

# Poboljšanje vrijednosti funkcije prilagođenosti genetskim algoritmima

Rava Filipović

student postdiplomskog magistarskog studija  
Slobomir P Univerzitet  
Bijeljina, BiH  
filipovicrava@yahoo.com

*Sadržaj*—U ovom radu ćemo se baviti grupisanjem datih  $N$  jedinki, osoba, ljudi itd. u definisan broj klastera  $K$  sa posebnim osvrtom na kvalitet vrijednosti funkcije prilagođenosti reprezentacije polaznih rješenja, što je implementirano u okviru rješavanja problema klasterovanja tako da se minimizira broj međjuklusterskih veza, u društvenim mrežama. Problem je NP-težak, tako da je gotovo nemoguće u razumnom vremenu naći optimalno rješenje. Za rješavanje ovog problema razvijeni su genetski algoritmi.

*Ključne riječi*—klasteri; optimizacija; heuristike;

## I. UVOD

Heuristike<sup>1</sup> su postale značajno, a najčešće i jedino, sredstvo u rješavanju realnih problema optimizacije<sup>2</sup>. Osnovni zahtjev je dobijanje rješenja bliskih optimalnom u razumnom vremenu. Klasterovanje<sup>3</sup>, kao jedna od aktivnosti istraživanja podataka, rješava optimizacione<sup>4</sup> probleme grupisanja podataka na osnovu nekih veza (relacija), koje postoje između njih. Ova tehnika istraživanja podataka otkriva objekte podataka sličnih osobina i dijeli ih u grupe (klaster), čineći ih preglednijim i korisnijim. Podjela u klaster doprinosi pronalaženju što veće količine informacija na osnovu analiziranih podataka.

Problem je NP-težak, tako da je gotovo nemoguće u razumnom vremenu naći optimalno rješenje. Ovaj rad predstavlja dio rješenja, u okviru rješavanja proučavanog problema podjele u klaster tako da se minimizira broj međjuklusterskih veza. Njegov cilj je da se razvije efektivna i efikasna metoda za raspoređivanje objekata u klaster, zatim da se kroz niz iteracija vrši popravka trenutno najboljeg rješenja, tj. transformacija polaznih rješenja. Za rješavanje ovog dijela proučavanog problema podjele u klaster implementirane su konstruktivne heuristike (pohlepni

algoritam-Greedy metode) i metaheuristike (genetski algoritmi-Genetic Algorithm).

## II. NEŠTO O HEURISTIKAMA

### A. Pohlepni algoritam (Greedy algorithm)

Jedna od konstruktivnih heuristika je pohlepna (Greedy) metoda. Konstruktivne heuristike generišu samo jedno dopustivo rješenje problema. Završavaju kada je konstruisano cjelovito rješenje. Pohlepni algoritmi su dosta brzi, jednostavni, koriste se za online probleme i daju dobre rezultate. Pohlepni algoritam koristi heuristiku za rješavanje problema, takvu da u svakom koraku bira lokalno najbolje rešenje, sa intuicijom da će tako naći globalni optimum.

### B. Genetski algoritam (Genetic algorithm)

Genetski algoritam<sup>5</sup> predstavlja metaheuristiku<sup>6</sup> koja koristi veći broj rješenja (populaciju) i bazirana je na principima prirodne evolucije. Metodu je prvi predložio J. Holland [1], dok je detaljan opis dat u [2]. Osnovne ideje metode mogu se naći i u [3], [4] i [5]. Genetski algoritmi iterativno vrše modifikacije polaznih rješenja po nekim, unaprijed zadatim, intuitivnim pravilima ili kombinovanje dva (ili više) rješenja u cilju generisanja novih i nalaženja među njima boljih od trenutno najboljeg rješenja. To je proces koji treba da simulira reprodukciju živih organizama u prirodi. Rješenja, koja su članovi neke populacije miješaju se i proizvode potomke koji bi trebalo da zadržavaju dobre osobine svojih predaka.

## III. RJEŠENJE PROBLEMA

### A. Definisani problem optimizacije

Proučavan problem klasterovanja se može definisati na sljedeći način: Dat je skup podataka od  $N$  osoba i matrica odnosa relacija prijateljstava između osoba. Skup podataka treba rasporediti u željeni broj klastera  $K$  tako da se

<sup>1</sup> Heuristika je nauka o mogućnostima naučnog istraživanja, i odgovara na pitanja; šta se može naučno istražiti? šta se ne može? šta je aksiom? (grčki: "Εὐρίσκειν", "naći" ili "otkriti").

<sup>2</sup> Optimizacija ili matematičko programiranje je grana matematike koja proučava maksimiziranje i minimiziranje funkcija.

<sup>3</sup> "Klastering" (Clustering): Klasterovanje se odnosi na grupisanje podataka po sličnosti.

<sup>4</sup> Optimizacija ili matematičko programiranje je grana matematike koja proučava maksimiziranje i minimiziranje funkcija.

<sup>5</sup> Genetski algoritam je tehnika pretraživanja korištena u računarstvu za pronalazak tačnih ili približno tačnih rješenja za probleme optimizacije i pretraživanja.

<sup>6</sup> Metaheuristika dolazi od grčke riječi „meta“ kao „metodologija višeg sloja“. Naime, metaheuristike su postupci za rješavanje optimizacionih problema koje predviđaju (propisuju) neki niz koraka.

minimizira broj medjuklusterskih veza, čiji krajevi su u različitim klasterima.

Elementi matrice koji imaju vrijednost jedan odgovaraju parovima osoba koji jesu prijatelji, dok nule odgovaraju parovima osoba koje nisu prijatelji. Ako neke dvije osobe jesu prijatelji onda su one povezane ivicom. Cilj istraživanja ovog proučavanog problema je razvoj metode za podjelu skupa osoba u klaster tako da broj ivica, čiji su krajevi u različitim klasterima, bude minimalan.

### B. Prikaz hromozoma (jedinke)

Svako pojedinačno rješenje optimizacionog problema predstavlja jedinku (*hromozom*). Rješenje može biti reprezentativno pomoću izabranog koda. Za proučavani problem, odabran je izgled hromozoma Sl. 1 prikazan cijelim brojevima. Svaki od simbola u kodu naziva se gen (jedinčna informacija). Jedinka-skup gena, predstavlja jedno rješenje načina smještanja objekata u klaster.

### C. Dobrota jedinke i funkcija dobrote

Broj medjuklusterskih veza je kriterij na osnovu koga svakom rješenju pridružujemo vrijednost funkcije prilagođenosti (dobrote). Vrijednost funkcije pokazuje efektivnost izvršenja zadatka radi postizanja cilja i naziva se kriterijum, kriterijumska funkcija, funkcija cilja ili indeks ili mjera performanse. Funkcija cilja (*cost* funkcija, funkcija dobrote) je *mjera kvaliteta pojedinog rješenja* (jedinke) i zavisi od problema koji se rješava.

Funkcija dobrote ili funkcija ocjene kvaliteta jedinke se u literaturi još naziva *fitness* funkcija, funkcija prilagođenosti i u najjednostavnijoj interpretaciji ekvivalent je funkciji koju treba optimizovati. Što je dobrota jedinke veća, jedinka ima manju vjerovatnoću preživljavanja, suprotno jedinka ima veću vjerovatnoću preživljavanja. Za proučavan problem, minimalan broj medjuklusterskih veza je ključ odabira kvalitetnijih jedinaka.

Dobrota neke jedinke govori o tome koliko je pojedina jedinka dobra, odnosno - u kojoj mjeri jedinka ispunjava svoju svrhu. Na primjer, ako jedinka predstavlja rješenje problema optimizacije rasporeda instance N osoba u K klastera, primjerena dobrota jedinke bi trebala odražavati opisanu funkciju cilja. Na taj se način za jedinku koja ima veliku dobrotu može reći da ona uistinu ne predstavlja rješenje, koje dobro raspoređuje instancu N osoba u željen broj klastera K, s obzirom na postavljeni kriterijum minimalnog broja medjuklaster veza. Dok, za jedinku koja ima malu dobrotu može se reći da ona predstavlja rješenje, koje dobro raspoređuje instancu N osoba u željen broj klastera K.

### D. Pohlepni postupak klasterovanja

Za proučavani problem, na početku algoritma slučajno generišemo jednu permutaciju cijelih brojeva bez ponavljanja



Slika 1. Primjer jedne permutacije brojeva instance 30 osoba

od 1 do N (gdje je N broj osoba). Pohlepni algoritmi uvijek kreću od praznog rješenja, gdje jednu slučajno generisanu permutaciju cijelih brojeva stavljamo u jedan klaster. Praznim rješenjem funkcija cilja nije zadovoljena. Zato, prvu kombinaciju permutovanih osoba (Permutacija osoba: 23 22 29 4 12 19 13 24 30 8 28 9 21 7 17 2 20 18 15 1 16 27 10 14 25 26 3 11 5 6 ) smještamo u klaster i to koliko osoba toliko klastera. Znači, u početku svaka osoba se stavlja u jedan prazan klaster  $C_1=\{23\}$ ,  $C_2=\{22\}$ ,  $C_3=\{29\}$ , ...,  $C_n=\{6\}$ , a u svakom narednom koraku se pohlepno sakuplja par klastera (prvi i zadnji klaster)  $C_1UC_n=\{23,6\}$ , pa (drugi i predzadnji klaster)  $C_2UC_{n-1}=\{22,5\}$ , ... Sakupljanje klastera prestaje onda kada se dobije K klastera (definisani ulazni parametar). Ovim dobijamo jedno moguće (pojedinačno) rješenje problema podjele instance N osoba u željen broj klastera K – *jednu jedinku* (23 6 16 17 28 24 10 19 25 2 20 26 22 5 1 7 8 13 14 12 29 11 15 21 30 27 4 3 18 9), kojoj je pridružen određen broj medjuklusterskih veza i broj veza unutar klastera:

Cluster[0]= 23 6 16 17 28 24 10 19 25 2 20 26

Cluster[1]= 22 5 1 7 8 13 14 12

Cluster[2]= 29 11 15 21 30 27

Cluster[3]= 4 3 18 9

Broj medjuklusterskih prijateljstava je 102

Broj prijateljstava u Cluster[0] = 30

Broj prijateljstava u Cluster[1] = 6

Broj prijateljstava u Cluster[2] = 2

Broj prijateljstava u Cluster[3] = 1

Vrijednost funkcije dobrote, pridružena dobijenoj jedinki, iznosi 102 medjuklusterske veze. Dobijeno rješenje, sa ovim brojem medjuklusterskih veza, predstavlja početno rješenje kome će se nastojati iterativno poboljšati kvalitet vrijednosti funkcije prilagođenosti genetskim algoritmima. Početno rješenje problema podjele instance N osoba u željen broj klastera K, njemu pridružen odgovarajući broj medjuklusterskih veza i broj veza prijateljstava unutar svakog klastera se pamte i postaju tekući.

### E. Poboljšanje kvaliteta vrijednosti funkcije prilagođenosti

U cilju minimiziranja broja relacija između klastera, odnosno poboljšanja kvaliteta vrijednosti funkcije prilagođenosti početnog rješenja, razvijeni su genetski algoritmi. Za svaki par klastera:

- Premjestimo jednu ili više osoba ponaosob iz jednog u drugi klaster, pri čemu svakom pojedinačnom rješenju pridružujemo broj medjuklusterskih veza i broj veza unutar klastera. Ako je broj medjuklusterskih veza pojedinačnog rješenja manji od broja medjuklusterskih veza trenutno najboljeg rješenja premještena osoba ostaje u klasteru, novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. Ako premještena osoba ne donosi poboljšanje kvaliteta funkcije prilagođenosti onda je vraćamo u klaster odakle je premještena, tekuće rješenje i njemu dodijeljena vrijednost funkcije dobrote ostaju nepromijenjeni.
- Premjestimo jednu ili više osoba ponaosob iz drugog u prvi klaster dodjeljujući svakom pojedinačnom rješenju broj medjuklusterskih veza i broj veza unutar klastera. Ako je broj medjuklusterskih veza

pojedinačnog rješenja manji od broja medjuklusterskih veza trenutno najboljeg rješenja premještena osoba ostaje u klasteru, novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. U suprotnom tu osobu vraćam u klaster odakle je premještena, tekuće rješenje i njemu dodijeljena vrijednost funkcije dobrote ostaju nepromijenjeni.

- Mijenjamo osobu po osobu iz jednog klastera sa svakom osobom iz drugog klastera, pri čemu pridružujemo broj medjuklusterskih veza i broj veza unutar klastera svakom pojedinačnom rješenju. Ako je broj medjuklusterskih veza pojedinačnog rješenja manji od broja medjuklusterskih veza trenutno najboljeg rješenja te osobe ostaju u premještenim klasterima, novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. Ako premještene osobe ne donose poboljšanje kvaliteta funkcije cilja onda te osobe vraćamo u klastere odakle su premještene, tekuće rješenje i njemu dodijeljena vrijednost funkcije dobrote ostaju nepromijenjeni.

Za prethodno opisane postupke, poboljšanja kvaliteta vrijednosti funkcije prilagođenosti početnog rješenja genetskim algoritimima, intuitivno je postavljen dodatni uslov da svaki klaster mora imati bar  $N/(2*K)$  osoba i ne više od  $(2*N)/K$  osoba. Razvijeni postupci za pronalaženje rješenja i poboljšanje (njima pridružene) funkcije dobrote, uz intuitivno postavljene kriterije minimalnog i maksimalnog broja osoba po klasterima, su se pokazali efikasnim.

Sljedeći primjer ilustruje poboljšanje kvaliteta vrijednosti funkcije prilagođenosti početnog rješenja problema podjele u klaster, tako što se razmještanjem instance  $N$  osoba u željen broj klastera  $K$  smanjuje, odnosno minimizira broj medjuklusterskih veza (prethodno opisanim) genetskim algoritimima. Svako pojedinačno rješenje je predstavljeno distribucijom gena (osoba) po klasterima, pri čemu im je pridružen odgovarajući broj medjuklusterskih prijateljstava, kao i broj prijateljstava unutar klastera:

```
Cluster[0]= 23 6 16 17 28 24 10 19 25 2 20 26
Cluster[1]= 22 5 1 7 8 13 14 12
Cluster[2]= 29 11 15 21 30 27
Cluster[3]= 4 3 18 9
Broj medjuklusterskih prijateljstava je 102
Broj prijateljstava u Cluster[0] = 30
Broj prijateljstava u Cluster[1] = 6
Broj prijateljstava u Cluster[2] = 2
Broj prijateljstava u Cluster[3] = 1
```

```
Cluster[0]= 23 6 16 17 28 24 20 26 2 25 19
Cluster[1]= 22 5 1 7 8 13 14 12 10
Cluster[2]= 29 11 15 21 30 27
Cluster[3]= 4 3 18 9
Broj medjuklusterskih prijateljstava je 101
Broj prijateljstava u Cluster[0] = 29
Broj prijateljstava u Cluster[1] = 8
Broj prijateljstava u Cluster[2] = 2
Broj prijateljstava u Cluster[3] = 1
```

```
Cluster[0]= 20 26 2 25 19 24 28 17 16 23 7
Cluster[1]= 22 5 1 6 10 12 14 13 8
Cluster[2]= 29 11 15 21 30 27
```

```
Cluster[3]= 4 3 18 9
Broj medjuklusterskih prijateljstava je 92
Broj prijateljstava u Cluster[0] = 35
Broj prijateljstava u Cluster[1] = 11
Broj prijateljstava u Cluster[2] = 2
Broj prijateljstava u Cluster[3] = 1
...
```

```
Cluster[0]= 20 26 27 25 19 24 28 17 16 23 7 18 11 3 15
Cluster[1]= 5 8 13 14 12 10 6
Cluster[2]= 29 2 21 30
Cluster[3]= 9 1 4 22
Broj medjuklusterskih prijateljstava je 59
Broj prijateljstava u Cluster[0] = 71
Broj prijateljstava u Cluster[1] = 8
Broj prijateljstava u Cluster[2] = 2
Broj prijateljstava u Cluster[3] = 1
```

```
Cluster[0]= 20 26 27 25 19 24 18 11 3 15 7 23 16 17 1
Cluster[1]= 5 8 13 14 12 10 6
Cluster[2]= 29 2 21 30
Cluster[3]= 9 28 22 4
Broj medjuklusterskih prijateljstava je 54
Broj prijateljstava u Cluster[0] = 74
Broj prijateljstava u Cluster[1] = 8
Broj prijateljstava u Cluster[2] = 2
Broj prijateljstava u Cluster[3] = 3
...
```

```
Cluster[0]= 18 11 3 15 7 23 16 17 1 24 19 25 9 28 20
Cluster[1]= 6 26 2 5 14 27
Cluster[2]= 29 21 30 10 13 8
Cluster[3]= 4 22 12
Broj medjuklusterskih prijateljstava je 41
Broj prijateljstava u Cluster[0] = 76
Broj prijateljstava u Cluster[1] = 15
Broj prijateljstava u Cluster[2] = 9
Broj prijateljstava u Cluster[3] = 0
```

```
Cluster[0]= 18 11 3 15 7 23 16 17 1 24 19 25 9 28 20
Cluster[1]= 26 2 5 14 27 6
Cluster[2]= 8 13 10 30 21 4
Cluster[3]= 12 22 29
Broj medjuklusterskih prijateljstava je 39
Broj prijateljstava u Cluster[0] = 76
Broj prijateljstava u Cluster[1] = 15
Broj prijateljstava u Cluster[2] = 9
Broj prijateljstava u Cluster[3] = 2
```

```
Cluster[0]= 18 11 3 15 7 23 16 17 1 24 19 25 9 28 20
Cluster[1]= 26 2 5 14 27 6
Cluster[2]= 8 13 10 30 4 29
Cluster[3]= 12 22 21
Broj medjuklusterskih prijateljstava je 37
Broj prijateljstava u Cluster[0] = 76
Broj prijateljstava u Cluster[1] = 15
Broj prijateljstava u Cluster[2] = 11
Broj prijateljstava u Cluster[3] = 2
```

```
Hromozom[0]= 18 11 3 15 7 23 16 17 1 24 19 25 9 28 20 || 26 2 5 14 27 6 || 8
13 10 30 4 29 || 12 22 21 ||
```

Razvojem genetskih algoritama, vrijednost funkcije dobrote početnog rješenja je optimizovana sa 102 na 37 medjuklusterske veze, pri čemu dobijeno rješenje, problema podjele u klaster, predstavlja cjelovito rješenje (Hromozom[0]= 18 11 3 15 7 23 16 17 1 24 19 25 9 28 20 || 26 2 5 14 27 6 || 8 13 10 30 4 29 || 12 22 21 || ).

Prethodno opisane postupke algoritma ponavljamo od početka, sve dok se ne generiše  $N$  jedinki. Ovim dobijamo skup jedinki koje čine početnu populaciju, odnosno skup mogućih rješenja kojima je dodijeljena odgovarajuća vrijednost funkcije cilja, u okviru rješavanja problema klasterovanja tako da se minimizira broj medjuklusterskih veza. Veličina populacije je definisana instancom  $N$  osoba. U osnovi genetskih algoritama su generacije<sup>7</sup> i populacije<sup>8</sup>. Primjenom genetskih operatora (selekcije, ukrštanja i mutacije) polazna populacija evoluirala kroz generacije, kreirajući u svakoj narednoj generaciji populaciju od  $N$  kvalitetnijih jedinki.

#### F. Doprinos problema podjele u klustere

U mnogim životnim oblastima, istraživači se često sreću sa situacijama koje su najbolje riješene definisanjem grupa homogenih objekata bilo da su individue, firme, proizvodi ili čak njihova ponašanja. Strateška rješenja bazirana na identifikaciji grupa unutar populacije pomažu u donošenju važnih odluka. Ova ista potreba se sreće u različitim područjima, od fizičkih do društvenih nauka.

Kako su društvene mreže u ekspanziji i kako postaju sve veće i veće nameće se potreba za efikasnim postupcima kojima bi se članovi tih mreža efikasno razvrstali u grupe. Takve grupe sadrže elemente sličnih svojstava, pa generalno gledajući svojstva grupe mogu biti definisana pomoću svojstava elemenata koji tu grupu čine. Klasterovanje služi za ispitivanje osobina posmatranih podataka i omogućava da više objekata analiziramo istovremeno ili zajedno. Particionisanjem konačnog skupa podataka dobiju se klasteri tako što su međusobno najbližnji objekti grupisani u isti klaster, a objekti koji se međusobno znatno razlikuju se nalaze u različitim klasterima.

U našem slučaju, uz unaprijed definisane odnose prijateljstva i funkcije cilja intuitivno smo stvorili grupe (klustere), koje predstavljaju stvarne grupe ljudi u stvarnom životu. Pri čemu je moguće pojednostaviti opažanje na način da se svojstva pojedinačnih objekata zamijene opštim svojstvom klastera kojem objekti pripadaju. Zbog značaja praktične primjene proučavanog problema, u ovom radu ćemo se držati pretpostavke da su objekti ili elementi, koji se grupišu zasebni entiteti istraživanja u najširem smislu. Prema tome, objekti mogu biti osobe (u našem slučaju), škole, preduzeća, gradovi, životinje i sl.

Najpoznatiji način korišćenja postupka klasterovanja je u istraživačke svrhe i to za formiranje principa, metoda i pravila grupisanja skupa podataka. Cilj istraživanja podataka, tehnikom klasterovanja, je pronalaženje modela koji će najbolje opisati podatke sa kojima se radi. Postupak klasterovanja, kao aktivnost u okviru istraživanja podataka, daje odgovore na postavljena pitanja za čije je rješavanje u tradicionalnom pristupu trebalo mnogo više vremena i vještina. Postupak podjele skupa objekata u grupe omogućava analizu velike količine podataka u cilju otkrivanja

neočekivanih veza među podacima ili u cilju sumiranja na neki nov način.

Analizu velike količine podataka je lakše izvesti nad grupisanim podacima u klustere, jer se time omogućava: pojednostavljenje skupa podataka, načina posmatranja, proučavanje postupaka, principa grupisanja, identifikacija odnosa, ... Sa definisanim klasterima i osnovnom strukturom podataka u njima, istraživač objašnjava odnos između posmatranja koje nije bilo moguće sa individualnim posmatranjem. Grupisanje skupa podataka u manje grupe (klustere) predstavlja izazov u otkrivanju pravila, modela i zakonitosti na osnovu kojih mogu da se donesu važne odluke. Ono omogućava bolje razumijevanje raščlanjivanjem velikog skupa podataka na manje cjeline (klustere), čime se jedostavnije može upravljati velikim količinama podataka.

Ideja ovog rada je bila da se prvo vidi priprema za genetske algoritme, stvaranje početnih rješenja, kojima se pridružuju vrijednosti funkcije cilja. Kao finalni proizvod nastala je približna metoda za klasterovanje društvenih mreža, tj. grupisanje članova neke društvene mreže u nekoliko manjih grupa (klastera), na osnovu njihovih veza. Svako potencijalno rješenje optimizacionog problema smještanja osoba u klustere se zamišlja kao genetski kod neke jedinke. Instancu  $N$  osoba smo rasporedili u željen broj klastera  $K$  tako da ukupan broj parova osoba, koje jesu prijatelji, ali se nalaze u različitim klasterima (medjuklustersko prijateljstvo) bude što manji. Simuliranjem evolucijskog procesa nad tim početnim rješenjima tražimo što bolje (poželjno najbolje) rješenje.

Težište ovog rada je bilo stavljeno na poboljšanje vrijednosti funkcije prilagođenosti, početnih rješenja. Kroz niz iterativnih koraka, vrijednost funkcije dobrote početnih rješenja je poboljšana genetskim algoritmima. Razvoj metoda za formiranje inicijalnih rješenja, tj. formiranje početne populacije za genetski algoritam predstavlja krajnji rezultat ovog dijela istraživanja, u okviru rješavanja problema podjele u klustere tako da se minimizira broj medjuklusterskih veza, u društvenim mrežama. Osnova genetskog algoritma je bazirana na traženju najbolje i najprilagodljivije jedinke, uz date uslove.

Kao rezultat istraživanja ovog dijela proučavanog problema klasterovanja predstavljaju osmišljene i implementirane heuristike, koje su doprinijele raznovrsnosti reprezentacije rješenja i poboljšanje vrijednosti funkcije prilagođenosti. Razvijene heuristike, za proučavani problem, su zadovoljile sljedeće zahtjeve. Da bi bile razumljive, za onoga koji ih koristi, razvijene heuristike su jednostavne. Njih odlikuje robusnost, odnosno one ne mijenjaju drastično svoje ponašanje za male promjene parametara. One posjeduju mogućnost da generišu veći broj rješenja, kako bi korisnik mogao izabrati najprihvatljivije od njih, u odnosu na postavljen kriterij minimalnog broja medjuklusterskih veza. Takođe imaju mogućnost interaktivnog rada, gdje korisnik može interaktivno da utiče, u većoj ili manjoj mjeri, na proces dobijanja rješenja donoseći neke bitne odluke u nekim koracima heuristike.

Svrha razvoja genetskih algoritama jeste da se pomoću njih rješava problem minimiziranja broja medjuklusterskih

<sup>7</sup> Generacija – skup jedinki.

<sup>8</sup> Populacija – skup jedinki u  $i$ -tom koraku algoritma.

veza, odnosno kako nije uvijek moguće minimizirati, onda pronaći što je moguće bolje rješenje. Optimizacija proučavanog problema podjele u klastere, tako da se minimizira broj medjuklusterskih veza, je šire primjenljiva u praktične svrhe. Tako isti programski kod se može lako prilagoditi različitim primjenama u svim oblastima ljudskih aktivnosti (biologije, anatomije, hemije, fizike, lingvistike, ekonomije, računarstva, sporta, muzike, ...).

#### ZAKLJUČAK

U raznim vrstama istraživanja istraživači se susreću s potrebom grupisanja ili izdvajanja sličnih objekata iz neke veće cjeline u manje grupe, čime se pojednostavljuje analiza podataka. Iz te činjenice proizlazi važnost klasterovanja. Dobijeni rezultati, postupka podjele u klastere, pružaju mogućnost da se na odgovarajući način interpretiraju za šta je neophodno dovoljno poznavati teorijsku pozadinu istraživanja problema, kako bi se dobijeni rezultati mogli iskoristiti za pronalaženje novih saznanja i spoznaja.

Opisan razvoj genetskih algoritama dao je efikasna i efektivna rješenja, tako što razmjenom genetskog materijala između klastera, dobro prilagođene jedinke generišu još bolje prilagođene jedinke. Isto tako, slabije prilagođene jedinke sa nekim dobro prilagođenim genima dobijaju svoju šansu da rekombinacijom dobrih gena proizvedu dobro prilagođene jedinke. Kvalitet vrijednosti funkcije prilagođenosti polaznih rješenja je poboljšan i formirana su inicijalna rješenja tj. formirana je početna populacija. Ovim ne samo da su poboljšane vrijednosti funkcije dobrote jedinki, već je i povećan kvalitet početne populacije. Efikasnost podrazumijeva dobijanje odgovarajuće podjele osoba u manje grupe (klastere) u što kraćem vremenu. Efektivnost se odnosi na kvalitet vrijednosti funkcije dobrote, u pogledu odgovarajuće podjele instance  $N$  osoba u definisan broj klastera  $K$  sa što manjim brojem medjuklusterskih veza.

Proučavan problem je razvijen u C# i testiran na primjerima iz literature, realnog živora i primjerima generisanim na slučajan način. Primjeri primjene su svuda oko nas (a prije svega u socijalnim mrežama), ali predviđeni prostor za rad ne dozvoljava kompletan prikaz nekog praktičnog primjera.

Koncept genetskih algoritama se sastoji i od primjene operatora (selekcije, mutacije i ukrštanja) na početnu populaciju jedinki, u cilju poboljšanja vrijednosti funkcije

dobrote, što će biti izloženo u nekim od sljedećih radova, kao nastavku istraživanja proučavanog problema klasterovanja tako da se minimizira broj medjuklusterskih veza, u društvenim mrežama.

#### ZAHVALNICA

Ovim putem zahvalila bih se mentoru dr Draganu Uroševiću vanrednom profesoru na Fakultetu Informacionih Tehnologija SPU na vođstvu, strpljenju i entuzijazmu koje je pokazao tokom čitavog procesa stvaranja ovog rada, kao dijela mog magistarskog rada.

#### LITERATURA

- [1] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.
- [2] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publ. Comp., Inc., 1989.
- [3] D. Cvetković, M. Čangalović, Đ. Dugošija, V. Kovačević-Vujičić, S. Simić, and J. Vuleta. *Kombinatorna optimizacija (Matematička teorija i algoritmi)*. DOPIS, Beograd, 1996.
- [4] Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. IPL: Information Processing Letters 76, 175–181 (2000)
- [5] Jörg Meyer-Stamer: *Clustering and the Creation of an Innovation*, Duisburg, 2002.
- [6] H. Muhlenbein. Genetic algorithms. In E. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial optimization*, pages 137–171. John Wiley & Sons Ltd., 1997.

#### ABSTRACT

In this paper, we address the clustering of given  $N$  individuals, persons, people, etc. the number of clusters  $K$  is defined with special emphasis on the quality of the value of fitness function representation of the starting solution, which is implemented in the framework of solving the problem of clustering to minimize the number of connections between clusters in social networks. The problem is NP-hard, so it is almost impossible within a reasonable time to find an optimal solution. To solve this problem, genetic algorithms were developed.

#### IMPROVING THE VALUE OF FITNESS FUNCTION FOR GENETIC ALGORITHMS

Rava Filipović