

NETOPIRJI, KRESNICE IN KUKAVICE PRI OBDELAVI SLIK

Iztok Fister Jr., Dušan Fister, Uroš Mlakar, Karin Ljubič, Janez
Brest, Iztok Fister

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

E-pošta: iztok.fister2@uni-mb.si

POVZETEK: Znanstveniki so vedno iskali vzor za reševanje najtežjih problemov, ki se pojavljajo na področju računalništva, matematike in industrije, v naravi. Narava ponuja različne smeri in poti za razvoj novih algoritmov. Dandanes se pojavlja veliko različnih algoritmov po vzorih iz narave, ki jih v splošnem lahko razdelimo v tri kategorije. Prva skupina algoritmov temelji na Darwinovi teoriji boja za obstanek, v drugo skupino spadajo algoritmi temelječi na obnašanju inteligence rojev ter drugih bioloških sistemov in zadnja skupina so algoritmi, ki delujejo po principih fizikalnih ali kemijskih procesov in celo po vzorih iz družboslovnih znanosti. Te vrste algoritmov so se v zadnjem času začele uporabljati tudi za reševanje problemov na področju obdelave slik. Ta članek obravnava najpomembnejše aplikacije algoritmov inteligence rojev novejšega datuma, kot npr. algoritmi na snovi obnašanja netopirjev, kresnic in kukavic, pri obdelavi slik.

1. UVOD

Računalniški strokovnjaki so se v zadnjih letih pri reševanju najtežjih problemov, s katerimi se soočajo vsakodnevno, vse bolj zatekali k vzorom iz narave. Večina naravnih procesov je v osnovi optimizacijskih, zato lahko modeliranje teh procesov na digitalnih računalnikih predstavlja osnovo za razvoj učinkovitih algoritmov za reševanje najtežjih problemov v realnem svetu. Z začetkom v 50 letih prejšnjega stoletja pa vse do danes je tako nastalo veliko algoritmov, ki temeljijo na vzorih iz narave.

Najpomembnejši predstavniki te vrste so prav gotovo evolucijski algoritmi [10], ki posnemajo Darwinovo teorijo boja za obstanek [11]. Po tej teoriji imajo največ možnosti za obstanek in razvoj tisti posamezniki, ki so najbolj prilagojeni pogojem v okolju. V devetdesetih letih prejšnjega stoletja so se tej družini algoritmov pridružili tudi algoritmi, ki temeljijo na kolektivnem obnašanju žuželk (npr. mravlje, čebele, termite, ipd.) in socialnih vrst živali (npr. jate ptičev, rib, delfinov, ipd.). Posebnost teh živalskih vrst je, da je njihova eksistenca odvisna od relativno enostavnih posameznikov, ki so sposobni avtonomnega izvajanja enostavnih akcij, vendar pa pri reševanju kompleksnejših opravil

znajo stopiti skupaj in se za izpolnitev skupnih ciljev samo-organizirati. Primere takih samo-organizacij lahko najdemo tudi v človeški družbi, kjer so se npr. v starem Egiptu, 4.000 let pred našim štetjem, organizirali sužnji pri gradnji veličastnih piramid. Ta decentralizacija in samo-organizacija socialnih vrst žuželk in živali sta postali osnovi za nastanek algoritmov na osnovi obnašanja inteligence rojev. Prvi je izraz inteligence rojev uporabil Beni leta 1989 [12]. Vzor za reševanje težkih problemov pa poleg biologije predstavljajo tudi druge naravoslovne znanosti, kot npr. kemija in fizika (npr. simulirano ohlajanje [16], črne luknje [15]), in nekatere družboslovne znanosti (npr. anarhična družba) ter celo glasba (npr. iskanje harmonij (*angl. Harmony Search*, krajše HS) [13]). Te tri skupine algoritmov so se uporabljale za reševanje velikega števila realnih problemov, kot so npr. reševanje: problema trgovskega potnika, barvanja grafov, generiranja šolskih urnikov, podatkovnega rudarjenja in ostalih.

Omenjene vrste algoritmov so zelo uporabne tudi pri obdelavi slik. Različni avtorji so aplicirali te algoritme na različna področja obdelave slik, kot so npr. izboljšava, kompresija, in segmentacija slik ter razpoznavanje objektov. Na teh področjih se srečujemo s specifičnimi problemi, kot npr. kako z odstranitvijo šuma oz. spreminjanjem kontrasta izboljšati sliko. Pri reševanju takih problemov imamo opravka z različnimi algoritmi obdelave slik, katerih največjo težavo predstavlja njihova velika časovna zahtevnost.

V tem članku se osredotočamo na pregled algoritmov na osnovi inteligence rojev (natančneje algoritmov na osnovi obnašanja netopirjev, kresnic in kukavic), ki so jih razvili na področju obdelave slik. V ta namen podrobneje analiziramo več člankov s tega področja. Na podlagi te analize poskušamo algoritme, opisane v njih, klasificirati v različne kategorije, glede na vrsto problemov, ki jih rešujejo. Na osnovi te klasifikacije poskušamo napovedati, v katero smer bo šel razvoj teh algoritmov v prihodnosti.

Struktura tega članka je v nadaljevanju naslednja: v naslednjem poglavju na kratko opišemo probleme, s katerimi se soočamo na področju obdelave slik. V tretjem poglavju izpostavimo značilnosti inteligence rojev, ter opišemo glavne značilnosti in principe delovanje algoritmov netopirjev, kresnic in kukavic. Četrto poglavje je osredotočeno na pregled aplikacij, ki jih lahko najdemo na področju obdelave slik. V zaključku povzamemo glavne značilnosti našega opravljenega dela in napovemo možne smeri prihodnjega razvoja tega naglo razvijajočega se področja.

2. OBDELAVA SLIK

Danes živimo v svetu, kjer se skoraj vsakodnevno srečujemo s signali, med katerimi so najbolj pogoste fotografije oz. videoposnetki. Razvitih je vedno več sistemov za analizo teh signalov, predvsem zaradi poenostavljanja, oz. avtomatizacije, nekaterih vsakdanjih opravil. Obdelava slik je področje, ki se ukvarja z analizo dvodimenzionalnih slik (2D) in se po navadi nanaša na digitalno obdelavo slik. Glavni cilj digitalne obdelave sliko je izluščiti pomembne informacije o sceni, ki je bila zajeta in jih na razumljiv način predstaviti človeku ali računalniku, ki te informacije uporablja za nadaljnje procesiranje.

Algoritmi, ki jih danes uporabljamo pri obdelavi slik, so razširjeni predvsem na področjih:

- računalniškega vida, kot npr.:
 - o industrijske proizvodnje,
 - o interakcije človek-računalnik (*angl. Human-Computer Interface*, krajše HCI).
- razpoznavanja obrazov,
- varne vožnje,
- zdravstva,
- računalniških iger,
- stiskanja slik,
- itd.

Računalniški vid je morda eno izmed najzanimivejših področij za raziskovalce, saj s pomočjo algoritmov računalniškega vida računalniki oz. stroji, ki ga uporabljajo, postajajo avtonomni sistemi. Zelo razširjeno aplikativno področje računalniškega vida je, t. i. strojni vid [25], kjer se zajete informacije iz slik uporabljajo za nadzor proizvodnih procesov, kot npr. nadzor kvalitete proizvodnje in izločanje nekakovostnih proizvodov, nadzor robotske roke, ipd. Računalniki so danes nepogrešljivo delovno orodje, zato postaja področje interakcije človeka z računalnikom vse pomembnejši predmet raziskav. Raziskovalci na tem področju skušajo razviti nove preproste načine za komunikacijo z računalnikom, kjer je v ospredje postavljeno intuitivno brez-kontaktno upravljanje. Primer take tehnologije je senzor Kinect [26], ki je sicer primarno namenjen za igranje računalniških iger, vendar je mogoče njegovo uporabo s pomočjo aplikacijskega vmesnika Microsoft relativno enostavno razširiti tudi na druga področja [27][28].

Zelo raziskano področje je tudi prepoznavanje obrazov, predvsem zaradi širokega aplikativnega področja. Danes je detektor obrazov vgrajen v skoraj vsako digitalno kamero in se uporablja za ostrenje zajete scene. Med drugim se za detekcijo obrazov zanimajo na področju marketinga, saj bi lahko s pomočjo vgrajene kamere razpoznali svoje kupce, njihovo starost in spol, ter na podlagi tega predvajali ustrezne oglase.

Novejše aplikativno področje obdelave slik je avtonomno vozilo oz. podporni sistemi za pomoč vozniku med vožnjo. Tukaj lahko uporabljamo senzorje za spremljanje koncentracije voznika pri vožnji, spremljanje nenadnih sprememb na cestišču, itd.

Eno izmed pomembnejših aplikativnih področij obdelave slik je zagotovo obdelava medicinskih slik. Zdravniki pogosto porabijo veliko časa za preglede različnih vrst medicinskih slik pacientov, zato bi jim postopki obdelave slik časovno precej skrajšali delo. Tukaj govorimo predvsem o procesiranju rentgenskih slik, angiogramov, ultrazvoka, ipd., za namene diagnosticiranja pacientov (npr. detekcije tumorjev, arterioskleroze, itd.). Na področju obdelave medicinskih slik se v literaturi pojavlja

veliko algoritmov [29], ki še ne delujejo popolnoma avtomatsko, saj od uporabnika zahtevajo vhodne podatke, njihove rezultate pa trenutno lahko uporabimo zgolj le kot pomoč pri postavljanju diagnoz.

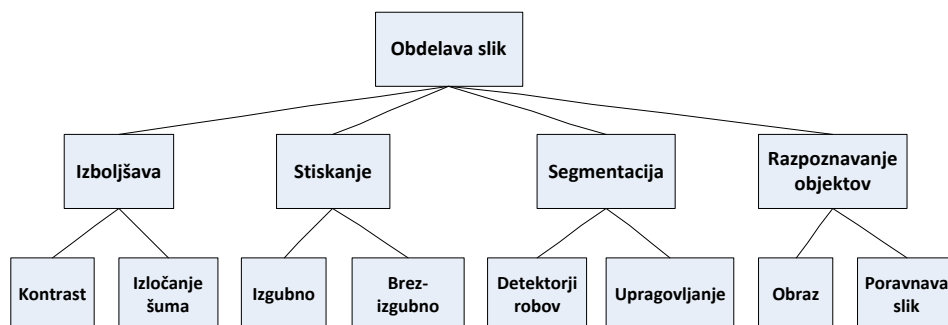
Z razvojem naprednih senzorjev za zajem slik v visoki ločljivosti se vzporedno pojavlja problem shranjevanja teh podatkov. S temi problemi se spopadajo raziskovalci na področju stiskanja slik. Algoritmi stiskanja slik v veliki meri iščejo redundanco v podatkih oz. skušajo surove podatke, zajete s senzorji, spraviti v kompaktno in predvsem manjšo obliko. Največja težava, s katero se srečujejo na tem področju je izguba informacij, ki jo vnašajo nekateri algoritmi stiskanja.

V nadaljevanju razdelimo domeno obdelave slik na problemska področja, te pa na specifične probleme, s katerimi se soočamo na problemskih področjih. Domeno obdelave slik delimo na naslednja problemska področja in specifične probleme (slika 1):

- izboljšava:
 - o odstranjevanje šuma: zajete slike po navadi vsebujejo odvečne informacije, ki nas ne zanimajo, oz. nad lahko motijo pri kasnejšem procesiranju slike, zato jih moramo odstraniti. Za odstranjevanje šuma je bilo razvitih že veliko postopkov, med katerimi največkrat uporabljamo filtriranje (npr. mediana, nizko sito, itd).
 - o izboljševanje kontrastov: zajeta slika je lahko slabe kvalitete, kar lahko popravimo npr. z izenačevanjem histogramov.
- stiskanje:
 - o izgubno: pri stiskanju podatkov se nekatere informacije o sliki izgubijo. Izgubno stiskanje je primerno za uporabo pri fotografijah zajetih s fotoaparatom.
 - o brez-izgubno: pri stiskanju podatkov ne izgubljam informacij v sliki. Brez-izgubno stiskanje uporabljamo večinoma za arhiviranje, stiskanje medicinskih in tehničnih risb, itd.
- segmentacija: na sliki izločimo objekte, ki nas zanimajo, od ozadja. V to področje spadajo:
 - o detektorji robov,
 - o upragovljanje (*angl. thresholding*).
- razpoznavanje objektov:
 - o razpoznavanje obrazov,
 - o poravnava slik.

Težava, s katero se srečujemo pri odstranjevanju šuma, je posredno poslabšanje kvalitete slike, predvsem zaradi napačne izbire filtrov oz. njegovih parametrov. Največja težava pri stiskanju slik je učinkovito zmanjšanje velikosti datoteke, ne da bi pri tem sliko poslabšali (brez-izgubno), oz. da slika ne izgubi preveč informacij (izgubno). Pri segmentaciji slik je glavna težava, da ne obstaja splošen algoritem, ki bi ga lahko uspešno aplicirali na poljubni sliki in bi pri tem dosegli dobre rezultate. Prav tako se pri

segmentaciji pojavlja problem osvetlitve scene na sliki, kar lahko rezultate segmentacije dodatno poslabša.



Slika 1: Področja in problemi, s katerimi se običajno srečujemo v domeni obdelave slik.

3. INTELIGENCA ROJEV

Vzor za nastanek algoritmov inteligence rojev (*angl. Swarm Intelligence*, krajše SI) prihaja iz biologije [17], kjer so socializirane žuželke (npr. kolonije termitov, mravelj, čebel, ipd.) in nekatere vrste socializiranih živali (npr. jate ptic, rib, ipd.) sposobne opravljanja kompleksnih opravil v skupini, čeprav jim je narava kot posameznikom namenila avtonomno izvajanje samo preprostih operacij. Značilnost teh bitij je, da sama brez pomoči drugih članov v skupini ne bi preživela. Tudi človek je družbeno bitje, ki je v skupini sposoben opravljanja zelo kompleksnih nalog (npr. poleta na Luno).

Socializacija je pojav, kjer posamezniki živijo skupaj v istih bivališčih in se med seboj sporazumevajo. Razlogov za tako obnašanje posameznikov je običajno več, najpogosteje pa so vzroki za socializirano vedenje nekaterih naravnih bitij predvsem naslednji: iskanje hrane, gradnja skupnih bivališč, in razmnoževanje.

Termiti so, npr., razmeroma enostavna bitja, ki so v interakciji z drugimi člani kolonije sposobna gradnje veličastnih bivališč (termitnjakov). Mravlje in čebele živijo skupaj zaradi iskanja hrane. Pri tem se mravlje sporazumevajo med seboj posredno, s pomočjo kemične snovi feromona, katerega količina določa najkrajšo pot do hrane. Čebele komunicirajo med seboj s pomočjo plesa z mahanjem (*angl. waggle dance*), s katerim izvidniki (*angl. scout*) vabijo ostale čebele delavke (*angl. employed bees*) na področja bogata s hrano (nektarjem). Netopirji se orientirajo v prostoru s pomočjo fizikalnega pojava eho-lociranja. Kresnice pri razmnoževanju uporabljajo fizikalni pojav svetilnosti, kjer se svetilnost zmanjšuje s kvadratom razdalje med dvema posameznikoma. Svetlobo oddajajo samo moški posamezniki, in jo lahko občudujemo v toplih poletnih nočeh. Na to svetlobo se odzivajo samice godne za parjenje. Kukavice izkazujejo parazitsko vedenje, pri čemer v gnezda ostalih ptičev podtikajo svoja jajca, za katera ti skrbijo, kakor da bi šlo za njihova. Ptice selivke se pred nastopom zime selijo v toplejše kraje.

Algoritmi SI izkoriščajo vse te pojave za svoje delovanje. Tabela 1 prikazuje omenjene vrste socializiranih žuželk in živali skupaj s pojavi, ki omogočajo interakcijo med njimi, vzroki za njihovo socializirano obnašanje, algoritmi SI, ki delujejo na teh principih, in avtorji teh algoritmov.

Tabela 1: Najpomembnejši algoritmi SI in njihovi vzori iz narave

| Vrsta | Interakcija | Social. vedenje | Algoritem SI | Vir |
|---------------|-----------------|------------------------|--------------|------|
| Termiti | Feromon | Iskanje hrane | TCO | [24] |
| Mravlje | Feromon | Iskanje hrane | ACO | [23] |
| Čebele | Ples z mahanjem | Iskanje hrane | ABC | [21] |
| Netopirji | Eho-lokacija | Orientacija v prostoru | BA | [20] |
| Kresnice | Svetilnost | Razmnoževanje | FA | [18] |
| Kukavice | Podtikanje jajc | Parazitsko vedenje | CS | [19] |
| Ptice selivke | Migracije | Kje živeti | PSO | [22] |

Iz tabele 1 razberemo, da so najpomembnejši algoritmi SI naslednji: optimizacija s kolonijami termitov (*angl. Termite Colony Optimization*, krajše TCO), optimizacija s kolonijami mravelj (*angl. Ant Colony Optimization*, krajše ACO), optimizacija s kolonijami umetnih čebel (*angl. Artificial Bee Colony*, krajše ABC), algoritem na osnovi obnašanja netopirjev (*angl. Bat Algorithm*, krajše BA), algoritem na osnovi obnašanja kresnic (*angl. Firefly Algorithm*, krajše FA), kukavičje iskanje (*angl. Cuckoo Search*, krajše CS) in optimizacija s roji delcev (*angl. Particle Swarm Optimization*, krajše PSO). Čeprav gre za različne algoritme, pa imajo vsi podobno strukturo prikazano v psevdokodu algoritma 1.

```

1: inicializiraj_populacijo_z_nakljucnimi_delci;
2: ocena = oceni_vsak_delec;
3: while !zakljucni_pogoj do
4:     premakni_delec_proti_najboljsemu_posamezniku;
5:     ocena+=oceni_vsak_delec;
6:     izberi_najboljse_posameznike_za_naslednjo_generacijo;
7: end while

```

Algoritem 1: Psevdo-kod algoritma SI

Algoritem SI na gornji sliki začenja z inicializacijo populacije (vrstica 1), ki sestoji iz posameznikov (delcev). Nato izračunamo funkcijo uspešnosti, da bi dobili oceno kakovosti vsakega delca (vrstica 2). Osrednji del algoritma predstavlja zanka **while ... end while** (vrstice 3-7), ki generira novi delec z operacijo premika (vrstica 4), ga oceni (vrstica 5) in izbere najboljše delce za naslednjo generacijo (vrstica 6). Zanko zaključimo z ustavitvenim pogojem '!zakljucni_pogoj' (vrstica 4), ki običajno konča optimizacijski proces po predvidenem številu izračunavanj funkcije uspešnosti.

Značilnost algoritma SI je, da je proces preiskovanja iskalnega prostora odvisen od najboljše rešitve v trenutni populaciji. Ta proces napreduje, dokler se ta najboljša rešitev izboljšuje, t.j. pri premikanju slabših rešitev proti najboljši v populaciji odkrivamo nove najboljše rešitve. Ko takih rešitev proces preiskovanja ne odkriva, ta ne napreduje več. V

tem primeru pravimo, da je prišlo do stagnacije oz. je proces preiskovanja padel v lokalni optimum.

V nadaljevanju članka se osredotočamo na algoritme na osnovi netopirjev, kresnic in kukavičjega iskanja.

2.1 Algoritem na osnovi obnašanja netopirjev

Algoritem na osnovi obnašanja netopirjev (*angl. Bat Algorithm*, krajše BA), katerega avtor je Xin-She Yang, je bil razvit leta 2010. Navdih za ta algoritem predstavljajo mikro-netopirji in njihov fizikalni pojav eho-lociranja, pri katerem netopirji med letenjem oddajajo zvočni signal, in poslušajo njegov odboj (eho), ki se odbija od ovir. Pri eho-lociranju so pomembni trije parametri: frekvenčno območje $Q_i \in [Q_{\min}, Q_{\max}]$, stopnja $r_i \in [0,1]$ in glasnost $A_i \in [A_0, A_{\min}]$ oddanega pulza. Frekvenca oddanega pulza je odvisna od velikosti žrtve, t.j. manjša kot je žrtev, višja je frekvenca. Stopnja pulza se povečuje, ko se netopir približuje svoji žrtvi. Glasnost je višja, ko so netopirji v lovu za žrtvami in nižja, ko se ti vračajo domov.

Obnašanje netopirjev pri eho-lociranju lahko modeliramo v optimizacijski algoritem, kjer to obnašanje zajamemo v funkciji uspešnosti optimizacijskega problema, ki ga rešujemo. Kakovost takega algoritma je seveda odvisna od odkrivanja novih rešitev v prostoru preiskovanja. Pri algoritmičnih SI generiramo nove rešitve s pomočjo operacije premika umetnih netopirjev po prostoru preiskovanja. Ta operacija izkorišča fizikalne zakone Dopplerjevega efekta, ki jih izrazimo z naslednjimi enačbami:

$$\begin{aligned} Q_i^{(t)} &= Q_{\min} + (Q_{\max} - Q_{\min})N(0,1), \\ v_i^{(t+1)} &= v_i^{(t)} + (x_i^{(t)} - best)Q_i^{(t)}, \\ x_i^{(t+1)} &= x_i^{(t)} + v_i^{(t+1)}, \end{aligned} \quad (1)$$

kjer $N(0,1)$ označuje generirano naključno število iz Gaussove distribucije s povprečno vrednostjo nič in standardnim odklonom ena, $\mathbf{v} \in \{v_i\}$ je hitrost premika, in $\mathbf{x}_i \in \{x_{ij}\}$ populacija umetnih netopirjev.

Med preiskovanjem lahko glasnost $A_i^{(t)}$ in stopnjo $r_i^{(t)}$ oddanega pulza spreminjamo. V naravi se glasnost netopirjev zmanjšuje in stopnja oddanih pulzov povečuje, ko netopirji najdejo svoje žrtve. To značilnost izrazimo v algoritmu BA z naslednjo enačbo:

$$A_i^{(t+1)} = \alpha A_i^{(t)}, \quad r_i^{(t)} = r_i^{(0)} [1 - \exp(-\gamma \varepsilon)], \quad (2)$$

kjer sta α in γ konstanti.

2.2 Algoritem na osnovi obnašanja kresnic

Kresnice oddajajo čudovito svetlobo, ki jo lahko občudujemo v jasnih poletnih nočeh. Produkt te svetlobe je plod zapletene biokemijske reakcije bioluminiscenca. Namen svetlikanja teh žuželk je dvojen, saj služi tako iskanju partnerjev, kakor tudi kot obrambni mehanizem, ki jih varuje pred plenilci. Obnašanja kresnic je navdihnilo avtorja Xin-She Yang [12][31][32], da je, leta 2010, razvil algoritem na osnovi obnašanja kresnic (*angl. Firefly Algorithm*, krajše FA).

Tudi v tem primeru je omenjeni pojav pri obnašanju kresnic zajet v funkciji uspešnosti. Operator premika umetnih kresnic po prostoru preiskovanja sloni na fizikalnem zakonu svetilnosti I , ki se zmanjšuje s kvadratom razdalje r^2 od svetlobnega vira. Z razdaljo od svetlobnega vira se povečuje tudi absorpcija svetlobe, ki povzroči, da svetloba postaja vse šibkejša. Svetilnost izrazimo z naslednjo enačbo:

$$I(r) = I_0 \exp^{-\gamma r^2}, \quad (3)$$

kjer I_0 označuje svetilnost svetlobnega vira in je γ fiksni absorpcijski koeficient svetlobe. Atraktivnost kresnic β je proporcionalna njihovi svetilnosti $I(r)$ in jo glede na enačbo (3) lahko izrazimo kot:

$$\beta = \beta_0 \exp^{-\gamma r^2}, \quad (4)$$

kjer je β_0 atraktivnost pri $r = 0$. Razdaljo med dvema kresnicama \mathbf{x}_i in \mathbf{x}_j izrazimo z Evklidsko razdaljo kot:

$$r_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}, \quad (5)$$

kjer n označuje velikost problema. Premikanje i -te kresnice privlači atraktivnejša kresnica j po naslednji enačbi:

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 \exp^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha \varepsilon_i, \quad (6)$$

kjer je ε_i naključno število iz Gaussove distribucije in $\alpha \in [0,1]$ naključno število iz uniformne distribucije.

2.3 Kukavičje iskanje

Kukavičje iskanje (*angl. Cuckoo Search*, krajše CS) je algoritem inteligence rojev, ki sta ga razvila, leta 2009, Xin-She Yang in Suash Deb [13][14][30]. Navdih za razvoj tega algoritma prihaja iz obnašanja kukavic v naravi. Kukavice so živali, ki izkazujejo parazitsko vedenje, saj samice odlagajo jajca v gnezda od ostalih ptic.

Avtorja sta definirala algoritem na naslednji način: vsako jajce v gnezdu predstavlja rešitev, medtem ko kukavičje jajce predstavlja novo rešitev. Te rešitve so po navadi

predstavljene kot vektorji realnih števil. Obstajajo pa tudi druge predstavitve, npr. binarna predstavitev. Namen kukavičjega iskanja je uporaba in raziskovanje novih boljših rešitev, z namenom zamenjave najslabših rešitev v gnezdih.

Ta algoritem poskuša uravnovežiti lokalno naključno iskanje, ki izkorišča pridobljeno znanje v populaciji, z globalnim naključnim iskanjem z namenom preiskovanja novih področij v preiskovalnem prostoru. Preklop med obema vrstama preiskovanja reguliramo s parametrom p_α . Lokalno naključno iskanje matematično izrazimo kot:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha s \otimes H(p_\alpha - \varepsilon) \otimes (\mathbf{x}_j^{(t)} - \mathbf{x}_k^{(t)}), \quad (7)$$

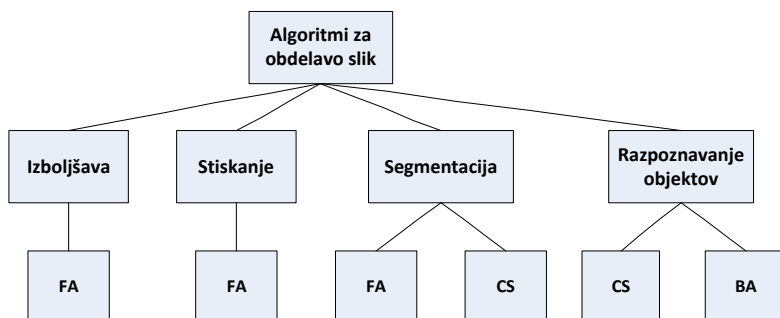
kjer sta $\mathbf{x}_j^{(t)}$ in $\mathbf{x}_k^{(t)}$ naključno izbrani rešitvi v populaciji, $H(u)$ je funkcija Heaviside, $\varepsilon \in [0,1]$ naključno število iz uniformne distribucije, in s je velikost koraka. Globalno naključno iskanje uporablja distribucijo Levi flights in generira novo rešitev po naslednji enačbi:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha L(s, \lambda), \quad (8)$$

kjer $L(s, \lambda)$ označuje naključno generirano število iz distribucije Levi flights.

4. APLIKACIJE

Namen tega članka je narediti podrobnejši pregled del, ki se ukvarjajo z aplikacijo algoritmov SI na področju obdelave slik. Pri tem so nas zanimali predvsem algoritmi novejšega datuma, t.j. BA, FA in CS. Čeprav je uporaba uveljavljenih algoritmov SI, kot npr. PSO, ABC in ACO, na tem področju zelo pogosta, ugotavljamo, da se novejši algoritmi SI tukaj še niso povsem uveljavili. Tukaj smo odkrili samo 9 člankov, ki rešujejo probleme obdelave slik z algoritmi BA, FA in CS. Po drugi strani pa so jih avtorji aplicirali že na večino problemov, ki nastopajo na tem področju (slika 2).



Slika 2: Implementacije algoritmov BA, FA in CS na področjih obdelave slik.

Iz slike 2 lahko ugotovimo, da z algoritmi BA, FA in CS rešujemo dandanes že praktično večino problemov, ki se pojavljajo na problemskih področjih domene obdelave slik. Največkrat se pri obdelavi slik pojavi algoritem FA, katerega so avtorji uporabili pri reševanju problemov iz področij izboljšave, stiskanja in segmentacije slik. Algoritem CS

je prisoten tudi pri segmentaciji slik in razpoznavanju objektov, medtem ko algoritem BA za enkrat uporabljamo samo pri razpoznavanju objektov.

Uporabo algoritmov BA, FA in CS na problemih obdelave slik prikazuje tabela 2.

Tabela 2: Uporaba algoritmov BA, FA in CS na problemih obdelave slik

| Problem | Avtorji | Alg. | Ref. |
|-----------------------------------|-----------------------|--------|----------|
| Upragovljene s pomočjo entropije | Horng in Liou | FA | [1] |
| Izgubno stiskanje slik | Horng in Jiang | FA | [2], [3] |
| Izboljšava kontrastov v sliki | Hassanzadeh in ostali | FA | [4] |
| Segmentacija slik - upragovljanje | Hassanzadeh in ostali | FA | [5] |
| Večnivojsko upragovljenje slik | Brajevic in ostali | FA, CS | [6], [7] |
| Razpoznavanje obrazov | Tiwari | CS | [8] |
| Poravnava slik | Zhang in Wang | BA | [9] |

Novejši algoritmi SI, t.j. BA, FA in CS, rešujejo večino problemov domene obdelave slik, ki smo jih navedli v sliki 1. Iz tabele 2 je razvidno, da največ problemov ponovno rešujemo z algoritmom FA. To dejstvo ne preseneča, saj je ta algoritem tudi najstarejši med opazovanimi. Po drugi strani pa je algoritem BA, najmlajši med njimi, uporabljen samo na problemu poravnave slik.

4. ZAKLJUČEK

V članku smo predstavili pregled uporabe novejših algoritmov SI, t.j. BA, FA in CS, na problemih iz domene obdelave slik. Podrobnejša analiza stanja tega področja je pokazala, da obstaja malo rešitev s to vrsto algoritmov, po drugi strani pa smo z njimi poskušali reševati že večino problemov iz te domene. Uporaba omenjenih algoritmov se je v praksi pokazala kot uspešna, rezultati obdelave slik pa so primerljivi s 'state-of-the-art' algoritmi iz te domene.

Glavni namen članka je seznaniti bralce z možnostmi uporabe teh novodobnih algoritmov v domeni obdelave slik in signalov. Po eni strani so ti algoritmi zelo enostavni za implementacijo, po drugi strani pa dovolj zmogljivi in hitri za uporabo v praksi. V prihodnosti lahko pričakujemo razvoj večjega števila aplikacij, ki bodo uporabljale omenjene algoritme po vzorih iz narave in se uspešno spopadale s problemi na področjih obdelave slik.

LITERATURA

1. M. Horng, R. Liou (2011), Multilevel minimum cross entropy threshold selection based on the firefly algorithm, *Expert Systems with Applications* vol. 38 (12), str. 14805–14811.

2. M. Horng, T. Jiang (2010), The codebook design of image vector quantization based on the firefly algorithm, *Computational Collective Intelligence Technologies and Applications*, str. 438–447.
3. M. Horng (2012), Vector quantization using the firefly algorithm for image compression, *Expert Systems with Applications*, vol. 39 (1), str. 1078–1091.
4. T. Hassanzadeh, H. Vojodi, F. Mahmoudi (2011), Non-linear grayscale image enhancement based on firefly algorithm, in: B.K. Panigrahi, P.N. Suganthan, S. Das, S.C. Satapathy (Eds.), *Swarm, Evolutionary, and Memetic Computing*, Springer, str. 174–181.
5. T. Hassanzadeh, H. Vojodi, A. Moghadam (2011), An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm, *The Seventh International Conference on Natural Computation*, vol. 3, IEEE, str. 1817–1821.
6. I. Brajevic, M. Tuba, N. Bacanin, (2012), Multilevel image thresholding selection based on the cuckoo search algorithm. In: *Proceedings of the 5th International Conference on Visualization, Imaging and Simulation (VIS'12)*, Sliema, Malta, str. 217–222.
7. I. Brajevic, M. Tuba (2014) , Cuckoo search and firefly algorithm applied to multilevel image thresholding, In *Studies in Computational Intelligence*, vol. 516, str. 115–139.
8. V. Tiwari (2012): Face recognition based on cuckoo search algorithm, *Indian Journal of Computer Science and Engineering*, vol. 3 (3), str. 401–405.
9. J. W.Zhang, G. G. Wang (2012), Image Matching Using a Bat Algorithm with Mutation, *Applied Mechanics and Materials*, vol. 203, str. 88–93.
10. A. Eiben, J. Smith (2003), Introduction to Evolutionary Computing. Berlin: Springer-Verlag.
11. C. Darwin (1859). On the Origin of Species. London: Harvard University Press (1964).
12. G. Beni, J. Wang (1989), Swarm Intelligence in Cellular Robotic Systems, *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, str. 26 – 30
13. Z. W. Geem (2012), Recent Advances in Harmony Search Algorithm. Berlin: Springer-Verlag.
14. I. Fister Jr., D. Fister, I. Fister (2013), A Comprehensive review of cuckoo search: variants and hybrids, *Int. J. Mathematical Modeling and Numerical Optimisation*, vol. 4 (4), str. 387–409.
15. A. Hatamlou (2013): Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, vol. 222, str. 175–184.
16. S. Kirkpatrick, C. J. Gellat, M. Vecchi (1983), Optimization by Simulated Annealing, *Science*, vol. 220(4578), str. 671–680.
17. C. Blum, D. Merkle (2008), Swarm Intelligence. Berlin: Springer-Verlag.

18. X.-S Yang (2008), Firefly Algorithm. V X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, London, UK: Luniver Press, str. 79–90.
19. X.-S. Yang, S. Deb (2009), Cuckoo Search via Levy Flights. *World Congress & Biologically Inspired Computing*, str. 210–214.
20. X.-S. Yang (2010), A New Metaheuristic Bat-inspired Algorithm. V C. Cruz, J. Gonzales, G. Krasnogor, & D. Pelta, Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Springer-Verlag: Berlin, str. 65–74.
21. D. Karaboga, B. Bastruk (2007), A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, *Journal of Global Optimization*, vol. 39(3), str. 459–471.
22. J. Kennedy, R. Eberhart (1999), The Particle Swarm Optimization; Social Adaptation in Information Processing. V D. Corne, M. Dorigo, & F. Glover, New Ideas in Optimization, London, UK: McGraw Hill, str. 379–387.
23. M. Dorigo, G. Di Caro (1999), The Ant Colony Optimization Meta-heuristic. V D. Corne, M. Dorigo, & F. Glover, New Ideas in Optimization, London, UK: McGraw Hill, str. 11–32.
24. R. Hedayatzadeh, F.A. Salmassi, M. Keshtgari, R. Akbari, K. Ziarati (2010), Termite colony optimization: A novel approach for optimizing continuous problems. *V 18th Iranian Conference on Electrical Engineering (ICEE)*, str. 553–558.
25. M. Sonka, V. Hlavac, R. Boyle (2008), Image Processing, Analysis, and Machine Vision, Second Edition.
26. Kinect. From Wikipedia, the free encyclopedia. Pridobljeno s <http://en.wikipedia.org/wiki/Kinect>, dne 1.2.2014.
27. K. Khoshelham, S. O. Elberink (2012), Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, vol. 12, str. 1437–1454.
28. Z. Ren, J. Meng, J. Yuan (2011), Depth Camera Based Hand Gesture Recognition and its Applications in Human-Computer-Interaction, *Proc. of 8th International Conference on Information, Communication and Signal Processing*, str. 1–5.
29. D. L. Pham, C. Y. Xu, J. L. Prince (2000), Current methods in medical image segmentation, *Annual review of biomedical engineering*, vol. 2, str. 315–337.
30. I. Fister Jr., X.-S. Yang, D. Fister, I. Fister (2014), Cuckoo search: A brief literature review, V Cuckoo search and firefly algorithm, str. 49–62.
31. I. Fister, X.-S. Yang, D. Fister, I. Fister Jr. (2014), Firefly algorithm: A brief Review of the expanding literature, V Cuckoo search and firefly algorithm, str. 347–360.
32. I. Fister, I. Fister Jr., X.-S. Yang, J. Brest (2013), A comprehensive review of firefly algorithms, *Swarm and Evolutionary Computation*, vol. 13, str. 34–46.