] R. C. Eberhart and Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimiyation", The 7th Annual Conf. on Evolutionary Programming , San Diego, USA, 1998

[3] M. Clerc, "Particle Swarm Optimization", ISTE, London, 2006, 210p.

[4] Ž. Praščević, "Operaciona istraživanja u građevinarstvu", Građevinski fakultet, Beograd, 1992, 270 s.

[5] Praščević, Ž., "Binarni genetski algoritmi", Izgradnja LVIII, br. 3-4, 2004, s. 55-69.