

Implementacija heuristika na istoj reprezentaciji rješenja

Rava Filipović

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

Sadržaj—U ovom radu je opisano poboljšanje vrijednosti funkcije dobrote jedinki inicijalne populacije integracijom genetskih algoritama (primjena genetskog operatora ukrštanja) i lokalnog pretraživanja, bazirano na istoj reprezentaciji rješenja i implementirano u okviru rješavanja problema podjele na klastere tako da se minimizira broj medjuklusterskih veza, u društvenim mrežama.

Ključne riječi—klasteri; optimizacija; heuristike;

I. UVOD

Osnovna ideja rješavanja ovog optimizacionog¹ problema klasterovanja² je bila da se poveća efikasnost i efektivnost primjene integracije metaheurističkih³ metoda i to uvođenjem dodatnih parametara, smanjenjem okolina za pretraživanje i hibridizacija metoda. Cilj ovog opisanog dijela proučavanog problema podjele na klastere⁴, tako da se minimizira broj medjuklusterskih veza u društvenim⁵ mrežama, je poboljšanje vrijednosti funkcije dobrote kombinacijom genetskih⁶ algoritama (genetskog operatora ukrštanja) i lokalnog pretraživanja, na istoj reprezentaciji rješenja. Takvi algoritmi se onda zovu hibridni genetski algoritmi (Hybrid Genetic Algorithm, HGA), koji su zasnovani na primijeni hibridizacije genetskih algoritama heuristikama lokalnog pretraživanja.

¹ Optimizacija ili matematičko programiranje je grana matematike koja proučava maksimiziranje i minimiziranje funkcija.

² "Klastering" (Clustering): Klasterovanje se odnosi na grupisanje podataka po sličnosti.

³ *Metaheuristika* dolazi od grčke riječi „meta“ kao „metodologija višeg sloja“. Naime, metaheuristike su postupci za rješavanje optimizacionih problema koje predviđaju (propisuju) neki niz koraka.

⁴ Klaster: grupa istih ili sličnih elemenata grupisanih blizu jedni drugih – gomila.

⁵ Društvena mreža (takođe i socijalna mreža) je društvena struktura sastavljena od pojedinaca (ili organizacija) koji se nazivaju "čvorovi", a koji su povezani jednim ili više specifičnih tipova „veza“, kao što su: prijateljstvo, vizije, ideje, finansijski interes, srodstvo, zajednički interes, finansijska razmena, nedopadanje, ili odnosi poverenja, znanja ili prestiža.

⁶ Genetski algoritam je tehnika pretraživanja korištena u računarstvu za pronalazak tačnih ili približno tačnih rješenja za probleme optimizacije i pretraživanja.

II. INTEGRACIJA HEURISTIKA

A. Genetski algoritmi

Heuristike su namijenjene za rješavanje raznih optimizacionih problema. Prilagođene su svojstvima i specifičnostima problema, koji se rješava. Razlika među heurističkim algoritmima je u tome na koji način obavljaju to pretraživanje. Upravo ta razlika omogućava kombinaciju heuristika u rješavanju optimizacionih problema. *Genetski algoritam* predstavlja metaheuristiku 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 poboljšanje početnog rješenja po nekim, unaprijed zadatim, 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 mješaju se i proizvode potomke koji bi trebalo da zadržavaju dobre osobine svojih predaka. Ovaj proces može se zamisliti kao lokalno pretraživanje u generalisanom smislu, pretražuje se okolina ne jednog rješenja, nego više rješenja iz populacije (ili čak svih rješenja iz populacije).

B. Lokalno pretraživanje

Lokalno pretraživanje (Local search, LS) podrazumijeva da se za svako novo rješenje, u okolini nekog početnog rješenja i njemu pridružene vrijednosti funkcije dobrote, dodjeljuje vrijednost funkcije dobrote. Ukoliko je vrijednost funkcije dobrote novog rješenja manja od vrijednosti funkcije dobrote početnog rješenja čuvaju se i pamte novo rješenje i njemu pridružena vrijednost funkcije dobrote. Kada se u okolini početnog rješenja obidu sva rješenja, pretraživanje se nastavlja u okolini novog rješenja. Pretraživanje okoline je ispitivanje svih susjeda nekog rješenja. Proces lokalnog pretraživanja odvija se u iteracijama i zaustavlja kada u okolini novog rješenja ne postoji bolje rješenje, što je i osnovni nedostatak lokalnog pretraživanja. Metaheuristike koriste eksplicitno ili implicitno ovu strategiju lokalnog pretraživanja, tako da raznim tehnikama prevazilaze zaglavljivanje u lokalnim minimumima.

III. RJEŠAVANJE PROBLEMA OPTIMIZACIJE

A. Definisani problem klasterovanja

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

B. Poboljšanje kvaliteta jedinki inicijalne populacije

Početnu populaciju čini skup rješenja (hromozoma, jedinki). Jedinka je jedno moguće rješenje problema podjele instance N osoba u K klastera tako da se minimizira broj medjuklusterskih veza, pri čemu joj je dodijeljena vrijednost funkcije prilagođenosti, koja ocjenjuje njen kvalitet (dobrotu). Veličina populacije je definisana instancom N osoba.

Primjenom genetskih operatora (selekcije, ukrštanja i mutacije) i metode lokalnog pretraživanja, polazna populacija evoluirala kroz generacije⁷ kreirajući u svakoj narednoj generaciji populaciju⁸ od N kvalitetnijih jedinki. Proces evolucije odvija se tako što se u svakoj generaciji odredi kvalitet svakog pojedinačnog rješenja (jedinke) pridruživanjem vrijednosti funkcije prilagođenosti. U procesu izvršavanja genetskih algoritama, lokalno pretraživanje se pojavljuje na više nivoa. Drugim riječima, jedna heuristika (Local Search) kontrolira izvršavanje druge heuristike (Genetic Algorithm) u rješavanju problema podjele na klastera, tako da se minimizira broj medjuklusterskih veza u društvenim mrežama.

U procesu ukrštanja učestvuju dvije jedinke, koje se nazivaju roditelji. Operator ukrštanja vrši razmjenu gena jedinki, čime doprinosi raznovrsnosti genetskog materijala. Ukrštanje je binarni operator i reprodukuje nove jedinke, koje se nazivaju potomci. Najvažnija karakteristika ukrštanja jest da potomci nasljeđuju svojstva svojih roditelja. Ako su roditelji dobri tada će najvjerojatnije i potomci biti dobri, ako ne i bolji od svojih roditelja. Za proučavani problem, primijenjen je mehanizam ukrštanja sa jednom, dvije, tri ili više tačaka presjeka, za svaki par hromozoma. Na slučajan način odaberemo tačke prekida na paru hromozoma (roditelja) te napravimo razmjenu roditeljskih hromozoma između tih tačaka i na taj način dobijamo nove jedinke (potomke). Ako analiziramo distribuciju gena po klasterima potomaka, uočavamo da su se izgubili pojedini geni, od sadržaja osnovnih gena instance osoba N i da se pojedini geni ponavljaju. Zadatak operatora ukrštanja je da obnovi izgubljeni genetski materijal potomaka. Geni koji se ponavljaju mijenjamo genima koji nedostaju. Popravljenom prvom i drugom potomku, pridružujemo vrijednost funkcije dobrote (broj medjuklusterskih veza) i broj veza unutar svakog klastera.

Operator ukrštanja daje mogućnost da, razmjenom genetskog materijala, 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.

Sada treba da vidimo da li su (dalje u tekstu popravljene) potomci, postupka ukrštanja, bolji ili lošiji po vrijednosti funkcije dobrote, od svojih roditelja. Zato, za svaki par ukrštenih jedinki odredimo minimum od brojeva medjuklusterskih veza za dva roditelja (tj. manji od brojeva medjuklusterskih veza dva roditelja) i minimum broja medjuklusterskih veza za njihove potomke (tj. manji od dva potomka). Ako je minimalan broj medjuklusterskih veza roditelja manji od minimalnog broja medjuklusterskih veza njihovih potomaka, tada kvalitet vrijednosti funkcije dobrote jedinki potomaka (svake ponaosob) poboljšavamo kombinovanjem genetskih algoritama i lokalnog pretraživanja.

Rješenje i njemu pridružen broj medjuklusterskih veza prvog potomka se pamte i postaju tekući. Slijede postupci poboljšanja vrijednosti funkcije prilagođenosti prvog potomka, razvijeni integracijom genetskih algoritama i lokalnog pretraživanja. Za svaki par klastera prve jedinke potomka:

- Premjestimo jednu ili više osoba ponaosob iz jednog u drugi klaster. Svakom pojedinačnom rješenju se pridruži broj medjuklusterskih veza i broj veza unutar klastera. Ako je broj medjuklusterskih veza pojedinačnog rješenja manji od broja medjuklusterskih veza trenutno najboljeg rješenja prvog potomka premještena osoba ostaje u klasteru, novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. Ako premještena osoba ne donosi poboljšanje vrijednosti 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 (svaku ponaosob) iz drugog u prvi klaster, pri čemu svakom pojedinačnom rješenju dodijeljujemo broj medjuklusterskih veza i broj veza unutar klastera. U slučaju da je broj medjuklusterskih veza pojedinačnog rješenja manji od broja medjuklusterskih veza trenutno najboljeg rješenja prvog potomka osoba premještena ostaje u klasteru, novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. U suprotnom, tu osobu vraćamo 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 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 prvog potomka 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 vrijednosti funkcije cilja onda te osobe vraćamo u klastera odakle su premještene, tekuće

⁷ Generacija – skup jedinki.

⁸ Populacija – skup jedinki u i -tom koraku algoritma.

rješenje i njemu dodijeljena vrijednost funkcije dobrote ostaju nepromijenjeni.

Razvijeni postupci poboljšanja vrijednosti funkcije dobrote jedinke prvog potomka su dali kvalitetna rješenja uz dodatno postavljene uslove da svaki klaster mora imati bar $N/(2 \cdot K)$ osoba i ne više od $(2 \cdot N)/K$ osoba. Uz intuitivno postavljene kriterije minimalnog i maksimalnog broja osoba po klasterima razvijeni postupci za pronalaženje rješenja su se pokazali efikasnim.

Prethodno opisani postupci premještanja instance N osoba u željen broj klastera K imaju u osnovi integraciju genetskih algoritama i lokalnog pretraživanja, na istoj reprezentaciji rješenja. Glavna odlika lokalnog pretraživanja je tip okoline. U postupku premještanja instance N osoba između K klastera prisutne su Swap i Interchange okoline. Swap okolina dobija se premještanjem po jedne osobe sa njene prvobitne pozicije u klasteru na sve dopustive pozicije svih K klastera. Interchange okolinu čine sve dopustive permutacije koje se dobijaju zamjenom pozicija para osoba za svaki par klastera K.

U cilju poboljšanja kvaliteta funkcije dobrote jedinke prvog potomka (prethodno opisani postupci) primjenu metode lokalnog pretraživanja prate genetski algoritmi, na istoj reprezentaciji rješenja. Lokalno pretraživanje podrazumijeva da se u okolini jedinke prvog potomka nađe bolje rješenje od trenutno najboljeg rješenja i njemu pridruži vrijednost funkcije dobrote, koji se čuvaju i pamte. Proces lokalnog pretraživanja odvija se u iteracijama i zaustavlja kada u okolini tekućeg rješenja ne postoji bolje rješenje. U tom momentu, poboljšanje vrijednosti funkcije dobrote tekućeg rješenja, nastavljaju genetski algoritmi. Ako u okolini tekućeg rješenja nađu bolje rješenje od tog trenutno najboljeg rješenja onda se rješenje lokalnog pretraživanja na kome bi pretraživanje stalo mijenja ovim boljim rješenjem i njemu pridruženim brojem medjuklusterskih veza.

Na Sl. 1 dat je prikaz integracije dvije različite heuristike (metode lokalnog pretraživanja i genetskih algoritama) na istoj reprezentaciji rješenja, koje rješavaju isti problem podjele instance N osoba u željen broj klastera K, tako da se minimizira broj medjuklusterskih veza. Domen koji se optimizuje je zbog jednostavnosti prikazan kao jednodimenzionalna funkcija. Prva od dvije heuristike je prikazana punom linijom, dok je druga prikazana tačkastom linijom. Obje heuristike traže minimalnu vrijednost i imaju isti

globalni optimum, ali se razlikuju u lokalnim optimumima. Ukoliko npr. gornja heuristika (puna linija) zaglavi u optimumu B, donja heuristika (tačkasta linija) će iz te tačke pomaknuti rješenje u tačku C koja je za donju heuristiku lokalni optimum. Nastavi li gornja heuristika obrađivati rješenje koje je dobila od donje heuristike (tačka D), penjući se po grafu završiti će u globalnom optimumu F (koji dijele obje funkcije).

Integracijom heuristika, genetskih algoritama i lokalnog pretraživanja, na istoj reprezentaciji rješenja poboljšali smo vrijednost funkcije prilagođenosti jedinke prvog potomka. Na isti način (prethodno opisani postupci poboljšanja vrijednosti funkcije prilagođenosti jedinke prvog potomka) ćemo poboljšati vrijednost funkcije prilagođenosti jedinke drugog potomka. Upoređivanjem vrijednosti funkcija dobrote jedinki prvog roditelja i prvog potomka odabiremo kvalitetniju jedinku u tekuću populaciju. Isto tako, upoređivanjem vrijednosti funkcija dobrote jedinki drugog roditelja i drugog potomka odabiremo kvalitetniju jedinku u tekuću populaciju.

Sljedeći primjer ilustruje postupak ukrštanja para jedinki (Parent[0], Parent[1]), instance N=30 osoba grupisanih u K=4 klastera, jednom tačkom presjeka, pri čemu je za ove dvije jedinke dat izgled distribucije gena (osoba), sadržaj klastera, ukupan broj medjuklusterskih veza i ukupan broj veza unutar svakog klastera:

↓ tačka ukrštanja

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

```
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
```

```
Ukupan broj medjuklusterskih prijateljstava je 37
```

```
Ukupan broj prijateljstava u Cluster[0] = 76
```

```
Ukupan broj prijateljstava u Cluster[1] = 15
```

```
Ukupan broj prijateljstava u Cluster[2] = 11
```

```
Ukupan broj prijateljstava u Cluster[3] = 2
```

```
Cluster[0] = 4 10 2 28 26 14 5 9 22 29 6 13 30 27
```

```
Cluster[1] = 11 17 19 15 24 25 16 18 7 23
```

```
Cluster[2] = 3 21 8
```

```
Cluster[3] = 20 1 12
```

```
Ukupan broj medjuklusterskih prijateljstava je 46
```

```
Ukupan broj prijateljstava u Cluster[0] = 46
```

```
Ukupan broj prijateljstava u Cluster[1] = 45
```

```
Ukupan broj prijateljstava u Cluster[2] = 2
```

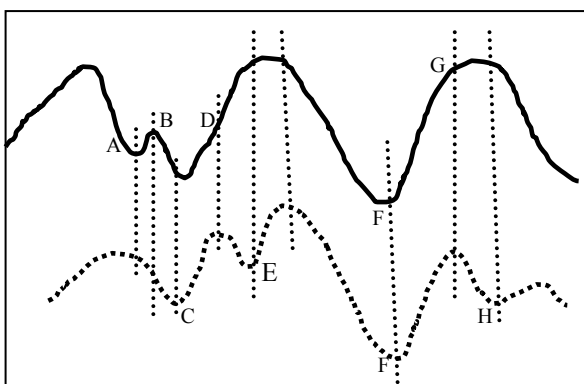
```
Ukupan broj prijateljstava u Cluster[3] = 2
```

Dvije jedinke predstavljaju roditelje:

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

Njima su dodijeljene vrijednosti funkcije dobrote. Jedinka prvog roditelja ima pridruženo 37 medjuklusterskih veza, a jedinka drugog roditelja ima pridruženo 46 medjuklusterskih veza. Posle ukrštanja ove dvije jedinke jednom slučajno odabranom tačkom presjeka i razmjenom genetskog materijala nastaju dvije jedinke potomaka:

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



Slika 1. Primjer integracije heuristika na istoj reprezentaciji rješenja

Jedinka prvog potomka ima duple gene: 18 3 15 7 23 16 1 24 25 20
Jedinka drugog potomka ima duple gene: 4 10 14 5 22 29 6 13 30 27
U prvom potomku nedostaju geni: 4 5 6 10 13 14 22 27 29 30
U drugom potomku nedostaju geni: 1 3 7 15 16 18 20 23 24 25

U prvom potomku duple gene mijenjamo genima koji nedostaju od osnovne instance N osoba. Na isti način to radimo sa jedinkom drugog potomka. Kao rezultat, dobijamo jedinke (dalje u tekstu popravljenih) potomaka, pri čemu je dat izgled distribucije gena, sadržaj klastera, pridružena vrijednost funkciju dobrote i ukupan broj veza unutar svakog klastera:

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

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

Ukupan broj medjuklusterskih prijateljstava je 87
Ukupan broj prijateljstava u Cluster[0] = 38
Ukupan broj prijateljstava u Cluster[1] = 7
Ukupan broj prijateljstava u Cluster[2] = 7
Ukupan broj prijateljstava u Cluster[3] = 2

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

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

Ukupan broj medjuklusterskih prijateljstava je 76
Ukupan broj prijateljstava u Cluster[0] = 47
Ukupan broj prijateljstava u Cluster[1] = 13
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

Pošto prvi roditelj ima pridruženo 37 medjuklusterskih veza, a drugi roditelj 46 slijedi da je $\text{Min}(\text{Parent}[0], \text{Parent}[1])=37$. Prvi potomak ima dodijeljeno 87 medjuklusterskih veza, a drugi potomak 76 slijedi da je $\text{Min}(\text{Childpopravka}[0], \text{Childpopravka}[1])=76$. Sada uporedimo odnos broja medjuklusterskih veza roditelja i njihovih potomaka. Vidimo da je manji broj medjuklusterskih veza roditelja od njihovih potomaka. Prethodno opisanim postupcima hibridizacije genetskih algoritama heuristikama lokalnog pretraživanja na istoj reprezentaciji rješenja nastojat ćemo prvo poboljšati kvalitet jedinke prvog potomka, a potom jedinke drugog potomka.

Rješenje i njemu pridružen broj medjuklusterskih veza prvog potomka se pamte i postaju tekući. Integracijom genetskih algoritama i lokalnog pretraživanja na istoj reprezentaciji rješenja poboljšavamo kvalitet jedinke prvog potomka premještanja instance $N=30$ osoba između $K=4$ klastera pri čemu slijede rješenja sa manjim brojem medjuklusterskih veza, izgled sadržaja klastera, broj medjuklusterskih veza i broj veza unutar svakog klastera:

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

Ukupan broj medjuklusterskih prijateljstava je 85
Ukupan broj prijateljstava u Cluster[0] = 36
Ukupan broj prijateljstava u Cluster[1] = 11
Ukupan broj prijateljstava u Cluster[2] = 7

Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 76
Ukupan broj prijateljstava u Cluster[0] = 41
Ukupan broj prijateljstava u Cluster[1] = 15
Ukupan broj prijateljstava u Cluster[2] = 7
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 51
Ukupan broj prijateljstava u Cluster[0] = 49
Ukupan broj prijateljstava u Cluster[1] = 3
Ukupan broj prijateljstava u Cluster[2] = 36
Ukupan broj prijateljstava u Cluster[3] = 2

Ako je broj medjuklusterskih veza pojedinačnog rješenja prvog potomka manji od broja medjuklusterskih veza trenutno najboljeg rješenja (prvog potomka) novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. U suprotnom, tekuće rješenje i vrijednost funkcije dobrote prvog potomka ostaju nepromijenjeni. Jedinka prvog potomka je imala pridruženo 87 medjuklusterskih veza. Postupcima integracije genetskih algoritama i lokalnog pretraživanja na istoj reprezentaciji rješenja je poboljšana vrijednost funkcije dobrote jedinke prvog potomka. Broj medjuklusterskih veza jedinke prvog potomka je smanjen na 51 medjuklustersku vezu.

Rješenje i njemu pridružen broj medjuklusterskih veza drugog potomka se pamte i postaju tekući. Takođe, integracijom genetskih algoritama i lokalnog pretraživanja na istoj reprezentaciji rješenja poboljšavamo vrijednost funkcije dobrote jedinke drugog potomka premještanjem instance $N=30$ osoba između $K=4$ klastera, pri čemu slijede rješenja sa manjim brojem medjuklusterskih veza, izgled sadržaja klastera, pridružen broj medjuklusterskih veza i broj veza unutar svakog klastera:

Cluster[1]= 11 17 19 5 14 27 6 8 13 10 9
Cluster[2]= 30 4 29
Cluster[3]= 12 22 21

Ukupan broj medjuklusterskih prijateljstava je 74
Ukupan broj prijateljstava u Cluster[0] = 46
Ukupan broj prijateljstava u Cluster[1] = 16
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 71
Ukupan broj prijateljstava u Cluster[0] = 44
Ukupan broj prijateljstava u Cluster[1] = 21
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 64
Ukupan broj prijateljstava u Cluster[0] = 54
Ukupan broj prijateljstava u Cluster[1] = 18
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 55
Ukupan broj prijateljstava u Cluster[0] = 65
Ukupan broj prijateljstava u Cluster[1] = 16
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 42
Ukupan broj prijateljstava u Cluster[0] = 78
Ukupan broj prijateljstava u Cluster[1] = 16
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 39
Ukupan broj prijateljstava u Cluster[0] = 77
Ukupan broj prijateljstava u Cluster[1] = 20
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 38
Ukupan broj prijateljstava u Cluster[0] = 78
Ukupan broj prijateljstava u Cluster[1] = 20
Ukupan broj prijateljstava u Cluster[2] = 3
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 37
Ukupan broj prijateljstava u Cluster[0] = 78
Ukupan broj prijateljstava u Cluster[1] = 18
Ukupan broj prijateljstava u Cluster[2] = 6
Ukupan broj prijateljstava u Cluster[3] = 2

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

Ukupan broj medjuklusterskih prijateljstava je 35
Ukupan broj prijateljstava u Cluster[0] = 78
Ukupan broj prijateljstava u Cluster[1] = 16
Ukupan broj prijateljstava u Cluster[2] = 10
Ukupan broj prijateljstava u Cluster[3] = 2

Ako je broj medjuklusterskih veza pojedinačnog rješenja drugog potomka manji od broja medjuklusterskih veza trenutno najboljeg rješenja drugog potomka, novo rješenje i njemu pridružena vrijednost funkcije dobrote se pamte i postaju tekući. U suprotnom, tekuće rješenje i njemu

dodijeljena vrijednost funkcije dobrote drugog potomka ostaju nepromijenjeni. Jedinka drugog potomka je imala 74 medjuklusterske veze. Kombinacijom genetskih algoritama i lokalnog pretraživanja na istoj reprezentaciji rješenja je poboljšana vrijednosti funkcije prilagođenosti jedinke drugog potomka. Broj medjuklusterskih veza drugog potomka je minimiziran na 35.

Ako uporedimo jedinke prvog roditelja (ima 37 medjuklusterskih veza) i prvog poboljšanog potomka (ima 51 medjuklustersku vezu), po broju medjuklusterskih veza, bolju vrijednost funkcije dobrote ima jedinka prvog roditelja. Ovu jedinku odabiremo u tekuću populaciju. Zatim, ako uporedimo jedinke drugog roditelja (ima 46 medjuklusterskih veza) i drugog poboljšanog potomka (ima 35 medjuklusterskih veza), po broju medjuklusterskih veza, bolju vrijednost funkcije dobrote ima jedinka drugog poboljšanog potomka i nju odabiremo u tekuću populaciju.

Na isti način nastavljamo proces ukrštanja jednom tačkom presjeka ostalih parova jedinki početne populacije, prethodno opisanim postupcima. Slično postupku ukrštanja jedinki jednom tačkom, ukrštamo jedinke sa dvije, tri ili više tačaka presjeka.

ZAKLJUČAK

Integracija genetskih algoritama i lokalnog pretraživanja, na istoj reprezentaciji rješenja postupka ukrštanja, predstavlja naučni doprinos poboljšanja kvaliteta jedinki inicijalne populacije, u okviru rješavanja problema podjele na klastere tako da se minimizira broj medjuklusterskih veza, u društvenim mrežama. Takođe, hibridizacija ovih metoda omogućava poređenje, kako osnovnih verzija metoda, tako i njihovih poboljšanja i kombinacija (hibrida). Korištenje više različitih heuristika osigurava bolju raznovrsnost reprezentacije rješenja populacije (jer se populacija optimizuje u različitim smjerovima), te lakše napuštanje lokalnih optimuma.

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. Isti programski kod se može lako prilagoditi različitim primjenama u svim oblastima ljudskih aktivnosti (biologije, anatomije, hemije, fizike, lingvistike, ekonomije, računarstva, sporta, muzike, ...).

Novi pravci istraživanja, donijeće rezultate hibridizacije primjenom nekih drugih heuristika, na istoj reprezentaciji rješenja u okviru implementacije proučavanog problema klasterovanja tako da se minimizira broj medjuklusterskih veza u društvenim mrežama i biće opisani u nekim od budućih radova.

ZAHVALNICA

Ovim putem zahvalila bih se mentoru dr Draganu Uroševiću vanrednom profesoru na Fakultetu Informatičkih Tehnologija SPU na vodstvu, 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.
- [7] M. Pirlot. General local search methods. *Europ. J. Oper. Res.*, 92:493-511, 1996.

ABSTRACT

In this paper we describe the value of improving the fitness function of individuals of the initial population of the integration of genetic algorithms (implementation of the genetic operators crosses), and local search, based on the same team and the solutions implemented in the framework of solving the problem of division in clusters so as to minimize the number of connections between clusters in social networks

IMPLEMENTING THE HEURISTICS IN THE SAME SOLUTION REPRESENTATION

Rava Filipović