Algoritam diferencijalne evolucije za problem grupiranja podataka

Marjanović, Mihael

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

ALGORITAM DIFERENCIJALNE EVOLUCIJE ZA PROBLEM GRUPIRANJA PODATAKA

Diplomski rad

Mihael Marjanović

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 04.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Mihael Marjanović
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1226R, 07.10.2021.
OIB studenta:	29935492870
Mentor:	doc. dr. sc. Dražen Bajer
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Bruno Zorić
Član Povjerenstva 1:	doc. dr. sc. Dražen Bajer
Član Povjerenstva 2:	dr. sc. Mario Dudjak
Naslov diplomskog rada:	Algoritam diferencijalne evolucije za problem grupiranja podataka
Znanstvena grana diplomskog rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Opisati problem čvrstog grupiranja podataka. Opisati algoritam diferencijalne evolucije kao vrstu evolucijskih algoritama te njegovu primjenu za rješavanja problema grupiranja podataka. Ugraditi barem dvije inačice algoritma diferencijalne evolucije koje se razlikuju u korištenom operatoru mutacije za potrebe eksperimentalne analize. Na nekoliko sintetičkih i standardnih skupova podataka ispitati učinkovitost ugrađenih inačica algoritma.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	04.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 28.09.2023.

Ime i prezime studenta:	Mihael Marjanović
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1226R, 07.10.2021.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Algoritam diferencijalne evolucije za problem grupiranja podataka**

izrađen pod vodstvom mentora doc. dr. sc. Dražen Bajer

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD 1
2. GRUPIRANJE PODATAKA I ALGORITAM DIFERENCIJALNE EVOLUCIJE 3
2.1. Kratak uvod u grupiranje podataka 3
2.1.1. Algoritam k-means 4
2.1.2. Vrednovanje učinkovitosti 6
2.2. Algoritam diferencijalne evolucije 7
2.2.1. Primjena za grupiranje podataka 10
2.3. Pregled postupaka za grupiranje podataka 11
3. OSTVARENO PROGRAMSKO RJEŠENJE 13
3.1. Način rada programskog rješenja 13
3.2. Prikaz i način uporabe programskog rješenja 15
4. EKSPERIMENTALNA ANALIZA 16
4.1 Postavke eksperimenta 16
4.2 Rezultati 17
5. ZAKLJUČAK 26
LITERATURA
SAŽETAK
ŽIVOTOPIS
PRILOZI

1. UVOD

U današnjem digitalnom dobu, generiranje, prikupljanje i pohranjivanje ogromnih količina podataka je postalo uobičajeno i sveprisutno. Kako bi se ta ogromna količina podataka mogla iskoristiti na pravi način, potrebno je analizirati te podatke. Jedan od načina kako se analiziraju podatci je grupiranje podataka (engl. *data clustering*). Grupiranje podataka je proces razdvajanja danog skupa podataka u različite grupe. Svaki podatak se dodjeljuje određenoj grupi na temelju nekih značajki. Glavni cilj grupiranja je da podatci unutar iste grupe budu slični jedni drugima, a istovremeno budu različiti od podataka u drugim grupama. Grupiranje podataka ima širok spektar primjena u različitim disciplinama kao što su strojno učenje, marketing, medicina i mnoge druge. Grupiranje podataka se dijeli na čvrsto (engl. *hard*) i neizrazito (engl. *fuzzy*). Čvrsto grupiranje je kada podatak pripada samo jednoj grupi, dok u neizrazitom jedan podatak može pripadati u više grupa. Ovaj rad se bavi čvrstim grupiranjem podataka. Grupiranje podataka je NP-težak problem što znači da je vrlo zahtjevan za rješavanje. Jedan od najpopularnijih algoritama za čvrsto grupiranje podataka je algoritam k-means. Algoritam k-means je jednostavan za implementaciju, no ima nedostataka zbog kojih se nekad preferiraju drugi algoritmi za grupiranje. Neki od tih nedostataka su zaglavljivanje u lokalnim optimumima, osjetljivost na ekstreme u skupu podataka i to što se mora postaviti broj grupa prije pokretanja algoritma. Jedan od algoritama koji se može primijeniti za čvrsto grupiranje podataka s ciljem izbjegavanja navedenih nedostataka algoritma k-means je i algoritam diferencijalne evolucije (engl. *differential evolution*, DE). Algoritam DE je popularna inačica evolucijskih algoritama (engl. *evolutionary algorithms*, EA) koja se posebno istakla u numeričkoj optimizaciji. Iako je algoritam DE namijenjen numeričkoj optimizaciji, uz neke prilagodbe može se iskoristiti za grupiranje podataka. Jedan od razloga zašto se koristi algoritam DE za grupiranje podataka je taj što može efikasno i opsežno pretražiti prostor pretrage te tako izbjeći zaglavljivanje u lokalnim optimumima. Jedan od dijelova algoritma DE je operator mutacije, on je ključan za ponašanje i učinkovitost algoritma. Postoji više različitih operatora mutacije te odabir koji će se koristiti treba napraviti prema problemu koji se rješava.

Drugo poglavlje sadrži detaljniji opis postupka grupiranja i algoritma DE. Također je opisan algoritam k-means i način na koji se algoritam DE može prilagoditi kako bi se primijenio za grupiranje. Dan je pregled nekih radova koji se odnose na grupiranje podataka zasnovanih na EA. U trećem poglavlju je opisano ostvareno programsko rješenje, uz detaljan opis njegova načina rada i prikazom dijagrama toka rješenja. Uz opis, prikazan je i način uporabe programskog rješenja. Četvrto poglavlje opisuje provedenu eksperimentalnu analizu. U prvom djelu je opisan

eksperiment i korištene postavke u eksperimentu, a u drugom djelu su prikazani i komentirani rezultati provedenog eksperimenta. Eksperimentalna analiza prikazuje učinkovitosti obje inačice algoritma DE i algoritma k-means te ih uspoređuje.

2. GRUPIRANJE PODATAKA I ALGORITAM DIFERENCIJALNE EVOLUCIJE

U današnje vrijeme, zbog moderne tehnologije na svakom koraku, svaki dan se generiraju velike količine podataka. Ti podatci se mogu podvrgnuti raznim analizama. Jedan vrlo važan način kako se rukuje ovim podatcima je taj da ih se grupira ili klasificira u nekakve kategorije ili grupe. Grupiranje podataka je određivanje kojoj grupi neki podatak pripada na temelju određenih značajki pomoću kojih se uspoređuje koji su podatci slični jedni drugima, a koji nisu. Grupiranje se primjenjuje u mnogobrojnim područjima, neki od kojih su medicina, ekonomija, računalne znanosti [1].

Postoje različiti algoritmi koji se koriste za grupiranje podataka i svi oni imaju svoje razumijevanje kako to učiniti. Jedni od algoritama koji se mogu koristiti za grupiranje podataka su i EA. EA su algoritmi koji su zasnovani na načelu „opstanka najjačih". Jedan od predstavnika EA je i algoritam DE, njega su 1995. godine objavili Storn i Price [2]. Algoritam DE se može primijeniti za grupiranje podataka na način da se od inicijalne populacije rješenja, koja kod grupiranja podataka predstavlja centroide grupa, iterativno dođe do što boljeg rješenja, to jest što boljih centroida grupa nekog skupa podataka.

2.1. Kratak uvod u grupiranje podataka

Čvrsto grupiranje podataka je kada podatak može pripadati jednoj i samo jednoj grupi, i na takvo grupiranje podataka se ovaj rad odnosi. Podatak koji se grupira je najčešće prikazan kao višedimenzionalni vektor, gdje svaka dimenzija predstavlja jednu značajku podatka. Ove značajke mogu biti i kvalitativne i kvantitativne [3]. Grupiranje podataka se može opisati na sljedeći način. Ako je X skup podataka, kao prema (2-1),

$$X = \{x_1, x_2, \dots, x_N\}, \tag{2-1}$$

gdje su $x_1, x_2, \dots, x_N$ podatci unutar skupa X, N je broj podataka unutar skupa. Kao što je rečeno, podatak je prikazan kao višedimenzionalni vektor, kao prema (2-2).

$$x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,d}\}, \tag{2-2}$$

gdje je x_i i-ti podatak skupa X, a d je broj dimenzije podatka x_i. Ako postoji k grupa, definira se grupiranje kao k-grupiranje (k-particija) skupa podataka X, kao prema (2-3),

$$G = \{g_1, \dots, g_k\}, \tag{2-3}$$

gdje je G particija, a $g_1, \ldots, g_k$ su grupe unutar te particije. Podatci se grupiraju tako da su sljedeća tri uvjeta prema (2-4) ispunjena,

$$g_i \neq \emptyset \;\; i = 1, \ldots, k, \quad \bigcup_{i=1}^{k} g_i = X, \quad g_i \cap g_j = \emptyset, \; i \neq j, i, j = 1, \ldots, k, \tag{2-4}$$

gdje prvi uvjet znači da nijedna grupa ne smije biti prazna, drugi uvjet znači da unija svih grupa mora biti jednaka skupu podataka X, a treći uvjet znači da presjek različitih grupa mora biti prazan skup jer se radi čvrsto grupiranje. Cilj grupiranja je da su vektori koji se nalaze u grupi g_i više slični jedni drugima i istovremeno manje slični vektorima drugih grupa [4].

Za grupiranje podataka mora se odrediti mjera sličnosti koja se koristi za uspoređivanje podataka. Pomoću te mjere sličnosti se određuje kojoj grupi pripada koji podatak. Zbog različitosti vrsti značajki podataka koji se grupiraju, mjera sličnosti mora biti pažljivo odabrana. Kriterij po kojem se najčešće grupiraju podatci je udaljenost, gdje podatci koji su blizu pripadaju istoj grupi. Najpopularnija metrika za računanje udaljenosti koja se koristi za realne značajke je Euklidska udaljenost, udaljenost između dviju točaka u Euklidskom prostoru. Prema [3], Euklidska udaljenost se opisuje formulom (2-5).

$$d_2(x_i, x_j) = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2\right)^{1/2} = \left\| x_i - x_j \right\|_2. \tag{2-5}$$

Euklidska udaljenost je popularna jer se obično koristi za određivanje blizine objekata u dvodimenzionalnom ili trodimenzionalnom prostoru. Radi dobro kada skup podataka ima izolirane grupe, a nedostatak je sklonost da najveće skalirana značajka dominira ostale [3]. Moguće je koristiti i druge mjere sličnosti ili različitosti kao što su Mahalanobisova udaljenost, Hammingova udaljenost, Pearsonov koeficijent korelacije.

2.1.1. Algoritam k-means

Algoritam k-means je, prema [4], jedan od najpoznatijih algoritama za grupiranje podataka. Ovaj algoritam grupira N podataka u k grupa, gdje k mora biti unaprijed poznat. Svaki podatak iz skupa se pridruži najbližem centroidu koji predstavlja jednu grupu. Kako bi se izračunala udaljenost između podatka i centroida, algoritam k-means koristi ranije spomenutu Euklidsku udaljenost. Nakon što se dodjele svi podatci centroidima, onda se radi ažuriranje centroida. To znači da se računa aritmetička sredina svih podataka u pojedinim grupama, i ta aritmetička sredina postaje novi centroid za tu grupu. Ovo je iterativni algoritam koji započinje s proizvoljnim početnim centroidima i svakom iteracijom pokušava pronaći što bolje rješenje, odnosno grupiranje. Funkcija koju optimira algoritam k-means je zbroj kvadratnih udaljenosti svih podataka do njihovih centroida (engl. *sum of squared errors,* SSE), to jest algoritam k-means

pokušava pronaći rješenje koje ima što manji SSE. Slijedi prikaz algoritma k-means pseudo-kodom na visokoj razini na slici 2.1.

- Izaberi proizvoljne početne centroide c_j za grupe g_j, j = 1, …, k gdje je k broj grupa
- Ponavljaj
  - Za i = 1 do N gdje je N broj podataka u skupu
    - Odredi za vektor x_i najbliži centroid c_j, smjesti ga u grupu g_j.
  - Završi petlju
  - Za j = 1 do k
    - Ažuriranje centroida: odredi c_j kao aritmetičku sredinu vektora $x_i \in X$ gdje je x_i dio grupe g_j
  - Završi petlju
- Završi ponavljanje nakon određenog broja iteracija ili ako je ispunjen neki zaustavni kriterij

Slika 2.1. *Algoritam k-means*

Prema [4], velika prednost algoritma k-means je njegova računska jednostavnost, što ga čini privlačnim kandidatom za različite primjene. Njegova vremenska složenost je O(N*k*q) gdje je q broj iteracija potrebnih da algoritam konvergira. U većini slučajeva k i q su puno manji od N pa se može reći da je vremenska složenost linearna O(N), što znači da je k-means prikladan algoritam kada ima mnogo podataka za grupiranje. Također je jako jednostavan za implementaciju i lako se može unaprijediti ili modificirati ako je to potrebno. Neki od nedostataka algoritma k-means, prema [4], su sljedeći:

1. Algoritam k-means ne može garantirati konvergenciju u globalni minimum. Drugačiji početni centroidi grupa mogu proizvesti drugačije završne grupe, što znači da konvergira u drugačiji lokalni minimum.
2. Broj grupa k za skup podataka X se mora postaviti kao ulazni parametar. Loša procjena broja grupa može dovesti do lošeg grupiranja podataka.
3. Algoritam k-means je osjetljiv na ekstreme. Ekstremi u skupu podataka X moraju biti pridodani u neku grupu, time utječu na aritmetičku sredinu centroida grupe i time na završno grupiranje podataka.

2.1.2. Vrednovanje učinkovitosti

Vrednovanje učinkovitosti predstavlja ocjenjivanje koliko je dobro neki algoritam grupirao podatke. Omogućuje uspoređivanje algoritama za grupiranje podataka. Vrednovanje učinkovitosti se može iskoristiti za određivanje broja grupa na način da se odradi nekoliko grupiranja s različitim početnim brojem grupa i odabire se ono grupiranje s najboljom učinkovitosti [5]. Prema [1], postoje tri vrste vrednovanja učinkovitosti grupiranja, to su unutarnje, vanjsko i relativno vrednovanje. Vanjsko vrednovanje uspoređuje koliko je dobro particija grupirala podatke s već postojećom strukturom koja sadrži informacije o skupu podataka. Unutarnje vrednovanje ocjenjuje grupiranje direktno iz skupa podataka. Relativno vrednovanje uspoređuje particiju s particijama drugih algoritama ili particijama istog algoritma, ali s drugačijim ulaznim parametrima.

Funkcija cilja je funkcija koju algoritam optimira prilikom grupiranja. Funkcija cilja algoritma k-means je ranije spomenuti SSE koji računa zbroj kvadratnih udaljenosti svih podataka do njihovih centroida. Što je SSE manji time su podatci manje udaljeni od svojih centroida, te je grupiranje bolje. SSE nam pokazuje koliko su podaci unutar grupe slični jedni drugima, a različiti od podataka drugih grupa. Prema [6], SSE se definira na sljedeći način, formulom (2-6). Ako postoji N vektora $x_1, \ldots, x_N \in R^d$, i cijeli broj k, pronađi k centroida $c_1, \ldots, c_k \in R^d$ koje minimiziraju sljedeću funkciju.

$$f_{k-means} = \sum_{i=1}^{N} \min_{j \in [k]} \left\| x_i - c_j \right\|_2^2. \tag{2-6}$$

Dok se funkcija cilja koristi prilikom grupiranja i usmjerava kako teče grupiranje podataka, relativni indeksi (engl. *relative cluster validity indices/criteria*) služe za ocjenjivanje rezultata nakon što je izvršen proces grupiranja. Oni služe kako bi se usporedila različita rješenja. Postoje mnogi različiti relativni indeksi za vrednovanje k-particija. Neki od poznatijih su Davies-Bouldinov indeks (DBI) i Dunnov indeks. DBI se temelji na omjeru udaljenosti unutar grupa i između grupa. DBI se računa, prema [5], na sljedeći način, prema (2-7).

$$DBI = \frac{1}{k} \sum_{l=1}^{k} D_l, \tag{2-7}$$

gdje je k broj grupa, a $D_l = max_{l \neq m}\{D_{l,m}\}$. Izraz $D_{l,m}$ se računa kao $D_{l,m} = \frac{(\overline{d_l} + \overline{d_m})}{d_{l,m}}$, gdje su $\overline{d_l}$ *i* $\overline{d_m}$ prosječne udaljenosti unutar grupe za l-tu i m-tu grupu, a $d_{l,m}$ je udaljenost između te dvije grupe. Udaljenosti su računate kao Euklidske udaljenosti. Što je DBI manji, to su podatci bolje grupirani, to jest grupe su kompaktnije i više udaljene jedni od drugih.

Dunnov indeks se također temelji na geometrijskim mjerenjima kompaktnosti i odvojenosti grupa. Prema [5], Dunnov indeks definira se formulom (2-8).

$$DN = \min_{\substack{p,q \in \{1,\dots,k\} \\ p \neq q}} \left\{ \frac{\delta_{p,q}}{\max_{l \in \{1,\dots,k\}} \Delta l} \right\}, \tag{2-8}$$

gdje je Δl promjer l-te grupe i $\delta_{p,q}$ je udaljenost između grupa p i q. Udaljenost $\delta_{p,q}$ je definirana kao minimalna udaljenost između para objekata grupa p i q, to jest definirana je prema [5], formulom (2-9),

$$\min_{x_i \in G_p} \left\{ \min_{x_j \in G_q} \left\| x_i - x_j \right\|_2 \right\}. \tag{2-9}$$

Promjer Δl grupe l je definiran kao maksimalna udaljenost između para objekata unutar te grupe, prema [5], formulom (2-10),

$$\max_{x_i \in G_l} \left\{ \max_{x_j \in G_l} \left\| x_i - x_j \right\|_2 \right\}. \tag{2-10}$$

Za razliku od DBI indeksa, kod Dunnovog indeksa su veće vrijednosti one koje ukazuju na kompaktnije i bolje odvojene grupe.

2.2. Algoritam diferencijalne evolucije

Algoritam DE je stohastička metoda zasnovana na populaciji za globalnu optimizaciju. Dijeli mnoge karakteristike s ostalim evolucijskim algoritmima kao što su korištenje operatora mutacije, križanja i selekcije nad svojom populacijom [7]. DE započinje stvaranjem populacije koja se sastoji od nasumičnih rješenja problema te optimira taj problem tako što iterativnim postupkom dobiva što kvalitetnija rješenja nastala korištenjem operatora mutacije i križanja. Novonastala rješenja dobivena mutacijom i križanjem zamjenjuju stara u generaciji procesom selekcije ako su kvalitetnija. Algoritam DE ponavlja ovaj proces sve dok se ne izvrši neki određen maksimalni broj iteracija ili dok se ne postigne neki zaustavni kriterij. Na slici 2.2 slijedi prikaz algoritma DE na visokoj razini pseudo-kodom, a nakon toga su koraci algoritma DE detaljnije opisani.

- Postavi parametre NP, F i CR
- Inicijaliziraj početnu populaciju rješenja
- Za i = 0 do N (maksimalan broj iteracija)
  - Za j = 1 do NP (broj rješenja u populaciji)
    - Generiraj mutirani vektor $w_{j,i}$ s operatorom mutacije
    - Križaj mutirani vektor $w_{j,i}$ s ciljnim vektorom $v_{j,i}$ kako bi se napravio probni vektor $y_{j,i}$
    - Ako je $f(y_{j,i}) \leq f(v_{j,i})$ onda
      - $v_{j,i+1} = y_{j,i}$
    - Inače
      - $v_{j,i+1} = v_{j,i}$
    - Završi
  - Završi petlju
- Završi petlju

Slika 2.2. *Algoritam DE*

Algoritam započinje inicijalizacijom populacije rješenja. Populacija se sastoji od NP d-dimenzionalnih realnih vektora. Početna populacija je najčešće generirana koristeći uniformne nasumične vrijednosti unutar prostora GR = $[gr^{DG}, gr^{GG}] \subset \mathbb{R}^d$, gdje je $gr^{DG} = (gr_1^{DG}, \dots, gr_d^{DG})$ donja granica, a $gr^{GG} = (gr_1^{GG}, \dots, gr_d^{GG})$ gornja granica za sve dimenzije vektora. Svakom iteracijom kreiraju se nova rješenja u populaciji korištenjem operatora mutacije i križanja.

Mutacija je ključni element izvedbe algoritma DE [8]. Ona je prvi i primarni operator koji se izvodi na populaciji. Mutacija se može opisati na sljedeći način. Za svako rješenje u populaciji, to jest za ciljni vektor (engl. *target vector*) $v_{j,i}$ kreira se mutirani vektor (engl. *mutant vector*) $w_{j,i}$ kao prema formuli (2-11),

$$w_{j,i} = v_{r_1,i} + F * (v_{r_2,i} - v_{r_3,i}), \tag{2-11}$$

gdje u vektoru $w_{j,i}$ j predstavlja indeks podatka u populaciji, i predstavlja broj iteracije, r_1, r_2 i r_3 su nasumično odabrani indeksi iz {1, …, NP} za koje vrijedi $j \neq r_1 \neq r_2 \neq r_3$. F predstavlja faktor skaliranja (engl. *scaling factor*) i vrijedi $F \in [0,+\infty\rangle$. Faktor skaliranja se mora postaviti kao ulazni parametar prije pokretanja algoritma. Najčešće se postavlja u intervalu <0, 1]. Na 0 ga nema smisla postaviti jer tada je mutirani vektor jednak vektoru $v_{r_1,i}$, a prema [7], vrijednosti veće od 1

rijetko daju kvalitetne rezultate. Vektor $v_{r_1,i}$ se još naziva i bazni vektor i može se označiti kao v_b. Iako se druga dva vektora najčešće izabiru nasumično, to ne mora značiti i za bazni vektor. Zbog važnosti mutacije, ponuđeno je mnoštvo različitih strategija mutacije u literaturi. Najčešće se razlikuju u načinu na koji se izabiru vektori koji sudjeluju u mutaciji. Dvije najčešće mutacije i one koje su originalno predložene su rand/1 i best/1. Mutacija rand/1 je već opisana u prethodnoj formuli (2-11), u ovoj mutaciji sva tri vektora koja sudjeluju u mutaciji su izabrana nasumično. Mutacija best/1 se izvodi na sljedeći način kao prema formuli (2-12).

$$w_{j,i} = v_{najbolji,i} + F * \left(v_{r_2,i} - v_{r_3,i}\right), \tag{2-12}$$

gdje indeks najbolji predstavlja najbolje rješenje u populaciji za tu iteraciju, F je ranije spomenuti faktor skaliranja, a r_2 i r_3 su nasumično odabrani indeksi iz {1, …, NP} tako da vrijedi $j \neq r_2 \neq r_3$. Prema [8], korištenje mutacije rand/1 rezultira opsežnim istraživanjem prostora pretrage, a korištenje mutacije best/1 rezultira pretragom prostora na najbolji mogući način. Mutacija može rezultirati mutiranim vektorom koji se nalazi izvan prostora pretrage, te se zbog toga mora implementirati rukovanje ograničenjem granica (engl. *bound-constraint handling*). Postoji mnogo predloženih načina rukovanja ograničenjima u literaturi, jedan od najjednostavnijih je postavljanje vektora na gornju granicu ako je mutirani vektor veći od nje, ili na donju granicu ako je manji, opisano kao prema formuli (2-13).

$$w_{d,j,i} \begin{cases} gr_d^{DG}, ako\ je\ w_{d,j,i} < gr^{DG}, \\ gr_d^{GG}, ako\ je\ w_{d,j,i} > gr^{GG}, \\ w_{d,j,i}, inače, \end{cases} \tag{2-13}$$

gdje u vektoru $w_{d,j,i}$ d predstavlja dimenziju vektora, a gr_d^{DG} i gr_d^{GG} su ranije spomenute donje i gornje granice za tu dimenziju vektora.

Nakon mutacije, slijedi križanje. U križanju ciljni i mutirani vektor se križaju kako bi se kreiralo novo rješenje koje se naziva probni vektor (engl. *trial vector*). Probni vektor je sastavljen od elemenata ciljnog i mutiranog vektora. Križanje se može prikazati na sljedeći način, kao prema formuli (2-14).

$$y_{d,j,i} = \begin{cases} w_{d,j,i}\ ako\ rand_d(0,1) \leq CR\ ili\ d = r_d, \\ v_{d,j,i}\ inače, \end{cases} \tag{2-14}$$

gdje je $rand_d(0,1)$ uniformna nasumična vrijednost u intervalu [0, 1], CR je stopa križanja (engl. *crossover rate*) koja utječe na to koliko komponenata je naslijeđeno od mutanta i za nju vrijedi $CR \in [0,1]$, a r_d je nasumično odabrana vrijednost iz {1, …, d} i ona osigurava da probni vektor

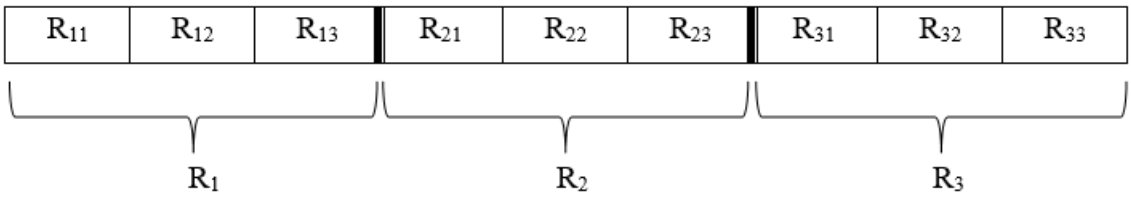
neće biti isti kao ciljni vektor. Novokreirani probni vektor se bori za opstanak s ciljnim vektorom. Procesom selekcije odabire se vektor za sljedeću generaciju, selekcija se može prikazati kao prema formuli (2-15).

$$v_{j,i+1} = \begin{cases} y_{j,i} \; ako \; f(y_{j,i}) \leq f(v_{j,i}), \\ v_{j,i}, inače. \end{cases} \tag{2-15}$$

Probni vektor se izabire ako ima manju ili jednaku funkciju dobrote od ciljnog, inače ostaje ciljni vektor u populaciji. Algoritam se može vrtjeti dok se ne dođe do nekog kriterija prekida algoritma ili dok se ne odradi neki određen broj iteracija.

2.2.1. Primjena za grupiranje podataka

Kako bi koristili evolucijske algoritme za grupiranje podataka, prvo se mora odrediti kako se predstavljaju rješenja u populaciji i kako se ta rješenja vrednuju, to jest mora se odrediti odgovarajuća funkcija cilja. U ovom radu, rješenja se predstavljaju redom kao centroidi grupa, tako da se populacija sastoji od NP k*d rješenja, gdje k predstavlja broj grupa, a d broj značajki skupa podataka. Primjer jednog rješenja s 3 grupe i 3 značajke u populaciji nalazi se na slici 2.3.



Slika 2.3. *Prikaz reprezentacije rješenja s tri grupe i tri značajke*

Na slici 2.3, R_1, R_2 i R_3 predstavljaju grupe unutar rješenja, a R_{11}, …, R_{33} značajke unutar tih grupa, dok sve to zajedno predstavlja jednu reprezentaciju rješenja. Populacija u algoritmu se sastoji od NP takvih rješenja. Centroidi unutar rješenja su izabrani nasumično iz korištenog skupa podataka s uvjetom da se centroidi ne smiju ponavljati u populaciji rješenja. Broj grupa i značajki ovisi o korištenom skupu podataka. S ovakvom reprezentacijom rješenja pripadajuće particije je jednostavno dobiti, podatci iz skupa se dodjeljuju grupi čijem su centroidu najbliže, a za računanje udaljenosti između podataka i centroida se koristi ranije spomenuta Euklidska udaljenost.

Kao funkcija cilja za način vrednovanja rješenja, to jest particija, koristi se ranije opisani DBI. To znači da algoritam pokušava pronaći rješenje koje rezultira s najmanjim mogućim DBI. Postoji mogućnost dobivanja i nevaljanih particija, a to je kada particija ima grupu koja je prazna, kojoj nijedan podatak nije dodijeljen. U tome slučaju se prije vrednovanja radi korekcija, grupi kojoj

nije dodan nijedan podatak dodaje se onaj podatak koji mu je najbliži te se tek onda radi vrednovanje s DBI.

2.3. Pregled postupaka za grupiranje podataka

U sljedećih par odlomaka je opisano nekoliko prijašnjih radova koji se odnose na grupiranje podataka s evolucijskim i sličnim algoritmima. Za svaki rad se navode ključni elementi kao što su korišteni algoritmi, ključni zaključci radova i korištene funkcije cilja za vođenje pretrage.

Paterlini i Krink su u radu [9] usporedili performanse genetskog algoritma (engl. *genetic algorithm,* GA), algoritma optimizacije rojem čestica (engl. *particle swarm optimisation,* PSO) i algoritma DE. Paterlini i Krink su u svojoj analizi došli do zaključka da je algoritam DE konzistentno imao bolje performanse od algoritama GA i PSO, bio je i precizniji i pouzdaniji. Algoritmi GA i PSO su imali slične performanse kao i algoritam DE samo kod jednostavnih skupova podataka. Također su zaključili da je algoritam DE jednostavniji za implementaciju i da za razliku od algoritama GA i PSO ne zahtjeva znatno ugađanje ulaznih parametara. Kao funkcija cilja za vođenje pretrage korištene su TRW (engl. *trace within criterion*), VRC (engl. *variance ratio criterion*) i MC (engl. *Mariott's criterion*).

Kwedlo je u radu [10] predložio algoritam DE-KM (engl. *differential evolution – k-means*), ovaj algoritam kombinira algoritam DE s algoritmom k-means. Algoritam k-means je uključen u proces algoritma DE na dva načina. Prvi način je da se na početku algoritam k-means koristi kako bi se dobili centroidi za svako rješenje koje predstavlja početnu populaciju u algoritmu DE. Drugi način je da algoritam k-means podesi svako novo rješenje koje se dobije operatorima algoritma DE. Kao funkcija cilja za vođenje pretrage je korištena mjera SSE. Algoritam je uspoređen s algoritmom DE i nekoliko drugih algoritama koji koriste k-means. Rezultati eksperimentalne analize su pokazali da je algoritam DE-KM u slučaju dovoljno velikog broja grupa pronašao rješenja s manjim vrijednostima SSE nego ostali analizirani algoritmi.

Martinović i Bajer su u radu [11] predstavili DEMM (engl. *differential evolution incorporating macromutations*). To je algoritam DE za grupiranje koji je koristio makromutacije kao dodatni mehanizam za pretraživanje prostora pretrage. Razlog tomu je to što rješenja unutar populacije s vremenom postanu jako slična pa sa standardnim operatorima mutacije i križanja pretraga postane teška ili nemoguća. Zbog toga su dodane makromutacije koje omogućuju bolje istraživanje prostora pretrage. Usporedili su ga s nekoliko različitih algoritama koji koriste DE i s algoritmom

PSO. Algoritam DEMM je pronašao rješenja visoke kvalitete i bio je vrlo stabilan. Kao funkcija cilja za vođenje pretrage korišten je DBI.

Das, Abraham i Konar su u radu [12] predstavili novi algoritam temeljen na DE koji se zove ACDE (engl. *automatic clustering using an improved differential evolution algorithm*). Važna značajka ovog predstavljenog algoritma je da automatski pronalazi optimalan broj grupa što znači da se broj grupa ne mora znati prije pokretanja algoritma. Usporedili su predloženi algoritam ACDE s dva evolucijska algoritma za grupiranje i s jednim bioinspiriranim algoritmom za grupiranje. U eksperimentalnoj analizi ACDE algoritam je imao bolje performanse od ostala tri analizirana algoritma nad korištenim skupovima podataka. Kao funkcije cilja za vođenje pretrage korištene su DBI i CS. CS je mjera za vrednovanje učinkovitosti predložena u [13].

Abdel-Kader je u radu [14] predstavio hibridni algoritam za grupiranje podataka koji spaja GAI-PSO (engl. *genetically improved PSO algorithm*) i algoritam k-means. Ideja iza ovog predloženog algoritma je da ponudi brzo grupiranje podataka i izbjegne preuranjenu konvergenciju u lokalni optimum. GAI-PSO kombinira značajke genetskih algoritama i algoritma PSO. GAI-PSO predstavlja prvu fazu hibridnog algoritma i on pretražuje prostor pretrage kako bi pronašao optimalno rješenje koje se koristi kao inicijalni centroidi grupa u sljedećoj fazi. Ti centroidi se u sljedećoj fazi optimiraju pomoću algoritma k-means. Predloženi hibridni algoritam je uspoređen s nekoliko algoritama za grupiranje inspiriranih evolucijom i s algoritmom k-means. U eksperimentalnoj analizi GAI-PSO+k-means je imao bolja rješenja i brže je konvergirao od ostalih algoritama. Kao funkcija cilja za vođenje pretrage korištena je funkcija koja računa sumu Euklidskih udaljenosti podataka od svojih centroida.

Cho i Nyunt su u radu [15] predstavili algoritam DE s modificiranim operatorom mutacije. Ideja rada je bila predstaviti algoritam koji pronalazi bolja rješenja s bržom konvergencijom. To su pokušali uz modificirani operator mutacije koji pokušava uravnotežiti pretragu algoritma tako da se u isto vrijeme dobije opsežnije pretraga prostora ali i da ona bude iskorištena na najbolji mogući način. Predloženi algoritam je uspoređen s dva algoritma DE koji koriste dva najčešća operatora mutacije rand/1 i best/1, te s još četiri bioinspirirana algoritma. Eksperimentalna analiza na pet skupova podataka je prikazala da predloženi algoritam CDE-MM pronalazi bolja rješenja od ostalih i da je stabilan. Kao funkcija cilja za vođenje pretrage korištena je suma Euklidskih udaljenosti podataka od svojih centroida.

3. OSTVARENO PROGRAMSKO RJEŠENJE

Programsko rješenje ostvareno je u programskom jeziku Python. Napisana su dva programska rješenja, jedno rješenje je algoritam DE s operatorom mutacije rand/1, a drugo je algoritam DE s operatorom mutacije best/1.

Oba programska rješenja omogućuju učitavanje skupa podataka u CSV (engl. *comma separated values*) formatu. Također omogućuju postavljanje željenog broja grupa, parametara algoritma DE i maksimalnog broja iteracija po izvođenju. Kao izlaz nakon svakog izvođenja ispisuju kvalitetu najboljeg rješenja u smislu korištene funkcije cilja za vođenje pretrage (DBI) i funkcije cilja koju optimira algoritam k-means (SSE) kako bi mogli napraviti usporedbu. Kako bi se olakšala izvedba eksperimentalne analize rezultati izvođenja se spremaju u zadanu tekstualnu datoteku.

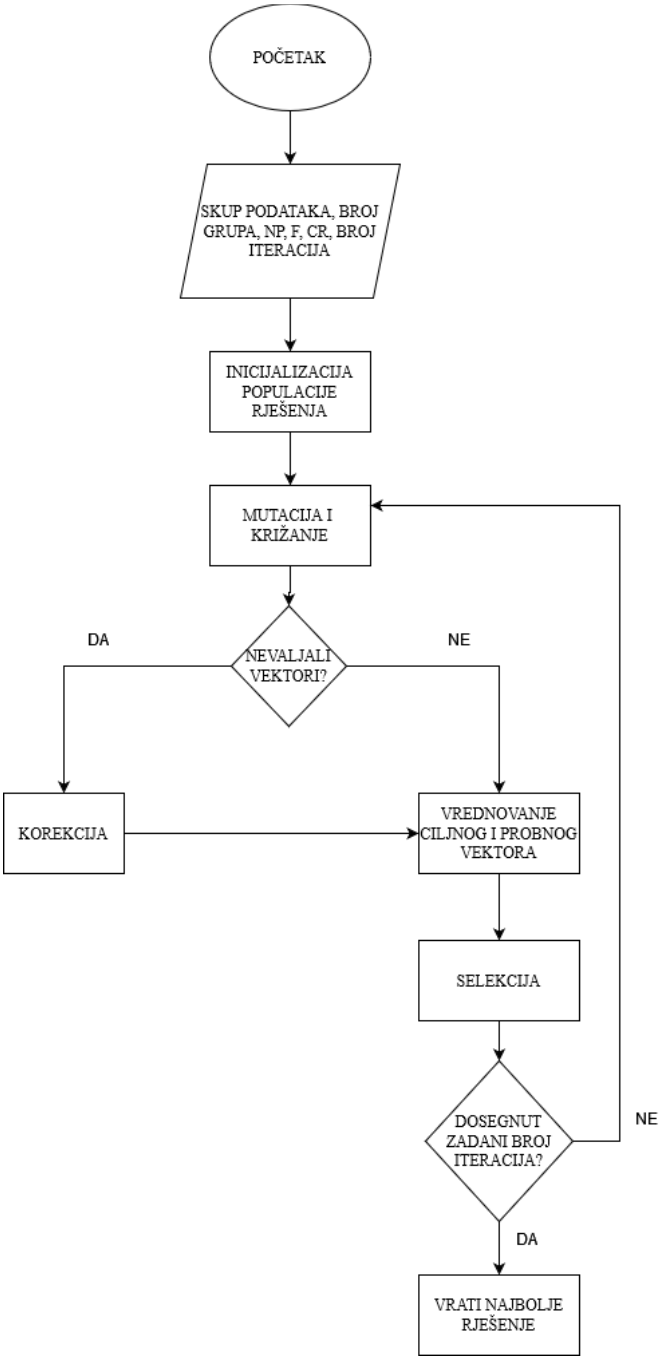
Algoritam k-means je implementiran pomoću Python biblioteke scikit-learn [16]. To je besplatna Python biblioteka koja sadrži razne klasifikacijske, regresijske algoritme i algoritme grupiranja, među kojima je i algoritam k-means. Ova biblioteka za algoritam k-means omogućuje postavljanje željenog broja grupa, način inicijalizacije centroida i maksimalan broj iteracija između ostalog. Za algoritam k-means također se ispisuju i zapisuju u određenu tekstualnu datoteku kvalitete dobivene particije u smislu DBI i SSE.

3.1. Način rada programskog rješenja

Ulaz programskog rješenja je skup podataka u CSV formatu, donja i gornja granica podataka, broj grupa i značajki, broj rješenja u populaciji NP, faktor skaliranja F, stopa križanja CR i broj iteracija jednog izvođenja. Korisnik ima mogućnost podešavanja broja grupa, parametara algoritma DE i maksimalnog broja iteracija po izvođenju. Ime skupa podataka i broj značajki se podešava u samom kodu. Donja i gornja granica podataka se postave automatski za učitani skup podataka.

Nakon što su definirani svi ulazi i korisnik je podesio sve mogućnosti, inicijalizira se populacija rješenja. Populacija rješenja se inicijalizira s nasumičnim podatcima iz skupa podataka s uvjetom da se podatci u rješenjima ne smiju ponavljati. Nakon inicijalizacije, algoritam započinje s izvedbom. Postupci mutacije, križanja i selekcije se ponavljaju dok se ne dosegne zadani broj iteracija. Prije selekcije izvodi se vrednovanje ciljnog i probnog vektora s DBI. U slučaju da je particija vektora nevaljala vrši se korekcija prije vrednovanja. Selekcija postavlja vektor s boljom funkcijom dobrote u novonastalu populaciju.

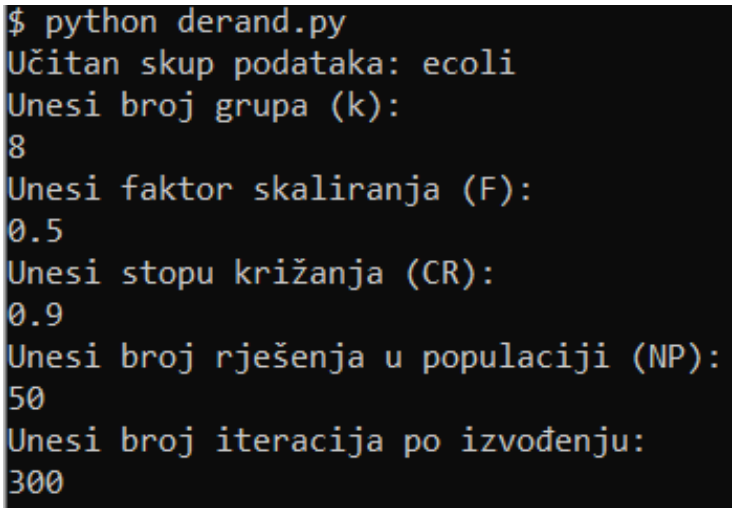
Kada se dosegne zadani broj iteracija, algoritam DE vraća najbolje rješenje u populaciji. Pomoću biblioteke scikit-learn algoritam k-means se izvodi nad skupom podataka. Vrednuje se najbolje rješenje algoritma DE i algoritma k-means sa SSE i DBI, dobivene vrijednosti se ispisuju na konzoli i zapisuju u određenu datoteku. Na slici 3.1 je prikazan dijagram toka algoritma DE.



Slika 3.1. *Dijagram toka algoritma DE*

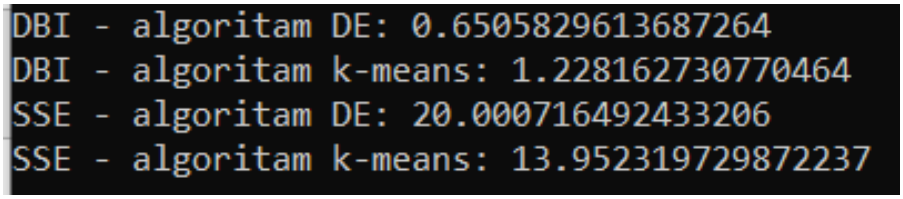
3.2. Prikaz i način uporabe programskog rješenja

Programsko rješenje se pokreće u konzoli, ispisuje se učitan skup podataka te korisnik ima mogućnost podešavanja broja grupa, parametara algoritma DE i broja iteracija po izvođenju. Na slici 3.2 je prikazan primjer unosa ulaznih parametara.



Slika 3.2. *Unos ulaznih parametara*

Nakon unosa potrebnih parametara, izvršava se algoritam DE. Po završetku izvršavanja programskog rješenja, u konzoli su ispisane dobivene vrijednosti mjera DBI i SSE algoritama DE i k-means. Te vrijednosti su također zapisane u tekstualne datoteke koje su definirane u kodu programskog rješenja. Primjer izlaza je prikazan na slici 3.3.



Slika 3.3. *Izlaz programskog rješenja*

4. EKSPERIMENTALNA ANALIZA

Cilj eksperimentalne analize je ispitati učinkovitost algoritma DE za grupiranje podataka prilikom korištenja dva različita operatora mutacije. Analiza prikazuje razliku u učinkovitosti grupiranja podataka tih operatora mutacije. Obje inačice algoritma DE su također uspoređene s algoritmom k-means za grupiranje podataka. Učinkovitost algoritama je uspoređena s mjerom DBI i mjerom SSE. Bolji rezultati odgovaraju manjim vrijednostima DBI i SSE. Analiza je provedena na nekoliko sintetičkih i stvarnih skupova podataka. Rezultati analize su statistički obrađeni i iz njih su prezentirani dobiveni zaključci. U tablici 4.1 su prikazane karakteristike sintetičkih skupova podataka. Opis sintetičkih skupova podataka je dan u prilogu 1. U tablici 4.2 su prikazane karakteristike stvarnih skupova podataka preuzetih s UCI repozitorija.

Tablica 4.1. *Karakteristike sintetičkih skupova podataka*

Skup podataka	Broj primjeraka	Broj značajki	Broj grupa
Data_k3_1	300	2	3
Data_k4_3	525	2	4
Data_k5_4	625	2	5
Data_k6_1	525	2	6

Tablica 4.2. *Karakteristike stvarnih skupova podataka*

Skup podataka	Broj primjeraka	Broj značajki	Broj grupa
Cancer	683	9	2
Iris	150	4	3
Glass	214	9	6
Ecoli	336	7	8
Yeast	1484	8	10

4.1 Postavke eksperimenta

Zbog stohastičke prirode algoritma DE, broj ponavljanja izvođenja je 30 puta kako bi se dobio uvid u učinkovitost algoritama. Algoritam k-means je također ponovljen 30 puta. U tablici 4.3 su prikazane postavke parametara algoritama. Za obje inačice algoritma DE to su parametri NP, CR, F i broj iteracija. Veličina populacije NP, prema [17], treba biti između 5*d i 10*d gdje d predstavlja dimenzionalnost skupa podataka. Najveću dimenzionalnost imaju skupovi podataka *Cancer* i *Glass*, te bi za njih NP trebala biti između [45, 90]. Za ovu analizu NP je postavljen na

50 za obje inačice algoritma DE za sve skupove podataka. Ta se veličina populacije često koristi u literaturi, a i nalazi se u ranije spomenutom intervalu [45, 90]. Stopa križanja, prema [17], treba biti 0.1 kao početni izbor, a 0.9 i 1 se mogu koristiti ako se želi postići brža konvergencija. Storn također navodi kako se u slučaju loše učinkovitosti, CR treba odabrati iz [0.8, 1] [18]. U literaturi se često koristi vrijednost 0.9 pa se ta vrijednost koristi i u ovom radu za obje inačice algoritma DE. Faktor skaliranja, prema [17], treba biti između 0.4 i 1, a pri podešavanju tog parametra za prvu vrijednost preporučuju 0.5. U radu se koristi vrijednost 0.5 kao faktor skaliranja za inačicu DE/rand/1/bin, a za inačicu DE/best/1/bin se obično koristi nešto veća vrijednost te je faktor skaliranja postavljen na 0.9 kao u radu [11]. Za algoritam k-means postavila se metoda za inicijalizaciju centroida i broj iteracija.

Tablica 4.1. *Postavke parametara*

Algoritam	**Parametri**
DE/rand/1/bin	NP = 50, CR = 0.9, F = 0.5, broj iteracija = 300
DE/best/1/bin	NP = 50, CR = 0,9, F = 0.9, broj iteracija = 300
k-means	nasumična inicijalizacija centroida prema podacima u skupu, broj iteracija = 300

4.2 Rezultati

Manja vrijednost SSE ukazuje na to da su podatci unutar grupe bliži svom centroidu, to jest da su grupe bolje definirane i kompaktnije. Dok SSE uzima u obzir samo udaljenost podataka od svog centroida, DBI u obzir osim toga uzima i udaljenost između centroida grupa. Zbog toga, DBI daje širu ocjenu kvalitete nego SSE. Manja vrijednost DBI ukazuje da su podatci grupirani u bolje, kompaktnije grupe, ali i da su grupe međusobno udaljenije. U tablici 4.4 i tablici 4.5 su prikazani rezultati mjerenja DBI i SSE algoritama u usporedbi postignuti na sintetičkim skupovima. Tablice prikazuju prosječni rezultat rješenja (f_{pro}), njihovu standardnu devijaciju (σ), rezultat najboljeg (f_{min}) i najgoreg (f_{max}) rješenja, njihov raspon (f_{max}-f_{min}), te medijan rješenja.

Tablica 2.4. *DBI rezultati postignuti na sintetičkim skupovima*

Skup podataka	Algoritam	f_{pro}	σ	f_{min}	f_{max}	f_{max}-f_{min}	medijan
Data_k3_1	DE/rand/1/bin	**2.68e-1**	**0**	**2.68e-1**	**2.68e-1**	**0**	**2.68e-1**
	DE/best/1/bin	**2.68e-1**	**0**	**2.68e-1**	**2.68e-1**	**0**	**2.68e-1**
	k-means	4.55e-1	3.11e-1	**2.68e-1**	1.078	8.1e-1	**2.68e-1**
Data_k4_3	DE/rand/1/bin	**5.05e-1**	1.45e-3	**5.02e-1**	**5.07e-1**	5e-3	**5.06e-1**
	DE/best/1/bin	5.07e-1	**4.59e-4**	5.04e-1	**5.07e-1**	**3e-3**	5.07e-1
	k-means	6.53e-1	2.38e-1	5.07e-1	1.087	5.8e-1	5.11e-1
Data_k5_4	DE/rand/1/bin	5.01e-1	**4.11e-4**	4.99e-1	5.01e-1	**2e-3**	5.01e-1
	DE/best/1/bin	**4.98e-1**	5.25e-4	**4.97e-1**	**5e-1**	3e-3	**4.98e-1**
	k-means	6.24e-1	1.78e-1	5.01e-1	9.9e-1	4.89e-1	5.04e-1
Data_k6_1	DE/rand/1/bin	3.77e-1	**1.85e-3**	3.74e-1	**3.78e-1**	**4e-3**	3.78e-1
	DE/best/1/bin	**3.75e-1**	2.09e-3	**3.73e-1**	**3.78e-1**	5e-3	**3.74e-1**
	k-means	6.71e-1	1.5e-1	3.78e-1	9.62e-1	5.84e-1	6.94e-1

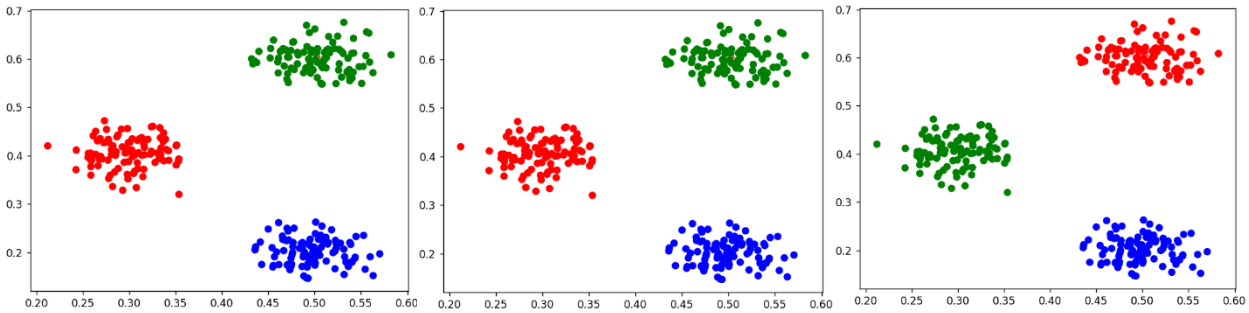
Tablica 4.3. *SSE rezultati postignuti na sintetičkim skupovima*

Skup podataka	Algoritam	f_{pro}	σ	f_{min}	f_{max}	f_{max}-f_{min}	medijan
Data_k3_1	DE/rand/1/bin	**5.25e-1**	2e-3	**5.25e-1**	5.38e-1	1.3e-2	**5.25e-1**
	DE/best/1/bin	**5.25e-1**	**1.66e-4**	**5.25e-1**	**5.25e-1**	**0**	**5.25e-1**
	k-means	1.572	1.737	**5.25e-1**	4.492	3.967	**5.25e-1**
Data_k4_3	DE/rand/1/bin	2.765	1.16e-1	**2.708**	3.162	4.54e-1	**2.708**
	DE/best/1/bin	**2.719**	**1.44e-2**	**2.708**	**2.785**	**7.7e-2**	2.715
	k-means	3.531	1.377	**2.708**	5.999	3.291	**2.708**
Data_k5_4	DE/rand/1/bin	**2.691**	1.68e-2	2.688	2.781	9.3e-2	**2.688**
	DE/best/1/bin	2.706	**7.07e-3**	2.693	**2.727**	**3.4e-2**	2.705
	k-means	3.492	1.218	**2.687**	6.253	3.566	**2.688**
Data_k6_1	DE/rand/1/bin	**1.987**	**2.62e-2**	**1.979**	**2.126**	**1.47e-1**	**1.979**
	DE/best/1/bin	2.05	1.93e-1	**1.979**	2.993	1.014	1.992
	k-means	3.29	1.048	**1.979**	5.084	3.105	2.968

Rezultati u smislu mjerenja DBI sugeriraju da su obje inačice algoritma DE bolje grupirale podatke od algoritma k-means. Obje inačice algoritma DE su također postigle malu standardnu devijaciju i raspon kvalitete rješenja, što ukazuje na to da su vrlo stabilne i da su za svako ponavljanje pronašli slično rješenje. Za razliku od njih, algoritam k-means je imao veću standardnu devijaciju i raspon kvalitete rješenja što ukazuje na to da u nekim slučajevima algoritam k-means pronalazi puno gore rješenje nego što sugerira prosjek i da je manje stabilan u odnosu na algoritam DE. To se može vidjeti i po tome da je k-means za svaki sintetički skup pronašao najgore rješenje (f_{max}) od svih algoritama. Inačica algoritma DE/best/1/bin je bila blago bolja kod sintetičkih skupova s većim brojem grupa (u ovom slučaju s pet i šest), dok je inačica DE/rand/1/bin imala blago bolji rezultat DBI kod sintetičkog skupa s četiri grupe. No, razlike u prosječnim rezultatima DBI su zanemarive. Kod sintetičkog skupa s tri grupe, obje inačice algoritma DE su postigli iste rezultate.

Rezultati u smislu mjerenja SSE sugeriraju da su kod obje inačice algoritma DE podatci unutar grupe bili manje udaljeni od svog centroida od onih kod algoritma k-means. Obje inačice algoritma DE su postigli iste ili jako slične rezultate SSE kod sintetičkog skupa s tri grupe. Dok kod ostalih sintetičkih skupova, rezultati za SSE kod inačica algoritma DE su bili suprotni od onih kod DBI. Kod skupa s četiri grupe blago bolji SSE je imala inačica algoritma DE/best/1/bin, a kod skupova s pet i šest grupa blago bolji SSE je imala inačica DE/rand/1/bin. No te razlike, isto kao kod DBI rezultata nisu velike. Standardna devijacija i raspon kvalitete su također bili manji kod obje inačice algoritma DE nego kod algoritma k-means, isto kao i kod DBI rezultata. To opet ukazuje na to, da su inačice algoritma DE stabilnije i daju konzistentnije rezultate nego algoritam k-means. Algoritam DE/best/1/bin je bio stabilniji kod svakog skupa osim kod onog sa šest grupa. Algoritam k-means je za svaki sintetički skup, kao i kod mjerenja DBI, pronalazio najgore rješenje SSE.

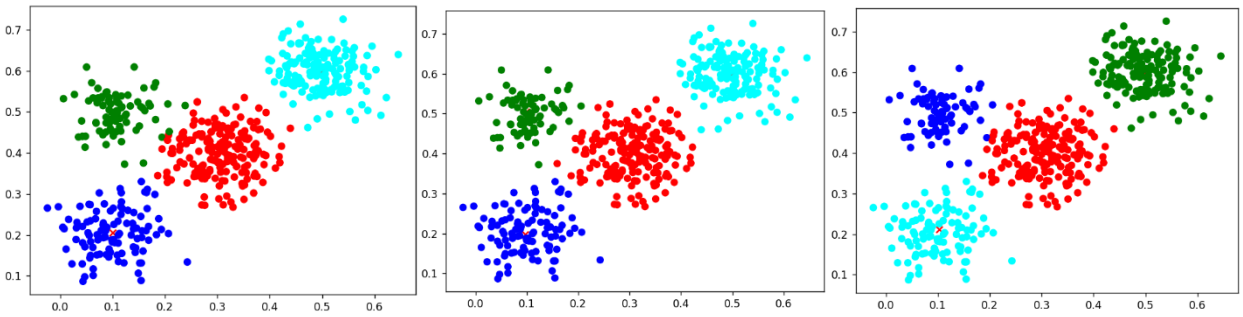
Mjerenja vrijednosti DBI i SSE ukazale su da kod grupiranja sintetičkih skupova podataka obje inačice algoritma DE grupiraju podatke u kompaktnije grupe koje su međusobno razdvojenije što sugerira njihov manji DBI i da su podatci unutar grupa manje udaljeni od svojih centroida što sugerira njihov manji SSE nego što je to slučaj kod algoritma k-means. S obzirom da su sintetički skupovi podataka u dvije dimenzije, njihovo grupiranje se lako može prikazati grafovima. Slijede slike raspršenih grafova koji prikazuju kako su algoritmi grupirali podatke sintetičkih skupova. Grafovi su dobiveni od posljednjeg ponavljanja svakog algoritma.



(a) (b) (c)

Slika 4.1. *Raspršeni graf za sintetički skup Data_k3_1 (a) k-means (DBI = 0.268, SSE = 0.525) (b) DE/rand/1/bin (DBI = 0.268, SSE = 0.525) (c) DE/best/1/bin (DBI = 0.268, SSE = 0.525)*

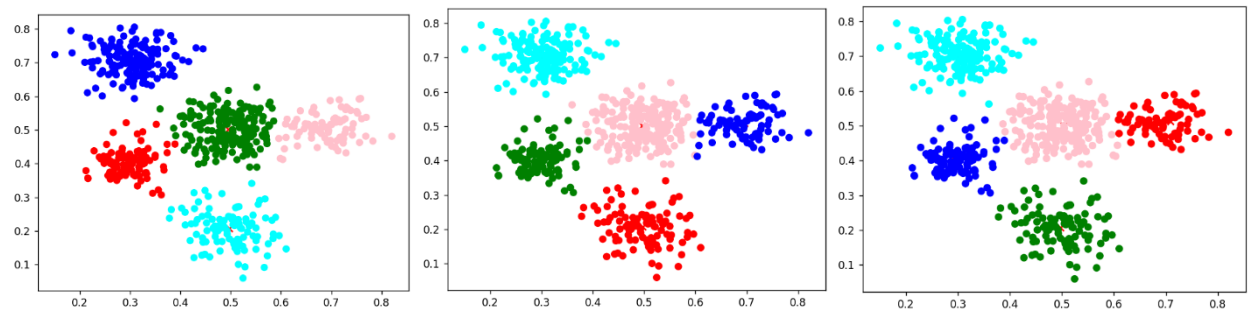
Na slici 4.1 se vidi raspršeni graf za sintetički skup s tri grupe. Sva tri algoritma su dali iste rezultate DBI i SSE, pa su tako i grafovi isti.



(a) (b) (c)

Slika 4.2. *Raspršeni graf za sintetički skup Data_k4_3 (a) k-means (DBI = 0.511, SSE = 2.708) (b) DE/rand/1/bin (DBI = 0.504, SSE = 2.746) (c) DE/best/1/bin (DBI = 0.507, SSE = 2.712)*

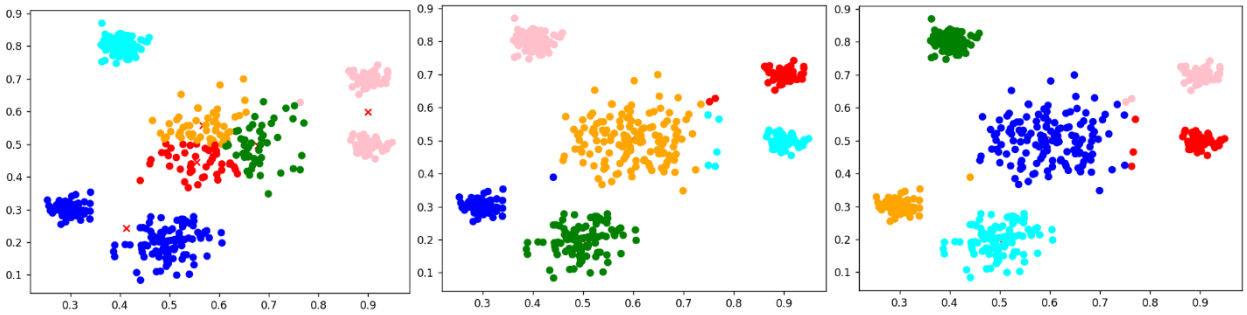
Na slici 4.2 se vidi raspršeni graf za sintetički skup s četiri grupe. Ovdje postoje neke razlike, razlike u DBI i SSE su jako male pa su i razlike u grafovima male. Vidljivo je prema vrijednosti DBI da je kod algoritma DE/rand/1/bin blago bolja odvojenost grupa nego kod ostala dva algoritma. Osim prema vrijednosti DBI to se može primijetiti i po grafu na slici 4.2 (b) jer je crvena grupa bolje odvojena od zelene nego kod ostala dva algoritma. Ali u ovom ponavljanju izvođenja sva tri algoritma su imali slično grupiranje podataka, i prema slikama i rezultatima.



(a) (b) (c)

Slika 4.3. *Raspršeni graf za sintetički skup Data_k5_4 (a) k-means (DBI = 0.504, SSE = 2.687) (b) DE/rand/1/bin (DBI = 0.501, SSE = 2.688) (c) DE/best/1/bin (DBI = 0.498, SSE = 2.71)*

Na slici 4.3 su prikazani raspršeni grafovi za sintetički skup s pet grupa. I u ovom slučaju su razlike teško vidljive. Svi algoritmi su za ovo ponavljanje izvođenja slično izvršili grupiranje, što se vidi prema slikama i mjerenjima.



(a) (b) (c)

Slika 4.4. *Raspršeni graf za sintetički skup Data_k6_1 (a) k-means (DBI = 0.765, SSE = 4.421) (b) DE/rand/1/bin (DBI = 0.378, SSE = 1.979) (c) DE/best/1/bin (DBI = 0.373, SSE = 1.992)*

Na slici 4.4 su prikazani raspršeni grafovi za sintetički skup sa šest grupa. U ovom slučaju se najviše vide razlike u grupiranju između tri algoritma. Algoritam k-means je u ovom ponavljanju izvođenja najgore grupirao podatke, to se vidi prema mjerenjima gdje algoritam k-means ima puno gori DBI i SSE nego inačice algoritma DE. Na slikama se također vidi da k-means nije najbolje grupirao podatke, jer je podatke koji su zbijeni u sredini grafa podijelio na tri grupe umjesto da je sve svrstao u jednu grupu kao što su to učinile obje inačice algoritma DE. Razlike između DE/rand/1/bin i DE/best/1/bin algoritma su minimalne, i po mjerenjima i po slikama.

U tablicama 4.6 i 4.7 su prikazani rezultati mjerenja DBI i SSE algoritama u usporedbi postignuti na stvarnim skupovima. Tablice kao i kod sintetičkih skupova prikazuju prosječni rezultat rješenja (f_{pro}), njihovu standardnu devijaciju (σ), rezultat najboljeg (f_{min}) i najgoreg (f_{max}) rješenja, njihov raspon (f_{max}-f_{min}) te medijan rješenja.

Tablica 4.4. *DBI rezultati postignuti na stvarnim skupovima*

Skup podataka	Algoritam	f_{pro}	σ	f_{min}	f_{max}	f_{max}-f_{min}	medijan
Cancer	**DE/rand/1/bin**	4.57e-1	9.27e-3	4.4e-1	4.63e-1	2.3e-2	4.63e-1
	DE/best/1/bin	**4.37e-1**	1.57e-2	**3.73e-1**	**4.59e-1**	8.6e-2	**4.4e-1**
	k-means	4.65e-1	**1.36e-3**	4.63e-1	4.66e-1	**3e-3**	4.66e-1
Iris	**DE/rand/1/bin**	6.62e-1	**0**	6.62e-1	6.62e-1	**0**	6.62e-1
	DE/best/1/bin	**6.17e-1**	2.2e-2	**5.7e-1**	**6.38e-1**	6.8e-2	**6.26e-1**
	k-means	6.87e-1	8.47e-2	6.62e-1	1.014	3.52e-1	6.66e-1
Glass	**DE/rand/1/bin**	**5.44e-1**	**1e-2**	5.1e-1	**5.65e-1**	**5.5e-2**	**5.46e-1**
	DE/best/1/bin	5.64e-1	4.66e-2	**4.53e-1**	6.93e-1	2.4e-1	5.65e-1
	k-means	1.11	1.6e-1	8.15e-1	1.531	7.16e-1	1.099
Ecoli	**DE/rand/1/bin**	**7e-1**	**7.46e-2**	6.39e-1	**8.64e-1**	**2.25e-1**	**6.51e-1**
	DE/best/1/bin	8.22e-1	7.89e-2	**6.34e-1**	9.83e-1	3.49e-1	8.47e-1
	k-means	1.401	1.27e-1	1.181	1.651	4.7e-1	1.395
Yeast	**DE/rand/1/bin**	**8.72e-1**	7.73e-2	**7.3e-1**	**1.046**	3.16e-1	**8.59e-1**
	DE/best/1/bin	9.71e-1	**6.88e-2**	8.09e-1	1.073	**2.64e-1**	9.89e-1
	k-means	1.541	1.03e-1	1.371	1.681	3.1e-1	1.562

Tablica 4.5. *SSE rezultati postignuti na stvarnim skupovima*

Skup podataka	Algoritam	f_{pro}	σ	f_{min}	f_{max}	f_{max}-f_{min}	medijan
Cancer	**DE/rand/1/bin**	2.275e7	2.647e6	**2.127e7**	3.091e7	9.649e6	**2.127e7**
	DE/best/1/bin	2.873e7	8.224e6	2.157e7	5.466e7	3.309e7	2.693e7
	k-means	**2.127e7**	**4.718e2**	**2.127e7**	**2.127e7**	**1.03e3**	**2.127e7**
Iris	**DE/rand/1/bin**	**7.885e1**	**0**	**7.885e1**	**7.885e1**	**0**	**7.885e1**
	DE/best/1/bin	1.139e2	1.3e1	1.023e2	1.4e2	3.77e1	1.064e2
	k-means	8.321e1	1.629e1	**7.885e1**	1.455e2	6.665e1	7.886e1
Glass	**DE/rand/1/bin**	4.927e2	**7.466e1**	4.712e2	7.745e2	3.033e2	4.75e2
	DE/best/1/bin	4.933e2	1.106e2	4.255e2	8.388e2	4.133e2	4.539e2
	k-means	**4.182e2**	8.307e1	**3.363e2**	**5.818e2**	**2.455e2**	**4.11e2**
Ecoli	**DE/rand/1/bin**	1.897e1	1.303	1.625e1	2e1	3.75	1.994e1
	DE/best/1/bin	1.737e1	9.74e-1	1.542e1	2.02e1	4.78	1.738e1
	k-means	**1.536e1**	**9.54e-1**	**1.39e1**	**1.716e1**	**3.26**	**1.519e1**
Yeast	**DE/rand/1/bin**	5.937e1	5.622	5.076e1	7.054e1	1.978e1	5.82e1
	DE/best/1/bin	5.374e1	4.435	4.72e1	6.333e1	1.613e1	5.346e1
	k-means	**5.056e1**	**2.919**	**4.578e1**	**5.366e1**	**7.871**	**5.221e1**

Tablica 4.6 sugerira da su obje inačice algoritma DE postigli bolje vrijednosti DBI od algoritma k-means. U skupovima podataka s malo grupa kao što su *Iris* i *Cancer* razlika nije bila velika, ali u ostalim skupovima s većim brojem grupa inačice algoritma DE imaju puno bolji DBI od algoritma k-means. Također, kod skupova podataka s većim brojem grupa standardna devijacija je bila manja kod inačica algoritma DE, što ukazuje da su inačice algoritma DE stabilnije pri grupiranju podataka nego algoritam k-means kada je zadan veći broj grupa. Algoritam k-means je također za svaki skup podataka pronašao najgore rješenje (f_{max}). Kod inačica algoritma DE, algoritam DE/best/1/bin je imao blago bolje rezultate kod skupova s manjim brojem grupa (dva i

tri), a inačica algoritma DE/rand/1/bin je imala blago bolje rezultate kod skupova s većim brojem grupa (šest, sedam, deset). Najveća razlika je bila u skupu podataka *Ecoli*, gdje je inačica algoritma DE/rand/1/bin prosječni rezultat imala za 0.122 bolji nego inačica DE/best/1/bin. Kod ostalih skupova podataka razlike prosječnih rezultata između inačica DE su bile ispod 0.1. Kod skupova s većim brojem grupa kao što su *Ecoli* i *Glass* se isto može primijetiti da inačica DE/best/1/bin postiže najbolji rezultat (f_{min}) od svih algoritama, ali nije toliko stabilan kao DE/rand/1/bin te zbog toga u tim slučajevima ima lošije prosječne vrijednosti.

SSE rezultati u tablici 4.7 pokazuju da je tu algoritam k-means imao najbolju prosječnu vrijednost svugdje osim kod skupa podataka *Iris* gdje je inačica algoritma DE/rand/1/bin imala blago bolju prosječnu vrijednost. Standardna devijacija je bila slična kod svih skupova podataka osim kod skupa podataka *Cancer* gdje je algoritam k-means imao puno manju standardnu devijaciju od obje inačice algoritma DE. Razlog tomu je što su inačice algoritma DE za taj skup podataka postigle velike razlike u rezultatima SSE u ponavljanjima izvođenja.

Izvršeni algoritmi nad stvarnim skupovima podataka nam sugeriraju da bolje DBI rezultate daju inačice algoritma DE nego algoritam k-means, a bolje SSE rezultate ima algoritam k-means. To znači da kompaktnije grupe, i grupe koje su međusobno razdvojenije daju inačice algoritma DE, dok algoritam k-means daje grupe kojima su podatci unutar grupa nešto bliži svojim centroidima, ali grupe međusobno nisu toliko razdvojene. Razlog tomu je što algoritam k-means optimira SSE te prema njemu izabire centroide grupa, dok algoritam DE optimira DBI. Između inačica algoritma DE koje se razlikuju u operatoru mutacije nema značajnih razlika. Obje inačice daju slične vrijednosti DBI, i razlike u većini slučajeva su manje od 0.1. Najveće razlike se mogu uočiti kod stvarnih skupova podataka s više grupa (osam i deset), te se kod takvih skupova podataka blaga prednost može dati inačici DE/rand/1/bin.

5. ZAKLJUČAK

U diplomskom radu opisan je problem čvrstog grupiranja podataka. Opisan je algoritam k-means koji se najčešće koristi za grupiranje, algoritam DE i kako se on može primijeniti za grupiranje. Dan je opis nekoliko relativnih indeksa koji se mogu koristiti za vrednovanje učinkovitosti algoritama za grupiranje podataka. Implementirane su dvije inačice algoritma DE za grupiranje podataka koje se razlikuju u korištenom operatoru mutacije i ispitane su učinkovitosti tih inačica na nekoliko sintetičkih i stvarnih skupova podataka. Učinkovitosti tih inačica su uspoređene međusobno i s algoritmom k-means. Implementirana je jedna inačica algoritma DE s operatorom mutacije rand/1, a druga s operatorom mutacije best/1. Za algoritam k-means korištena je implementacija iz biblioteke scikit-learn. Mjere korištene za vrednovanje učinkovitosti su SSE i DBI. Eksperimentalna analiza je prikazala da na sintetičkim skupovima podataka obje inačice algoritma DE imaju bolje vrijednosti SSE i DBI od algoritma k-means. Na stvarnim skupovima podataka inačice algoritma DE imaju bolje vrijednosti DBI, dok algoritam k-means češće daje bolju vrijednost SSE. Ti rezultati su takvi jer inačice algoritma DE optimiraju DBI, dok algoritam k-means optimira SSE. Između inačica algoritma DE nema velikih razlika, obje inačice daju slične rezultate. Kod stvarnih skupova podataka, inačica algoritma s operatorom mutacije rand/1 daje blago bolju prosječnu vrijednost DBI kod skupova s više grupa, dok inačica s operatorom mutacije best/1 ima blago bolju prosječnu vrijednost DBI kod skupova s manje grupa. No razlike su zanemarive jer su u većini slučajeva manje od 0.1.

U budućem radu se programsko rješenje može unaprijediti tako da se omogući automatsko određivanje najboljeg broja grupa za određeni skup podataka, a ne da se broj grupa mora postaviti kao ulazna vrijednost. Također, mogu se analizirati učinkovitosti inačica algoritma DE s drugim ulaznim parametrima kao što su veličina populacije, stopa križanja i faktor skaliranja što bi moglo dovesti do drugačijih rezultata. Postoji i velik broj različitih operatora mutacije, te bi se u budućem radu moglo više njih implementirati i usporediti njihov utjecaj na učinkovitost algoritma DE.

LITERATURA

[1] R. Xu, D. C. Wunsch, Clustering, John Wiley & Sons Inc., New Jersey, 2008.

[2] A. E. Eiben, J. E. Smith, Introduction to Evolutionary Computing. Springer, 2015.

[3] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Survey, vol. 31, pp. 264–323, rujan 1999.

[4] S. Theodoridis, K. Koutroumbas, Pattern Recognition. Academic Press, 4th edition, 2008.

[5] L. Vendramin, R. J. G. B. Campello, E. R. Hruschka, Relative clustering validity criteria: A comparative overview, Statistical Anaysis and Data Mining, vol. 3, pp. 209–235, kolovoz 2010.

[6] E. Liberty, k-means clustering, Algorithms in Data Mining, jesen 2013, dostupno na: https://www.cs.yale.edu/homes/el327/datamining2013aFiles/10_k_means_clustering.pdf

[7] D. Bajer, Parameter control for differential evolution by storage of successful values at an individual level, Journal of Computational Science, vol. 68, 2023.

[9] D. Bajer, Adaptive k-tournament mutation scheme for differential evolution, Applied Soft Computing, vol. 85, 2019.

[9] S. Paterlini, T. Krink, Differential evolution and particle swarm optimisation in partitional clustering, Computational Statistics & Data Analysis, vol. 50, pp. 1220–1247, ožujak 2006.

[10] W. Kwedlo, A clustering method combining differential evolution with the k-means algorithm, Pattern Recognition Letters, vol. 32, pp. 1613–1621, rujan 2011.

[11] G. Martinović, D. Bajer, Data clustering with differential evolution incorporating macromutations, Swarm, Evolutionary, and Memetic Computing, vol. 8297 of Lecture Notes in Computer Science, pp. 158–169, Springer International Publishing, 2013.

[12] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Transactions on Systems, Man, and Cybernetics - Part A, vol. 38, pp. 218–237, siječanj 2008.

[13] C. H. Chou, M. C. Su, E. Lai, A new cluster validity measure and its application to image compression, Pattern Analysis and Applications, vol. 7, no. 2, pp. 205–220, srpanj 2004.

[14] R. F. Abdel-Kader, Genetically Improved PSO Algorithm for Efficient Data Clustering, 2010 Second International Conference on Machine Learning and Computing, pp. 71-75, Bangalore, India, 2010

[15] P. P. Win Cho, T. Thi Soe Nyunt, Data Clustering based on Differential Evolution with Modified Mutation Strategy, 2020 17th International Conference on Electrical

Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 222-225, Phuket, Thailand, 2020

[16] Biblioteka scikit-learn, korišteni algoritam k-means, dostupno na: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[17] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, vol. 11, pp. 341–359, prosinac 1997.

[18] R. Storn, On the usage of differential evolution for function optimization, Proceedings of North American Fuzzy Information Processing, pp. 519-523, Berkeley, CA, USA, 1996

SAŽETAK

U radu je opisan problem grupiranja podataka, popularni algoritam za grupiranje k-means te algoritam diferencijalne evolucije (DE), popularan evolucijski algoritam koji može biti primijenjen za grupiranje podataka. Implementirane su dvije inačice algoritma DE koje se razlikuju u korištenom operatoru mutacije, te je dan opis tog programskog rješenja, način rada i prikaz njegove uporabe. S implementiranim programskim rješenjem provedena je eksperimentalna analiza na nekoliko sintetičkih skupova podataka i na nekoliko često korištenih stvarnih skupova podataka. U analizi su uspoređeni algoritam k-means i dvije inačice algoritma DE. Vrednovane su njihove učinkovitosti za grupiranje podataka u smislu mjera SSE i DBI. Rezultati prikazuju da u većini slučajeva inačice algoritma DE pronalaze particije s boljim vrijednostima DBI, dok algoritam k-means pronalazi particije s boljim vrijednostima SSE. Razlike između inačica algoritma DE za korištene skupove podataka su zanemarive.

Ključne riječi: diferencijalna evolucija, grupiranje, k-means, mutacija, relativni indeksi

ABSTRACT

A differential evolution algorithm for data clustering

The paper describes the concept of data clustering, popular algorithm for clustering k-means and differential evolution (DE) algorithm which is a popular evolution algorithm that can be used for data clustering. There are two versions of DE algorithm implemented which differ in the mutation operator. A description is given of the implemented software solution, the way it works and how it is used. Using the implemented software solution, an experimental analysis was carried out on several synthetic datasets and on several often used real world datasets. This analysis compared the quality of partitions acquired by the k-means algorithm and the two versions of DE algorithm. The measures used for the quality of partitions were sum of squared errors (SSE) and Davies-Bouldin index (DBI). The analysis showed that both versions of DE algorithm got better DBI results, while k-means algorithm got better SSE results. The differences in results between the two versions of DE algorithm for the datasets used were negligible.

Keywords: differential evolution, clustering, k-means, mutation, relative cluster validity indices

ŽIVOTOPIS

Mihael Marjanović rođen je 9. kolovoza 1999. godine u Novoj Gradišci. Završio je Osnovnu školu Matije Antuna Relkovića u Davoru. 2014. godine upisuje Elektrotehničku i ekonomsku školu Nova Gradiška koju završava 2018. godine. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek. 2021. godine završava preddiplomski sveučilišni studij Računarstvo i stječe akademski naziv sveučilišni prvostupnik (baccalaureus) inženjer računarstva. Iste godine upisuje diplomski sveučilišni studij Računarstvo, izborni blok Programsko inženjerstvo.

PRILOZI

Prilog 1.

Sintetički podatci su podatci koji su umjetno generirani, najčešće od strane nekog računalnog algoritma. Koriste se za potvrđivanje matematičkih modela i za treniranje modela strojnog učenja. Sintetički skupovi podataka korišteni za eksperimentalnu analizu u ovom radu su dobiveni pomoću normalne razdiobe $\mathcal{N}(\mu,\sigma)$ tako što je oko zadanih centara ($\mu \equiv c \equiv \mathbb{R}^2$) stvoren odgovarajući broj normalno raspodijeljenih podataka sa zadanom standardnom devijacijom $\sigma \in \mathbb{R}^2$. Slijedi opis sintetičkih skupova podataka gdje su $g_i, \dots, g_k$ grupe sintetičkog skupa podataka, k je broj grupa za skup podataka, $|g_i|$ je kardinalnost ili broj elemenata i-te grupe, $c_i, \dots, c_k$ su centroidi grupa gdje je $c_i = (c_i^x, c_i^y)$ centroid i-te grupe i $\sigma_i = (\sigma_i^x, \sigma_i^y)$ je standardna devijacija normalne razdiobe i-te grupe.

Skup podataka data_k3_1 se sastoji od tri grupe g_1, g_2, g_3 gdje su kardinalnosti $|g_1| = |g_2| = |g_3| = 100$, centroidi $c_1 = (0.3, 0.4)$, $c_2 = (0.5, 0.2)$ i $c_3 = (0.5, 0.6)$, a standardna devijacija normalne razdiobe $\sigma_1 = \sigma_2 = \sigma_3 = (0.03, 0.03)$.

Skup podataka data_k4_3 se sastoji od četiri grupe g_1, g_2, g_3, g_4 gdje su kardinalnosti $|g_1| = 200, |g_2| = 100, |g_3| = 150, |g_4| = 75$, centroidi $c_1 = (0.3, 0.4)$, $c_2 = (0.1, 0.2)$, $c_3 = (0.5, 0.6)$ i $c_4 = (0.1, 0.5)$, a standardna devijacija normalne razdiobe $\sigma_1 = (0.06, 0.06), \sigma_2 = \sigma_3 = (0.05, 0.05)$ i $\sigma_4 = (0.04, 0.04)$.

Skup podataka data_k5_4 se sastoji od pet grupe g_1, g_2, g_3, g_4, g_5 gdje su kardinalnosti $|g_1| = 200$, $|g_2| = |g_3| = 100, |g_4| = 75, |g_5| = 150$, centroidi $c_1 = (0.5, 0.5)$, $c_2 = (0.3, 0.4)$, $c_3 = (0.5, 0.2), c_4 = (0.7, 0.5)$ i $c_5 = (0.3, 0.7)$, a standardna devijacija normalne razdiobe $\sigma_1 = \sigma_3 = \sigma_5 = (0.05, 0.05)$ i $\sigma_2 = \sigma_4 = (0.04, 0.04)$.

Skup podataka data_k6_1 se sastoji od šest grupe $g_1, g_2, g_3, g_4, g_5, g_6$ gdje su kardinalnosti $|g_1| = 75, |g_2| = |g_6| = 50, |g_3| = 150, |g_4| = |g_5| = 100$, centroidi $c_1 = (0.3, 0.3)$, $c_2 = (0.9, 0.7), c_3 = (0.6, 0.5), c_4 = (0.4, 0.8), c_5 = (0.5, 0.2)$ i $c_6 = (0.9, 0.5)$, a standardna devijacija normalne razdiobe $\sigma_1 = \sigma_2 = \sigma_4 = \sigma_6 = (0.02, 0.02), \sigma_3 = (0.07, 0.07)$ i $\sigma_5 = (0.04, 0.04)$.