Primena algoritma sivog vuka za rešavanje inženjerskih optimizacionih problema

BRANISLAV N. MILENKOVIĆ, Matematički institut SANU, Beograd

MLADEN T. KRSTIĆ, Univerzitet u Kragujevcu,

Fakultet za mašinstvo i građevinarstvo, Kraljevo

DORDE M. JOVANOVIĆ, Matematički institut SANU, Beograd

Originalni naučni rad

UDC: 519.863:004.021

DOI: 10.5937/tehnika2101050M

U ovom radu će biti prikazana optimizacija sivog vuka (Grey Wolf Optimization-GWO). Nakon objašnjenja biološke osnove GWO algoritma, prelazi se na objašnjenje same metode, a zatim će biti reči o osnovnim algoritmima GWO metode, kao i njihovim matematičkim modelima. Potom će detaljno biti predstavljen algoritam sivog vuka, način njegovog rada i njegova primena na optimizacionim primerima inženjerskih problema kao što su: reduktor, sud pod pritiskom, opruga, bočni sudar automobila, konusno kvačilo i konzolna greda. Na kraju će biti prikazani rezultati dobijeni metodom GWO u odnosu na prethodno dobijene rezultate sa drugim metodama.

Ključne reči: optimizacija, sivi vuk, algoritam, inženjerski problemi

1. UVOD

U poslednje dve decenije pojavio se skup metoda koje su neočekivano efikasne pri rešavanju teških optimizacionih problema. Ove metode se nazivaju metaheuristički algoritmi, i često su inspirisane pojavama u prirodi, jer oponašaju neke dobre karakteristike iz prirode. Zato se ovi algoritmi još i nazivaju prirodom-inspirisani ili biološki metaheuristički algoritmi.

Porast popularnosti ovih algoritama privlači veliku pažnju inženjera i industrije. Jedan od razloga za ovu popularnost je što su ovi algoritmi prilagodljivi i efikasni, naizgled jednostavni s jedne strane, a s druge rešavaju veoma kompleksne optimizacione probleme. Metaheuristički algoritmi čine veoma važan deo savremenih globalnih optimizacionih algoritama, veštačke inteligencije i računarstva.

Mnogi metaheuristički algoritmi imaju osobinu da postižu konvergenciju ka globalnom optimumu problema nakon relativno malog broja iteracija. Najpoznatiji metaheuristički biološki inspirisani algoritmi su: diferencijalna evolucija (Differential evolution-DE), genetski algoritam (Genetic algorithm - GA),

algoritam slepog miša (Bat Algorithm-BA), algoritam sivog vuka (Grey Wolf Optimization-GWO), algoritam svica (Firefly Algorithm-FA), algoritam rojem čestica (Particle Swarm Optimization-PSA) i mnogi drugi

U nastavku ovog rada biće reči o algoritmu sivog vuka.

Algoritam sivog vuka pripada klasi biološki inspirisanih algoritama koje je kreirao Mirjalili [1]. Nakon njegovog objavljivanja 2014. godine, privukao je pažnju mnogim istraživačima koji su počeli njegovu implementaciju i usavršavanje u mnogim oblastima.

U radu Sahin et al. [2] razmatra se problem projektovanja opruge uzimajući minimalnu zapreminu kao kriterijum. Promenljive koje su razmatrane su prečnik opruge, prečnik žice i broj aktivnih namotaja. Funkcija cilja predstavlja zapreminu opruge u odnosu na pređe pomenute parametre. Algoritam sivog vuka je pokazao bolje rezultate od većine algoritama za taj problem.

U radu Jayakumar et al. [3] razmatra se problem kombinovanog snabdevanja toplotnom i električnom energijom sa kogeneracionim sistemima. Problem je formulisao na četiri načina: linearni sa jednom funkcijom cilja, nelinearni sa više funkcija cilja, nelinearni sa jednom funkcijom cilja, i problem sa ograničenjima rezervi u kojem se razmatra ekonomično snabdevanje. Za svaki od sistema koji se testira uzeto je razmatranje da se sastoji od 4, 7, 11 i 24 jedinice, koje mogu biti za distribuciju toplotne energije, električne

Adresa autora: Branislav Milenković, Matematički institut SANU, Beograd, Kneza Mihaila 36

e-mail: bmilenkovic@mi.sanu.ac.rs

Rad primljen: 04.12.2020.

Rad prihvaćen: 12.01.2021.

energije ili kogenerativne jedinice. Testirani sistemi su: kombinovano snabdevanje električnom i toplotnom energijom sa kogenerišućim izvorima, kombinovano snabdevanje električnom i toplotnom energijom sa uticajima sredine, ne-glatko ekonomično snabdevanje sa kogenerišućim izvorima i dinamično snabdevanje pri ograničenim rezervama sa prekidima u snabdevanju.

Testiranjem različitih veličina populacija je zaključeno da populacija od 30 jedinki daje najbolja rešenja. Što se tiče vremena izvršavanja, ovaj algoritam je pokazao kraće vreme izvršavanja od drugih metoda.

U radu Long et al. [4] koristi se poboljšani algoritam sivog vuka sa modifikovanim uvećanim Lagranžijanom za rad sa ograničenjima. Modifikovani uvećani Lagranžijan uklanja ograničenja tako što ih integriše u funkciju cilja. Algoritam vuka je modifikovan na taj način da se faktor istraživanja globalnog prostora smanjuje sporije od linearnog.

U ovom radu je odabran skup od 24 optimizaciona problema za testiranje algoritma. Napravljeno je poređenje standardnog i poboljšanog algoritma sivog vuka, pri čemu je zaključeno da poboljšani algoritam daje bolje rešenje za većinu problema. Takođe, za prvih 13 problema, napravljeno je poređenje poboljšanog algoritma sivog vuka sa drugim populacionim algoritmima. U većini slučajeva se dobije ili jednak ili bolji rezultat u odnosu na druge algoritme.

U ovom radu ćemo primeniti algoritam sivog vuka na rešavanje inženjerskih problema.

Prvi inženjerski problem je optimizacija jedno-stepenog reduktora kog koga je cilj minimizacija težine reduktora prema ograničenjima napona savijanja zuba zupčanika, površinski napon, transversalni ugib vratila i naponi u vratilu. Navedeni problem prvi je analizirao Coello primenom genetskog algoritma [5].

Drugi optimizacioni problem je optimizacija suda pod pritiskom. Cilj ovog problema je da se minimiziraju ukupni troškovi materijala, zavarivanja i oblikovanja. Navedeni problem prvi je predložio i rešio Sandgren [6].

Treći problem koji ćemo rešavati je optimizacija opruge. Problem opruge formulisao je opisao Arora [7], a sastoji se u smanjenju težine opruge koja podleže ograničenjima minimalnog ugiba, trenja, frekvencije ugiba, ograničenja spoljašnjeg prečnika i projektnih promenljivih.

Četvrti problem je optimizacija bočnog sudara automobila a cilj je minimiziranje ukupne težine vozila, korišćenjem zadatih jedanaest promenljivih. Ovaj problem je pri put predložen u radu Gu [8].

Peti problem je preuzet iz [9]. Cilj je naći minimalnu zapreminu konusnog kvačila kako bi se omogućilo prenošenje obrtnog momenta.

Poslednji inženjerski problem je optimizacija konzolne grede kod koga je potrebno naći minimalnu težinu tako da budu ispunjeni uslovi ograničenja. Gandomi je rešio ovaj problem primenom algoritma kukavičije pretrage (Cuckoo Search Algorithm-CSA) [10].

U ovom radu će rezultati algoritma sivog vuka (GWO) biti poređeni sa rezultatima dobijenim korišćenjem modifikovanog algoritma kolonije mrava (MACA – Modified Ant Colony Optimizaion), algoritma skakavca (GOA – Grasshopper Optimization Algorithm), algoritma vodenog ciklusa (WCA – Water Cycle Algorithm), optimizacije moljac-plamen (MFO – Moth-Flame Optimization), algoritma kita (WOA – Whale Optimization Algorithm), algoritma svica (FA – Firefly Algorithm), algoritma pretrage kukavice (CS – Cuckoo Search), optimizacije mravljeg lava (ALO – Ant Lion Optimization) i metode pokretnih asimptota (MMA – Method of Moving Asymptotes), zavisno od rešenja prisutnih u literaturi.

2. ALGORITAM SIVOG VUKA

Sivi vuk (lat. naziv *Canis lupus*) je predator, što znači da je na vrhu lanca ishrane. Živi u čoporima veličine od 5 do 12 članova, koji ima hijerarhiju koja se sastoji od alfa, beta, delta i omega članova. Slika socijalne hijerarhije u čoporu je data na slici 1. [11]

Slika 1 - Hijerarhija sivog vuka

Vođe čopora su alfa mužjak i alfa ženka. Alfa najčešće donosi odluke o lovu, spavanju, vremenu buđenja. Odluke alfe sluša ostatak čopora. Alfa nije nužno najjači član čopora, već onaj koji najbolje upravlja njime.

Nivo ispod alfe u hijerarhiji su beta vukovi. Bete pomažu alfama u donošenju odluka. U slučaju da alfa uginu, beta preuzima njegovu ulogu. Dakle, beta sluša komande alfe, ali komanduje ostalim vukovima. On predstavlja savetnika za alfu i zavodi disciplinu u ostatku čopora.

Najniži članovi hijerarhije su omega vukovi. Omega igraju ulogu žrtvenog jarca. Omega vukovi moraju da slušaju naređenja ostalih vukova, i poslednji su u redu za dobijanje hrane. Iako se čini da omega vukovi nisu važni u hijerarhiji, u čoporu mogu da nastanu unutrašnji sukobi ukoliko nema omega vuka. Ovo pomaže čoporu da održi strukturu dominacije. U nekim slučajevima omega čuvaju mladunce u čoporu.

Ukoliko vuk nije alfa, beta, ili omega, on pripada klasi delta vukova. Delta vukovi slušaju naređenja alfa i beta vukova, ali dominiraju nad omega vukovima. U ovu kategoriju spadaju izviđači, stari vukovi, lovci i negovatelji u čoporu.

Glavne faze u lovu sivog vuka su:

- faza približavanja
- faza okruživanja i
- faza napada

Na slici 2 je prikazan ponašanje sivih vukova pri lovljenju plena.

Slika 2 - Ponašanje sivih vukova tokom lova

Kako bi se matematički modelirala socijalna hijerarhija vukova, trenutno najbolje rešenje je alfa, dok su drugo i treće po redu beta i delta, respektivno. Ostatak rešenja predstavljaju omegi.

U fazi približavanja, vukovi se približavaju plenu, što se matematički modelira jednačinama:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

Gde je t trenutna iteracija, $\vec{A}$ i $\vec{C}$ vektori koeficijenta, $\vec{X}_p$ je pozicioni vektor plena, a $\vec{X}$ pozicioni vektor sivog vuka. Vektori $\vec{A}$ i $\vec{C}$ se računaju na sledeći način:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

Gde se komponente vektora $\vec{a}$ linearno smanjuju od 2 do 0 tokom iteracija, a r_1 i r_2 su slučajni vektori čiji elementi su u skupu $[0, 1]$. Svi vukovi se upravljaju prema poziciji plena, koja služi za promenu

sopstvene pozicije. Različita mesta oko najboljeg rešenja se postižu vrednostima vektora $\vec{A}$ i $\vec{C}$, dok r_1 i r_2 dozvoljavaju da se ta pozicija oko plena izabere nasumično. Vektor $\vec{a}$ služi da linearno smanji veličinu vektora $\vec{A}$ tokom iteracija. Vektor $\vec{A}$ unosi konvergenciju ka plenu, kada je veličina manja od jedan, odnosno divergenciju od plena, kada je veličina veća od jedan. Vektor $\vec{C}$ je nasumična komponenta koja unosi istraživanje prostora problema. Takođe se može smatrati i da modelira prepreke koje mogu se pojaviti na putu do plena.

Sivi vukovi imaju sposobnost da uoče lokaciju plena i okruže ga. Lov najčešće vodi alfa, dok povremeno beta i delta učestvuju u lovu. Ipak, u apstraktnom prostoru pretraživanja, lokacija optimuma, tj. plena, nije unapred poznata. Kako bi se matematički simuliralo ponašanje sivog vuka prilikom lova, pretpostavlja se da alfa beta i delta imaju bolji osećaj za lokaciju plena. Stoga se prva tri najbolja rešenja čuvaju, ostali članovi populacije menjaju svoje pozicije na osnovu trenutnog najboljeg rešenja. Za to se koriste sledeće formule:

$$\vec{D}_\alpha = |\vec{C} \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C} \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C} \cdot \vec{X}_\delta - \vec{X}| \quad (5)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \quad (6)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (7)$$

Ove jednačine prikazuju kako se menjaju pozicije alfa, beta i delta vukova u odnosu na poziciju lova. Pozicija lova je fiktivna, i onda zavisi od trenutnih pozicija alfa, beta i delta vukova, dok ostali vukovi se nasumično približavaju plenu.

Kada se sivi vukovi približe plenu, napadaju ga čim prestane da se kreće. Ovaj korak se matematički modelira tako što se vrednost vektora $\vec{a}$ linearno smanjuje sa iteracijama.

Algoritam je do sada opisivao kako se postiže konvergencija ka optimumu. Problem ovog pristupa je što se može zaći u lokalni minimum, bez da se istraži globalni minimum. Stoga, svaki populacioni metaheuristički algoritam mora da sadrži u sebi korak koji predstavlja istraživanje prostora rešenja.

Kod za algoritam sivog vuka je dat u nastavku.

Inicijalizacija populacije X_i ($i = 1, 2, \dots, n$)

Inicijalizacija a , A i C

Kalkulacija funkcije cilja za svakog člana populacije:

Inicijalizacija X_α , X_β , X_δ

While($t < \text{Maksimalni broj iteracija}$)

for član populacije in populacija

```

Izmeni poziciju agenta
End for
Izmeni a, A i C
Izračunaj funkciju cilja za sve agente
Izmeni  $X_\alpha$ ,  $X_\beta$ ,  $X_\delta$ 
 $t = t + 1$ 
end while
return  $X_\alpha$ 

```

U ovom algoritmu se za to koristi korak približavanja plenu, koji je u svojoj biti nasumičan. Modelira se vektorom $\vec{A}$, čiji članovi uzimaju nasumične vrednosti veće od 1 ili manje od -1. Drugi faktor koji favorizuje istraživanje prostora pretraživanja je vektor $\vec{C}$, koji sadrži vrednosti u opsegu od 0 do 2. Ova komponenta daje nasumične težinske vrednosti za plen, kako bi se naglasio uticaj plena u definisanju distance. Ova dva faktora utiču na nasumično ponašanje algoritma, favorizujući istraživanje i izbegavanje lokalnog minimuma.

Algoritam sivog vuka se može koristiti i za višekriterijumsku optimizaciju [20]. Prvo je potrebno rešenja podeliti na frontove, po definiciji iz NSGA-II algoritma, kako bi se ostvario poredak među njima. Kako bi se jednako istražio prostor rešenja, potrebno je podeliti ga na hiperblasti, i koristiti princip točka ruleta za izbor vođa čopora, imajući u vidu da bi trebalo da se favorizuju manje istraživane oblasti.

3. EKSPERIMENTALNA ISPITIVANJA

Zbog potrebe da se razvijene metaheurističke metode primene na inženjerske probleme optimizacije, neophodno je izvršiti verifikaciju rada metoda.

U ovom poglavlju izvršena su eksperimentalna ispitivanja za šest različitih, poznatih inženjerskih problema. Matematička formulacija funkcije cilja, ograničenja i opseg projektnih promenljivih za svih šest inženjerskih problema dat je u referenci [16].

Na slici 3 predstavljen je šematski prikaz prvog analiziranog inženjerskog problema.

Konstrukcioni parametri a ujedno i promenljive jednostepenog reduktora su:

- širina između vratila (x_1)
- modul zuba zupčanika (x_2)
- broj zuba manjeg zupčanika (x_3)
- dužina prvog vratila između ležajeva (x_4)
- dužina drugog vratila između ležajeva (x_5)
- prečnik prvog vratila (x_6)
- prečnik drugog vratila (x_7)

Treća promenljiva je celobrojna vrednost dok su ostale vrednosti neprekidne. Takođe, definisanu su četiri linearna i sedam nelinearnih ograničenja u

obliku nejednačina. Metode iz literature koje su korišćene za poređenje u ovom istraživanju su GOA, MACA i WCA.

Slika 3 - Jednostepeni reduktor

Detaljan prikaz rezultata dobijenih metodom GWO i poređenje nekoliko najboljih rezultata dobijenih korišćenjem drugih algoritma dati su u tabeli 1.

Tabela 1. Poređenje rezultata za optimizaciju reduktora

Promenljive vrednosti	MACA[14]	GOA[12]	WCA[13]	GWO
x_1	3.499	3.5	3.5	3.502
x_2	0.699	0.7	0.7	0.7
x_3	17	17	17	17
x_4	8.051	7.3	7.3	7.333
x_5	8.084	7.8	7.715	7.8
x_6	3.351	3.350	3.350	3.351
x_7	5.286	5.287	5.286	5.288
$f(x)$	3009.669	2996.964	2994.471	2998.299

Za ovaj optimizacioni problem je algoritam sivog vuka dao rešenje približno rešenjima iz literature. Iako je rešenje bolje od MACA algoritma, rešenja GOA i WCA algoritmi daju bolje rezultate od GWO.

Optimizacioni problem suda pod pritiskom (slika 4) analiziran je samo za kontinualne promenljive. Problem sadrži tri linearna i jedno nelinearno ograničenje u obliku nejednačine, kao i četiri kontinualne promenljive.

Slika 4 - Šematski prikaz suda pod pritiskom

Četiri promenljive koje je potrebno optimizovati su:

- debljina lima T_s (x_1)
- debljina lima na poklopcu T_h (x_2)
- unutrašnji poluprečnik R (x_3)
- dužina cilindričnog dela suda, ne uključujući poklopce L (x_4)

Očekivana optimalna vrednost funkcije cilja ovog problema je 5885.3327, a rezultati su predstavljeni u tabeli 2.

Tabela 2. Poređenje rezultata za optimizaciju suda pod pritiskom

Promenljive vrednosti	MACA[14]	GOA[12]	WCA[13]	GWO
x_1	0.822	0.8736	0.7781	0.779
x_2	0.406	0.4318	0.3846	0.385
x_3	42.602	45.2666	40.3196	40.383
x_4	170.484	199.9998	200	199.116
$f(x)$	5964.50	7666.1258	5888.3327	5888.23

Za ovaj problem GWO je dao rezultate bolje od svih rezultata pronađenih u literaturi.

Na slici 5 dat je šematski prikaz modela opruge, gde su i prikazane projektne promenljive koje figurišu u matematičkom modelu.

Slika 5 - Šematski prikaz opruge za optimizaciju

Ovaj problem formulisan je sa tri kontinualne promenljive, dva linearna i pet nelinearnih ograničenja u vidu nejednačina.

Projektne promenljive su:

- prečnik opruge D (x_1)
- prečnik žice d (x_2)
- broj aktivnih namotaja N (x_3)

Izvršena su eksperimentalna ispitivanja i u tabeli 3 predstavljene su dobijene vrednosti i vrednosti algoritama iz literature. Očekivana vrednost funkcije cilja ovog problema je 0.012665. Za ovaj problem rezultati svih prikazanih rešenja su jako slični. GWO algoritam je pokazao bolje rezultate od MACA i GOA algoritama, ali lošije od WCA algoritma.

Automobil, koji je prikazan na slici 6, izložen je bočnom sudaru na osnovu procedura Evropskog komiteta za poboljšanu bezbednost vozila. Promenljive pod rednim brojem 8 i 9 zauzimaju vrednosti iz skupa celih brojeva, dok se ostale vrednosti uzimaju iz skupa realnih brojeva. Potrebno je ispuniti i deset ograničenja, koja su data u formi jednakosti.

Tabela 3. Poređenje rezultata za optimizaciju opruge

Promenljive vrednosti	MACA[14]	GOA[12]	WCA[13]	GWO
x_1	0.0523	0.05	0.0516	0.05
x_2	0.3722	0.3174	0.3562	0.3173
x_3	10.4141	4.4584	11.3004	14.058
$f(x)$	0.0128	0.0128	0.0126	0.0127

Metode iz literature koje su korišćene za poređenje u ovom istraživanju su GOA, MFO i WOA, a sami rezultati su prikazani u tabeli 4.

Kao što je prethodno navedeno cilj je minimiziranje ukupne težine vozila koristeći devet uticajnih parametara koji uključuju: unutrašnju debljinu B – stuba (x_1), debljina potpore B – stuba (x_2), debljina unutrašnje ivice bočne strane poda (x_3), debljina prednjeg nosača za vešanje (x_4), debljina stuba vrata (x_5), debljina potpore za vrata u ravni pojasa (x_6), debljina krovnog nosača (x_7), debljina materijala za unutrašnju stranu B – stuba (x_8) i debljina materijala za unutrašnju ivicu bočne strane poda (x_9). U razmatranom modelu uvode se još dve promenljive visina barijere (x_{10}) i pozicija udara (x_{11}).

Slika 6 - Problem bočnog sudara automobila

Tabela 4. Poređenje rezultata za optimizaciju bočnog sudara automobila

Promenljive vrednosti	MFO[15]	GOA[12]	WOA[14]	GWO
x_1	0.5	0.5	0.5	0.5
x_2	1.116	1.115	1.108	1.14
x_3	0.5	0.5	0.534	0.5
x_4	1.301	1.303	1.305	1.268
x_5	0.5	0.5	0.5	0.5
x_6	1.5	1.5	1.473	1.5
x_7	0.5	0.5	0.5	0.5
x_8	0.345	0.345	0.345	0.345
x_9	0.345	0.286	0.192	0.192
x_{10}	-19.530	-19.715	-19.699	-20.605
x_{11}	0.0000	0.320	3.481	0.5
$f(x)$	22.842	22.8434	23.042	22.878

Za ovaj problem, GWO je dao bolje rezultate od WOA algoritma, ali lošije od MFO i GOA algoritama.

Peti inženjerski problem se sastoji u optimizaciji konusnog kvačila (slika 7). Ovaj problem ima za cilj minimizaciju zapremine konusnog kvačila, kako bi se prenosio minimalni obrtni moment.

Projektne promenljive koje se optimizuju su:

- spoljašnji radijus kvačila R_1 (x_1)
- unutrašnji radijus kvačila R_2 (x_2)

Metode koje su korišćene za poređenje u ovom istraživanju su GOA, FA i CS, a rezultati su prikazani u tabeli 5.

Slika 7 - Konusno kvačilo

Tabela 5. Poređenje rezultata za optimizaciju konusnog kvačila

Promenljive vrednosti	FA[16]	GOA[12]	CS[16]	GWO
x_1	4.298	4.286	4.285	4.286
x_2	2.140	2.143	2.142	2.142
$f(x)$	69.627	68.894	68.887	68.893

U ovom slučaju, GWO daje bolje rezultate od FA i CS algoritma, ali lošije od GOA algoritma.

I na kraju ćemo razmotriti popularni problem strukturalnog projektovanja konzolne grede kvadratnog preseka (slika 8). Cilj je optimizovati minimalnu težinu konzolne grede.

Slika 8 pokazuje da je konzolna greda napravljena korišćenjem pet, šupljih kutijastih nosača koji su kvadratnog preseka. Takođe, postoji samo jedno ograničenje za ovaj problem.

Slika 8 - Konzolna greda

Rezultati su prezentovani i upoređeni sa metodama ALO, MMA i GOA za verifikaciju u tabeli 6.

Projektne promenljive predstavljaju dužine stranice kvadrata pet kvadratnih preseka iz kojih se sastoji konzolna greda.

Tabela 6. Poređenje rezultata za optimizaciju konzolne grede

Promenljive vrednosti	ALO[18]	GOA[17]	MMA[19]	GWO
x_1	6.018	6.011	6.010	6.029
x_2	5.311	5.312	5.300	5.306
x_3	4.488	4.483	4.490	4.480
x_4	3.497	3.502	3.490	3.516
x_5	2.158	2.163	2.150	2.140
$f(x)$	1.339	1.339	1.340	1.340

Za problem optimizacije konzolne grede, GWO daje isti rezultat kao MMA algoritam, ali su ALO i GOA algoritmi dali bolji rezultat.

4. ZAKLJUČAK

Savremena optimizacija podrazumeva primenu modernih metoda optimizacije, a naročito u savremenom inženjerstvu, gde se na rešavanje optimizacionih problema gotovo uvek formiraju nelinearne funkcije cilja sa velikim brojem ograničenja i projektivnih promenljivih. Grupa najpopularnijih metoda koje se danas primenjuju na polju rešavanja optimizacionih rešenja su biološki inspirisani optimizacioni algoritmi.

Za rešavanje optimizacionog problema, neophodno je da se učine različiti koraci. Prvo, parametri problema trebaju da se identifikuju. Drugo, ograničenja koja su primenjena na parametre moraju biti prepoznata. Treće, ciljevi zadatog problema trebaju biti istraženi i razmotreni. I na kraju, na osnovu identifikovanih vrsta parametara, ograničenja i broja ciljeva treba izabrati pogodnu optimizacionu metodu za rešavanje konkretnih problema.

Ovaj rad opisuje primenu jednog metaheurističkog algoritma, algoritam sivog vuka GWO, na optimizaciju primera inženjerskih problema sa konstantnim brojem promenljivih. Primenom algoritma sivog vuka pri rešavanju prethodno navedenih šest inženjerskih problema korišćen je broj agenata $n=50$ i 1000 iteracija. Povećavanjem broja iteracija i agenata primećeno je da nisu dobijena bolja rešenja, te je ova kombinacija parametara odabrana jer ima najbrže vreme izvršavanja.

Korišćenjem GWO algoritma, dobijaju se optimalni ili približni rezultati, što je i prikazano primerima. U slučaju suda pod pritiskom dobijeni su bolji rezultati od metoda koje su navedene u tabeli 2. Dobijeni rezultati ostalih razmatranih inženjerskih problema (reduktor, opruga, bočni sudar automobila, konusno kvačilo i konzolna greda) su približni ali optimalni.

Pri rešavanju problema optimizacije, cilj je postići što bolji rezultat optimizacione funkcije u odnosu na postojeće rezultate u literaturi, te je stoga i mali napredak u odnosu na postojeća rešenja značajan. Ovaj rad se bavi primenom optimizacionih algoritama na inženjerske probleme, a prostor za dalja istraživanja u mašinstvu i drugim granama nauke je beskonačno veliki.

5. ZAHVALNICA

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja, Republike Srbije, kroz Matematički institut SANU.

LITERATURA

- [1] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software* 69, 46-61, 2014.
- [2] M. D. H. G. İsmail Şahin, Optimum Design of Compression Spring According to Minimum, *Gazi Journal of Engineering Sciences*, vol. 3, no. 2, pp. 21-27, 2017.
- [3] S. S. S. G. E. E. N. Jayakumar, Grey wolf optimization for combined heat and power dispatch with cogeneration systems, *Electrical Power and Energy Systems*, 2015.
- [4] X. L. S. C. J. J. W. Z. Wen Long, A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems, *Neural Computing & Applications*.
- [5] Carlos A. Coello Coello: Constraint-handling in genetic algorithms through the use of dominance based tournament selection. *Advanced Engineering Informatics*, 16, 193–203, 2002.
- [6] Sandgren E: Nonlinear integer and discrete programming in mechanical design optimization. *J Mech Des ASME*, 112(43):223-9, 1990.
- [7] Arora J. S, Introduction to optimum design. New York: *McGraw-Hill*; 1989.
- [8] Gu L, Yang R. J, Cho C. H, Cho C. H, Makowski M, Faruque M, Li Y, Optimization and robustness for crashworthiness, *International Journal Vehicle Des* 26(4):348-60, 2001.
- [9] R, V, J, Savsani, *Mechanical Design Optimization Using Advanced Optimization Techniques*, Springer-Verlag London, 2012.
- [10] A. Gandomi, X. S. Yang, A. H. Alavi, *Cucko search algorithm: a metaheuristic approach to solve structural optimization problems*, Springer-Verlag London, 2011.
- [11] M. Bošković, *Savremeni pristupi u kinematičkoj i dinamičkoj optimizaciji ravanskih mehanizama, doktorska disertacija*, Fakultet za mašinstvo i građevinarstvo u Kraljevu, Univerzitet u Kragujevcu, 2019.
- [12] Dj. Jovanović, B. Milenković, M. Krstić, Application of Grasshopper Algorithm in Mechanical Engineering, *YOURS*, pp.1-6, 2020.
- [13] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi: Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures* 2012.
- [14] V. Grković, R. Bulatović, Modified Ant Colony Algorithm for Solving Engineering Optimization Problems, *IMK-14-Research and Development* 18, 4, 2012.
- [15] Mirjalili S, *Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm*. Knowl Based Syst 89:228–249, 2015.
- [16] G. Miodragović, *Razvoj naprednih biološki inspirisanih algoritama za rešavanje optimizacionih problema primenjene mehanike*, Fakultet za mašinstvo i građevinarstvo u Kraljevu, Univerzitet u Kragujevcu, 2015.

- [17]Shahrzad S, Seyedali Mirjalili and Andrew Lewis, Grasshopper Optimisation Algorithm: Theory and Application, *Advances in Engineering Software*, Volume 105, pp 30-47, March 2017.
- [18]Mirjalili S, The Ant Lion Optimizer, *Adv Eng Software*,;83:80-98, 2015.
- [19]Chickermane H, Gea H. *Structural optimization using a new local approximation method*, Int J Number Methods Eng, 39:829-46, 1996.
- [20]Šošić D, Stefanov P, Višekriterijumska optimalna rekonfiguracija distributivne mreže primenom algoritma sivih vukova, *Infoteh-jahorina* Vol. 16, March 2017.
- [21]<https://seyedalimirjalili.com/gwo>
- [22]<https://kids.nationalgeographic.com/animals/mammals/gray-wolf/>

SUMMARY

APPLICATION OF GREY WOLF ALGORITHM FOR SOLVING ENGINEERING OPTIMIZATION PROBLEMS

This paper presents grey wolf optimization - GWO. After presenting the biological basis of GWO, it explains the method itself and then the main algorithms of the GWO method as well as their mathematical models. The Grey Wolf Algorithm (GWO) is presented in detail as well as the manner of its operation and its application to optimization examples of engineering problems, such as: optimization of speed reducer, pressure vessel, spring, car side impact, cone coupling and cantilever beam. At the end, the results obtained by the GWO method are compared to the results previously obtained by other methods.

Key words: Optimization, Grey wolf, Algorithm, Engineering problems