OPTIMIZACIJA POMOĆU ČLANOVA ROJEVA

PARTICLE SWARM OPTIMIZATION

UDK: 624.01.001.26:519.245.863=861
Originalni naučni rad

Prof. dr Živojin PRAŠČEVIĆ, dipl. građ. inž.
Doc. dr Nataša PRAŠČEVIĆ, dipl. građ. inž.

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. Postupak je ilustrovan sa dva primera, od kojih se jedan odnosi na optimizaciju armiranobetonskog nosača. Izvršeno je upoređenje rezultata dobijenih ovom metodom sa rezultatima dobijenim primenom gentskog algoritma i sa egzaktnim rešenjima na bazi teorije Koruša-Kuna-Takera. Razlike izmedju rešenja dobjenog ovom metodom u odnosu na tačno rešenje su zanemarljive, zbog čega je ovaj metod veoma preporučljiv za rešavanje mnogih praktičnih problema.

Ključne riječi: *nelinearna optimizacija, heurističke metode, inteligencija rojeva.*

SUMMARY

In this work is presented one newer method for optimization of continuous functions which belongs to the heuristic stochastic methods, ie. to the methods of so called swarm intelligence. This method is mathematically and algorithmically 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 the describd procedure. The method is ilustrated by two examples, where one concerns to the optimization of a reinforced concrete beam with minimum costs as a objective function. Results obtained by this method are compared with the results obtained by the genetic algorithm and with the exact solution according to the Korush-Kuhn-Tucker theory. The differences between results obtained by this and exaact method are negligible, so this method is very recomendable for solving big number of practical problems.

Key words: *nonlinear optimization, heuristic methods, swarm intelligence.*

1. UVOD I DEFINICIJE OSNOVNIH POJMOVA[1)]

Inteligencija rojeva (*Swarm intellignce*) je deo veštačke inteligencije koji je zasnovan na kolektivnom ponašanju decentralizovnih i samoorganizovanih sistema. Naziv su uveli Gerardo Beng i Jing Wang 1989. god. u vezi sa razvojem celularnih (ćelijskih) robota. Sistemi inteligencije rojeva sastavljeni su od populacije prostih članova (čestica), koji se nazivju i *agenti*, koji se nalaze u međusobnim lokalnim interakcijama kao i u interakciji sa svojim okruženjem. Čestice ili agenti u skupu ili roju se pridržavaju prostih pravila i ne postoji centralna kontrolna struktura koja bi njima upravljala. *Inteligentni agent* je sistem koji opaža ili shvata svoje okruženje i preduzima akcije koje maksimiziraju šansu njegovog uspeha. U primere Inteligencije rojeva uključije se ponašanje kolonije mrava, jata ptica ili riba, krda životinja, grupacija bakterija i sl.

Veštačka inteligencija (*Artificial intelligence*) obuhvata proučavanje i projektovanje veštačkih inteligentnih sistema koji mogu da opažaju i shvataju svoje okruženje i preduzimaju akcije za maksimizaciju šansi svog uspeha. Naziv je predložio John McCarthy 1956. god., koji je veštačku inteligenciju definisao kao nauku i tehniku projektovanja i proizvodnje *inteligentnih mašina*. Među svojstvima koja su istraživači očekivali od ovakvih mašina bila su rezonovanje, znanje, planiranje, učenje, percepcija, komunikacija, sposobnost kretanja i manipulacije objekata. U dosadašnjem naučnom i tehnološkom razvoju nisu ostvarene mašine koje imaju sva ova svojstva, a naročito saznajne i kreativne mogućnosti, apstraktno mišljenje, kreiranje i shvatanje ideja i učenje koje poseduje samo ljudska inteligencija. Veštačka inteligencija

Adresa autora: Građevinski fakultet, Beograd, Bulevar kralja Aleksandra 73
e-mail:zika@grf.bg.ac.rs
e-mail:natasa@grf.bg.ac.rs

[1)] Definicije pojedinih pojmova date su prema Internet enciklopediji „Wikipedia“, kao i prema izvornoj literaturi čiji je spisak na kraju ovog rada čini osnovu *kompjuterskih nauka* i *kompjuterske inteligencije* koja se odnosi na heurističke algoritme, rasplinute (fuzzy) sisteme, neuralne mreže i evoluciono računanje. Ona obuhvata tehnike inteligencije rojeva, fraktale i teoriju haosa.

U algoritme inteligencije rojeva spadaju:

- Optimizacija kolonije mrava (Ant Colony Optimizacion),
- Optimizacija čestica (članova) rojeva (Particle Swarm Optimization) i
- Stohastičko difuzno pretraživanje (Stochastic Difussion Search).

Iako metode inteligencije rojeva po svom nastanku spadaju u novije metode, o njima i njihovim primenama u svetskoj literaturi postoji veliki broj radova, monografija i drugih publikacija. Neke od tih publikacija date su u popisu referenci koje su korišćene za pripremu i pisanje ovog rada.

Optimizacija kolonije mrava je metaheuristička tehnika, koju je predložio M. Dorigo za rešavanje problema kombinatorne optimizacije. Ti problemi se mogu svesti na nalaženje najboljih puteva na grafu ili mreži. Metoda je inspirisana ponašanjem kolonije mrava u potrazi za hranom, pri čemu se svaki mrav kreće slučajno i kada nađe hranu vraća se u svoj mravinjak obeležavajući put u povratku specijalnom hemikalijom koju luči koja se naziva feromon. Sledeći mravi koji naiđu na taj put neće se više kretati slučajno, nego će koristiti taj put i ponovo ga obelžiti u povratku. Feromon brzo isparava i što se nekim putem kreće više mrava koji su pronašli hranu to će sadržaj feromona na tom putu biti veći i taj put će biti najpovoljniji. M. Dorigo je matematički obradio ovo ponašanje i detalnije je kasnije razvio u monografiji [7] i kasnijim svojim radovima i radovima svojih saradnika (na primer radu [8]). Metoda se koristi za rešavanje većeg broja problema kao što su određivanje optimalnih ruta kretanja vozila, problema „trgovačkog putnika“, optimalnog raspoređivanja poslova i resursa i drugih problema. S. S. Arum je u svojoj BSc tezi [2] odbranjenoj u Indijskom institutu za tehnologiju u Guwarati-u ovu metodu primenio za optimizaciju mrežnog plana sa ograničenim resursima. G. Ćirović i S. Mitrović su u radu [6] dali prikaz ove metode sa jednim primerom optimizacije mrežnih planova.

Stohastičko difuzno pretraživanje je tehnika koja se primenjuje za optimizaciju globalnih funkcija cilja koje su sastavljene od više parcijalnih funkcija. Polazi se od skupa agenata koji postavljaju određene hipoteze koje se odnose na te funkcije. Svaki agent bira proizvoljno funkciju cilja i proverava hipotezu u vezi ispunjenosti te funkcije cilja, a zatim vrši individualnu razmenu (difuziju informacija) sa ostalim agentima i na taj način se identifikuju najbolje hipoteze. Metodu je predložio J. M. Bishop 1989. god. Našla je primenu u pretraživanju tekstova, robotici, izboru sistema bežičnih veza i drugim primenama.

Metoda optimizacije članova (čestica) rojeva spada u evolucione kompjuterske metode zasnovane na populaciji za optimizaciju kontinualnih funkcija sa i bez ograničenja. Metodu su predložili socijalni psiholog James Kennedy (Džejms Kenedy) i elektroinženjer Russell C. Eberhart (Rasel C. Eberhart) 1995. god. [10]. 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 drgim 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 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 [10], i bile su predmet njihovog proučavanja.

Prilikom donošenja odluka u praksi i životu, donosilac 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 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. U naizgled haotičnim kretanjima i dešavanjima u prirodi postoji nešto što unosi određeni poredak, što doprinosi održavanju života i kvaziravnotežnih stanja. Pored urođenih sposobnosti za održavanje života svake jedinke u prirodi i instikata, ovde važnu ulogu igra sposobnost da se primaju i koriste informacije. Ovo je naročito izraženo kod ljudi, koji zbog svojih ogromnih saznajnih i kreativnih mogućnosti, u odnosu na ostala živa bića na našoj planeti, poseduju ogromnu stvaralačku moć i potencijale da utiču na ukupna kretanja i pojave u prirodi.

Postoje velike razlike u ponašanju ljudi i njihovih zajednica ili grupa i ponašanja rojeva ili jata drugih živih bića. Kako zaključuju J. Kennedy i R. C. Eberhart [10] u vezi sa primenom ove metode, dva čoveka mogu imati o nekom cilju koji žele da postignu identična shvatanja i uverenja i doneti identične odluke a da se pri tom fizički ne sudare, dok članovi jata ne mogu zauzeti isti položaj u prostoru a da se ne sudare. Drugim rečima, mišljenaja i odluke više ljudi mogu biti podudarna a da se oni ne sudaraju fizički, dok dva člana jata ili dva fizička tela ne mogu zauzimati isti deo prostora.

Ova metoda za rešavanje problema nelinearnog programiranja, odnosno optimizacije je, kako će se videti iz daljeg izlaganja, matematički i algoritamski je veoma jednostavna i daje u mnogim slučajevima rezultate visoke tačnosti. Zbog toga ima određene prednosti za rešavanje većeg broja problema u odnosu na neke druge složenije heurističke metode kao što su genetski algoritmi, evolutivno programirnje i dr., kao i u odnosu na klasične numeričke metode koje su zasnovane na teoriji matematičkog programiranja i uslovima optimalnosti Korusha-Kuhna-Tuckera. U ovoj metodi simuliraju se početna rešenja primenom Monte Carlo simulacija i ta rešenja poboljšavaju ovim simulacijama u sledećim iteracijama dok se ne dobije optimalno rešenje. Zbog toga ona spada u stohastičke i heurističke metode. Kako zaključuju J. Kennedy i R. Eberhart [10], ova metoda konceptualno se nalazi između genetskih algoritama i evolucionog programiranja i veoma je zavisna od stohastičkih procesa odnosno Monte Carlo simulacija, jer se simuliraju ne samo početna rešenja nego i promene rešenja u toku iterativnog procesa.

2. ZADATAK OPTIMIZACIJE – NELINEARNI PROGRAM

Treba odrediti vrednost funkcije cilja

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

sa uslovima ograničenja

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

Ovim uslovima ograničenja treba dodati još i uslove

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

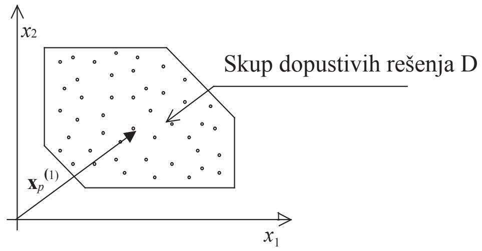
Vektor $\mathbf{x}$ je tačka u n-dimenzionalnom prostoru R^n a ograničenja (2) i (3) određuju skup dopustivih rešenja D.

3. ALGORITAM OPTIMIZACIJE

Procedura optimizacije izvršava se u više iteracija ili koraka. U prvoj iteraciji, koja se naziva *inicijalizacija*, u oblasti dopustivih rešenja D simuliraju se metodom Monte Carlo, kako je to prikazano na slici 1, vektori $\mathbf{x}_p^{(1)}$ (p=1,2,…,n_p), koji predstavljaju početne vektore položaja članova (čestica) roja u skupu D u kojem se kreću, čije su komponente

$$x_{j,p}^{(1)} = \min x_j + (\max x_j - \min x_j)(rnd),$$
$$j = 1,2,...,n; p = 1,2,...,n_p; \quad 0 \le (rnd) \le 1; \tag{4}$$

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



Slika 1 Simulirana početna rešenja

Za svaki od ovih vektora:

$$\mathbf{x}_p^{(1)} = [x_{1,p}^{(1)}, x_{2,p}^{(1)}, ..., x_{n,p}^{(1)}]$$

određuje se vrednost funkcije cilja

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

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

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

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

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

kako je to prikazano na slici 2.

Ovde je:

$$\mathbf{v}_p^{(k-1)} = [v_{1,p}^{(k-1)}, v_{2,p}^{(k-1)}, ..., v_{n,p}^{(k-1)}]$$

promena vektora položaja člana (čestice) roja p posle iteracije k-1. Ovaj vektor se naziva i *vektor brzine*, jer on predstavlja promenu vektora položaja člana (čestice) roja p u jedinici priraštaja vremena Δt = 1 i sračunava se prema obrascu

$$\mathbf{v}_p^{(k-1)} = \omega \mathbf{v}_p^{(k-2)} + \varphi_1[\mathbf{x}_{l,p}^{(k-1)} - \mathbf{x}_p^{(k-1)}](rnd) +$$
$$+ \varphi_2[(\mathbf{x}_g^{(k-1)} - \mathbf{x}_p^{(k-1)}](rnd); \quad 0 \le (rnd) \le 1. \tag{8}$$

$\mathbf{x}_{l,p}^{(k-1)}$ je najbolji položaj člana (čestice) roja p kojem odgovara najmanja vrednost funkcije cilja u prethodnim iteracijama 1,2,…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.

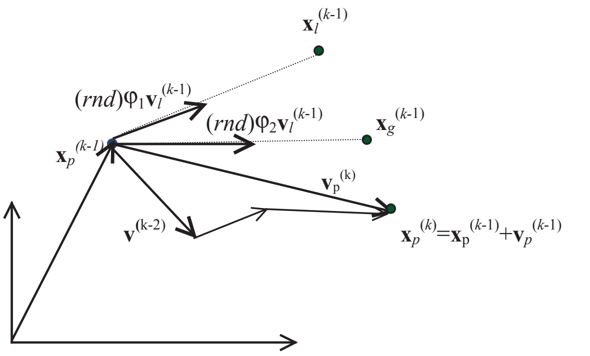
$\mathbf{x}_g^{(k-1)}$ je vektor položaja onog člana (čestice) roja za koji funkcija cilja u iteraciji k-1 ima najmanju vrednost $z^{(k-1)}$. Ovaj vektor naziva se i *socijalni* ili *socijalna komponenta*, kojom se uzima u obzir najbolji trenutni položaj pomenutog člana roja. Prilikom izbora ovog vektora mogu se uzeti susedni članovi roja po određenom pravilu ili svi članovi roja. U ovom radu su uzimani u obzir svi članovi roja.

ω je *faktor inercije* i uveli su ga Y. Shi i R. C. Eberhart u radu [9] i ima vrednost $\omega \le 1$. Često se uzima ω=0.9 i on utiče na smanjenje vektora promene (brzine) u narednim iteracijama. Ovi autori su preporučili da se računa sa vrednostima 0.7 ili 0.8.

φ_1 i φ_2 se nazivaju *faktori učenja* (*learning factors*) i njima se određuje relativni uticaj komponente saznanja o sopstvenom najboljem položaju člana (čestice) p roja $\mathbf{x}_{l,p}^{(k-1)}$ i tzv. socijalne komponente $\mathbf{x}_{g,p}^{(k-1)}$ na položaj člana (čestice) roja p u sledećoj iteraciji k. Često se ovi koeficijenti uzimaju da su:

$\varphi_1 \approx 2,\ \varphi_2 \approx 2.$

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



Slika 2 Određivanje novog vektora položaja $\mathbf{x}_p^{(k)}$ *čestice*

Da bi se sprečilo da neke čestice dobiju veliku brzinu i da se kreću suviše brzo u prostoru dopustivih rešenja, što može dovesti do divergencije procesa optimizacije, predlaže se u literaturi [11] da se ona limitira, tako da maksimalna apsolutna vrednost komponente brzine u iteraciji k bude manja ili jednaka nekoj unapred propisanoj maksimalnoj brzini $v_{\max}$, tj.

$$\max|v_{j,p}^{(k)}| \le v_{\max};\ p=1,2,...,n,\ \ k=2,3,... \tag{9}$$

U literaturi neki autori preporučuju da ona bude 10% od maksimalne vrednosti d_j

$$d_j = \max x_j - \min x_j;\ j=1,2,...,n.$$

M. Clerc (Klerk) [5] je, radi ograničavanja brzine i sprečavanja divergencije ili tzv. „eksplozije“ procesa, uveo *faktor suženja* (*constriction factor*) K koji se sračunava prema sledećem izrazu

$$K = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}\,|},\ \ \varphi=\varphi_1+\varphi_2>4. \tag{10}$$

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

$$\begin{aligned}\mathbf{v}_p^{(k-1)} &= K\{\mathbf{v}_p^{(k-2)}+\varphi_1[\mathbf{x}_{l,p}^{(k-1)}-\mathbf{x}_p^{(k-1)}](rnd)+\\ &+\varphi_2[(\mathbf{x}_{g,p}^{(k-1)}-\mathbf{x}_p^{(k-1)}](rnd)\}.\end{aligned} \tag{11}$$

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

$$f_p^{(k)} = f_p^{(k)}(\mathbf{x}_p^{(k)}),\ \ p=1,2,...,n_p, \tag{12}$$

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

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

$$z^{(k)} = \min f(\mathbf{x}_p^{(k)}) = f(\mathbf{x}_g^{(k)}),\ k=1,2,.... \tag{13}$$

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

Vektor $\mathbf{x}_{l,p}^{(k)}$, kojem odgovara „najbolja“ 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

$$\begin{aligned}&\mathbf{x}_{l,p}^{(k)} = \mathbf{x}_{l,p}^{(k-1)}\ \ za\ \ f_p^{(k)} > f_p^{(k-1)},\\ &\mathbf{x}_{l,p}^{(k)} = \mathbf{x}_{l,p}^{(k)}\ \ za\ \ f_p^{(k)} < f_p^{(k-1)},\end{aligned} \tag{14}$$

p = 1,2,…,n_p, k = 2,3,4,….

U prvoj iteraciji k = 1 su

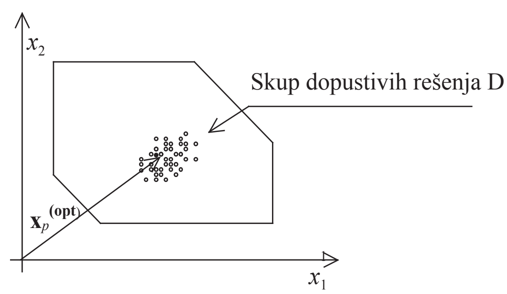
$$\mathbf{x}_{l,p}^{(1)} = \mathbf{x}_g^{(1)}\quad \text{i}\quad \mathbf{v}_p^{(1)} = \mathbf{0},\ p=1,2,...,n_p. \tag{15}$$

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

$$|z^{(k)} - z^{(k-1)}| \le \delta, \tag{16}$$

gde je δ neki unapred izabrani mali broj u zavisnosti od tačnosti rešenja koja se želi postići.

Položaj članova (čestica) roja koji odgovara optimalnom rešenju rešenju prikazan je na slici 3



Slika 3. Optimalno rešenje

Parametri φ_1 φ_2 prema izrazima (8) i (10) utiču na vektore brzina $\mathbf{v}_p^{(k)}$ odnosno promenu položaja člana (čestice) roja p, odnosno na njegovu trajektoriju kretanja u Euklidovom prostoru R^n. Kako konstatuju J. Kennedy i R. C. Eberhart u radu [11] ako su ovi parametri bliski nuli trajektorije kretanja teže glatkim krivim linijama jer su promene u iteracijama male. Trajektorije posle više iteracija se kreću ka najboljim pozicijama, odnosno rešenjima. Medjutim, ako funkcija cilja ima više lokalnih minimuma može se dogoditi da se brzo dostigne jedan od njih koji ne mora da bude globalni minimum sa najmanjom vrednošću u skupu dopustivih rešenja D koji treba odrediti u procesu optimizacije. Ako su ovi parametri veliki trajektorije osciluju oštrije i vektor $\mathbf{v}_p^{(k)}$ može dobiti velike vrednosti koje ne teže najboljoj poziciji, a može doći do nestabilnosti odnosno divergencije ili tzv. „eksplozije" rešenja. Problemi stabilnosti i konvergencije rešenja u multidimenzionalnom i kompleksnom prostoru su analirani u radu M. Clerka i J. Kennedy-a [4]. M. Clerk u knjizi [5] konstatuje da faktor inercije i maksimalne vrednosti ovih parametara nisu nezavisni i predlaže da se uzimaju parovi njihovih vrednosti $\omega = 0.7$ max φ_1 = max $\varphi_2 = 1.47$ i $\omega = 0.8$ max φ_1 = max $\varphi_2 = 1.62$.

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 k=1,2,3,…, pošto se odrede komponente vektora $\mathbf{x}_p^{(k)}$ i odgovarajuće funkcije cilja $f_p^{(k)}$, za svaki član (česticu) p roja proverava se da li dobijene komponente ovog vektora, koje predstavljaju tražene varijable, zadovoljavaju uslove ograničenja (2) i (3), odnosno da li se njegov vektor položaja $\mathbf{x}_p^{(k)}$ nalazi u skupu dopustivih rešenja. Za onaj član roja p za koje uslovi ograničenja nisu ispunjeni stavlja se da je funkcija cilja neki veliki broj. Na taj način ovaj položaj člana $\mathbf{x}_p^{(k)}$ 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.

Ako u skupu dopustivih rešenja problema D postoji više lokalnoh minimuma, onda se može dogoditi da trajektorije teže jednom od njih koji nije globalni minimum. U ovakvim slučajevima se preporučuje da se similura što veći broj članova roja i da se postupak ponavlja više puta, odnosno u više ciklusa i da se u svakom ciklusu odrediti lokalni minimum. Između ovih minimuma bira se onaj koji ima najmanju vrednost i koji predstavlja globalni minimum.

P. J. Angeline (Endželajn) je za ovakve slučajeve u radu [1] predložio proces selekcije članova roja. Na kraju svake iteracije k vrši se izmena (apdejtovanje) članova roja. Redosled članova roja se određuje prema vrednosti njihove funkcije cilja $f_p(k)$. Gornja polovina članova roja sa boljim vrednostima funkcije cilja se zadržava i duplira. Donja polovina članova roja sa lošijim vrednostima funkcije cilja se odbacuje i na njeno mesto dolaze duplirani članovi iz gornje polovine roja i tako proces nastavlja prema prethodno opisanom postupku dok se ne dobije rešenje koje ispunjava uslov (16) koje se može smatrati za optimalno rešenje. Autori ovog rada ukazuju na mogućnost da se donja polovina roja formira tako što bi se na parovima članova gornje polovine roja primenio operator *ukrštanja* (*crossover*) slično kao u genetskim algoritmima prilikom ukrštanja hromozoma, kako je to opisano u radu [13].

Tillett J. i saradnici su u radu [14] predložili strategiju koja je bazirana na prirodnoj selekciji u kojoj kada rešenje teži ka lokalnom optimumu pretraživanje u toj oblasti se napušta i pretražuje druga oblast. Ovu proceduru, koja je formulisana po uzoru na sekciju vrsta u prirodi, su nazvali *Darvinovska optimizacija članova rojeva* (*Darwinian Particle Swarm Optimization*). Simulira se veći broj rojeva koji zadovoljavaju uslove ograničenja. Za svaki roj se posebno primenjuje prikazani algoritam optimizacije i odredi član sa najpovoljnijom vrednošću funkcije cilja. Zatim se sračunavaju brzine i novi položaji članova roja i odrede vrednosti funkcija cilja. Utvrđuje se da li je novi položaj roja povoljniji od prethodnog u zavisnosti od vrednosti funkcija cilja. Ako se ponaša lošije briše njegov član sa najnepovoljnijom funkcijom cilja. Kada broj ovakvih članova u daljem iterativnom procesu bude manji od minimalnog dozvoljenog broja članova roja min n_p taj roj se dalje isključuje iz procedure. Roj čiji članovi imaju u posmatranoj iteraciji povoljnije položaje od položaja u prethonoj iteraciji stvara novi roj kao svoga „naslednika" tako što se polovina članova tog novog roja uzima po slučajnom izboru iz posmatranog roja, a druga polovina uključivanjem članova iz ostalih rojeva koji se nalaze u kolekciji aktivnih rojeva. Rojevi koji su se u ovom iterativnom procesu zajedno sa svojim „naslednicima" najduže održali imaju i najduži „životni vek". Proces se nastavlja dok se ne ispuni uslov za roj sa najdužim „životnim vekom", tako da komponente vektora položaja člana toga roja sa najmanjom vrednošću funkcije cilja predstavljaju globalno optimalno rešenje.

Na osnovu ovih prostih ideja, kako zaključuju autori rada [14] u kojem je ova procedura detaljno opisana, implementira se algoritam prirodne selekcije, jer u prirodi pojedinci ili grupe koji poseduju sposobnost povoljne adaptacije na uslove u kojima egzistiraju imaju više šansi da napreduju i da stvaraju.

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

4. BROJČANI PRIMERI

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 ovako dobijene rezultate 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, iako i pomenute metode daju takođe rezultate velike tačnosti.

Radi ilustracije ovde će ukratko biti prikazan dva primera.

Primer 1. Ovo je jednostavan problem kvdratnog programiranja koji služi za jasno prikazivanje početnog i konačnog optimalnog rešenja.

Odrediti min funkcije cilja

$$z = \min f(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 3)^2,$$

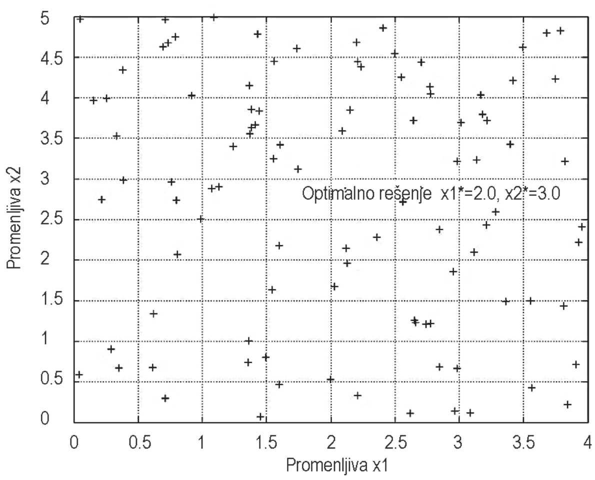
sa uslovima ograničenja

$$0 \le x_1 \le 3; \;\; 1 \le x_2 \le 4.$$

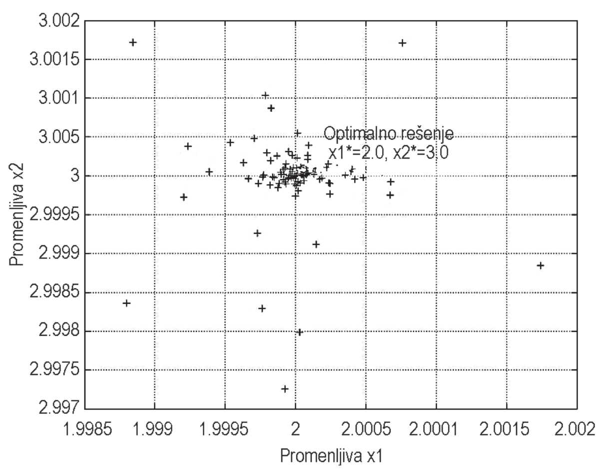
Izabran je broj članova roja $n_p = 100$, $\delta = 0.00001$.

Računato je sa sledećim parametrima $\omega = 0.9$, $\varphi_1 = \varphi_2 = 2.50$, K=0.382.

Primenom pomenutog računarskog programa simulirane su početne komponente vektora $\mathbf{x}_p^{(1)}$, koje predstavljaju polazne položaje članova roja, koje su prikazane na slici 4. Zatim su prema prikazanoj proceduri određeni položaji članova (čestica) roja u njihovom konačnom položaju koje odgovaraju optimalnom rešenju problema prikazanom na slici 5.



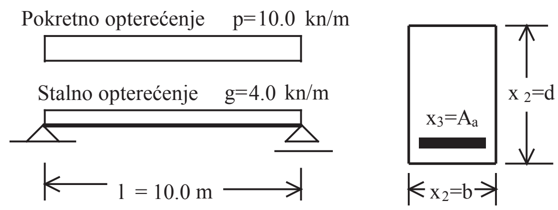
Slika 4. Početni simulirani položaji čestica roja (početna rešenja)



Slika 5. Konačan položaj članova roja (optimalno rešenje)

Optimalno rešenje ovog problema dobijeno ovim postupkom je x_1=2,00, x_2=3,00 i min $z = 0$. Ovo rešenje je identično tačnom rešenju koje se može sračunati metodom kvadratnog programiranja, odnosno primenom Lagranžovih (Lagrange) multiplikatora.

Primer 2. Ovaj primer se odnosi na određivanje optimalnih dimenzija poprečnog preseka armirnobetonskog nosača sistema proste grede, koju je izvršio ranije Ž. Praščević primenom teorije Korusha-Kuhna-Tuckera (TKKT) u knjizi [12] i primenom binarnog genetskog algoritma u radu [13], koji je publikovan u časopisu „Izgradnja". 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

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

Uslovi ograničenja su:

$$-0.81 x_1 x_2^2 + 13.0169 x_1 x_2 + 79\,403 \le 0,$$

$$-19.272 x_2 x_3 + 5 x_1 x_2 + 30\,500 \le 0,$$

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

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 $n_p = 100$, sa koeficijentima φ_1=2.50, φ_2=2.50, uz primenu koeficijenta $K = 0.382$ prema obrascu (10). 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 [13] dobijeno je $x_1 = 40.00$ cm, $x_2 = 58.98$ cm, $x_3 = 37.25$ cm^{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 prema broju i površini izabranih profila armature. Cena dužnog metra nosača dobila bi se, prema radu [13] kada se vrednost funkcije cilja z pomnoži sa cenom kubnog metra betona.

5. ZAKLJUČAK

Metoda optimizacije primenom članova rojeva spada u grupu metoda *inteligencije rojeva* ili u širem smislu u oblast *računarske inteligencije* odnosno *veštačke inteligencije*. Ona se nalazi, kako tvrde njeni autori J. Kennedy i R. Eberhart, koji su je predložili pre nešto više od jedne decenije, između genetskih algoritama i evolucionog programiranja. Algoritam ove metode se izvršava u iteracijama i primenjuju Monte Carlo simulacije ne samo prilikom simuliranja početnih rešenja, odnosno početnih položaja članova roja čije koordinate predstavljaju moguća rešenja problema, nego i promena njihovog položaja u toku ovog iterativnog procesa. Algoritam je veoma jednostavan i podesan za računarsko programiranje u poređenju sa drugim algoritmima i daje uz podesan izbor nekoliko parametara rezultate velike tačnosti. Zbog toga se ova metoda veoma može uspešno primeniti za rešavanje širokog spektra problema optimizacije.

LITERATURA

[1] Angeline, P. J., „Using Selection to Improve Particle Swarm Optimization“, in „*IEEE World Congress on Computational Inteligence*“, Anchorage, Alaska, USA, 1998.

[2] Arum, S. S., „*Construction Scheduling Using Nontraditional Optimization*“, BSc thesis¸Indian Instit. of Technology, Civ. Eng. Depart., 2005.

[3] Chan, F. T. S., and Tiwari, M. K., „*Swarm Intelligence – Focus on Ant and Particle Swarm Optimization*“, I-TECH Education and Publishing, Viena 2007, 532 p.

[4] Clerk, M. and Kennedy, J., „The Particle Swarm-Explosion, Stability and Convergence in Multidimensional and Complex Space“, *IEEE Transactions in Evolutionary Comuting*, Vol. 6, 2002

[5] Clerc, M.,“*Particle Swarm Optimization*“, ISTE, London, 2006, 210p.

[6] Ćirović, G., Mitrović, S.,“Evolucioni algoritmi u optimizaciji mrežnih planova“, *Izgradnja* LVI, br. 5–6/2007, s. 171-177.

[7] Dorigo, M.,“*Ant Colony Optimization*“, MIT Boston, 2004.

[8] Dorigo, M., „The Ant Colony Optimization Meta-Heuritc“, in „*New Ideas in Optimization*“ (Champter Two) edited by D. Corne, M. Dorigo and F. Glover, McGraw-Hill Comp., London, 2003., pp. 11-34.

[9] Eberhart, R. C. and Shi, Y. „Comparison between Genetic Algorithms and Particle Swarm Optimization“, *The 7th Annual Conf. On Evolutionary Programming* , San Diego, USA, 1998.

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

[11] Kennedy, J., Eberhart, R. C., „The Particle Swarm: Social Adaptation in Information-Processing Systems“, in „*New Ideas in Optimization*“ (Champter Twenty-Five) edited by D. Corne, M. Dorigo and F. Glover, McGraw-Hill Comp., London, 2003., pp. 379-389.

[12] Praščević, Ž., „*Operaciona istraživanja u građevinarstvu*“, Građevinski fakultet, Beograd, 1992, 270 s.

[13] Praščević, Ž., „*Binarni genetski algoritmi*“, Izgradnja LVIII, br. 3-4, 2004, s. 55-69.

[14] Tillett, J. Rao, T. M.,Sahin, F. and Rao, R., „Darwinian Particle Optimization“, *Proc. of Internat. Conference on Artificial Intelligence*, Pune, India, 2005, pp. 1474-1487.

[15] *Wikipedija. The Free Enciclopedia*. http://wikipedia.org