

Modificiran genetski algoritam za konstrukciju BIBD-a

Doris Dumičić
(ddumicic@math.uniri.hr)
Odjel za matematiku, Sveučilište u Rijeci

- I. Martinjak, Mario-Osvin Pavčević: Modified Genetic Algorithm for BIBD Construction

Sadržaj

1 Uvod u genetski algoritam

- Povijesni razvoj GA
- Primjena GA
- Osnovno o GA
- Kodiranje kromosoma
- Selekcija
- Genetski operatori
- Parametri u GA
- Primjer

Sadržaj

1 Uvod u genetski algoritam

- Povijesni razvoj GA
- Primjena GA
- Osnovno o GA
- Kodiranje kromosoma
- Selekcija
- Genetski operatori
- Parametri u GA
- Primjer

2 BIBD

Sadržaj

1 Uvod u genetski algoritam

- Povijesni razvoj GA
- Primjena GA
- Osnovno o GA
- Kodiranje kromosoma
- Selekcija
- Genetski operatori
- Parametri u GA
- Primjer

2 BIBD

3 Modificirani GA za konstrukciju BIBD-a

Sadržaj

1 Uvod u genetski algoritam

- Povijesni razvoj GA
- Primjena GA
- Osnovno o GA
- Kodiranje kromosoma
- Selekcija
- Genetski operatori
- Parametri u GA
- Primjer

2 BIBD

3 Modificirani GA za konstrukciju BIBD-a

4 Optimizacija ulaznih parametara

Sadržaj

1 Uvod u genetski algoritam

- Povijesni razvoj GA
- Primjena GA
- Osnovno o GA
- Kodiranje kromosoma
- Selekcija
- Genetski operatori
- Parametri u GA
- Primjer

2 BIBD

3 Modificirani GA za konstrukciju BIBD-a

4 Optimizacija ulaznih parametara

5 Rezultati

Povijesni razvoj GA

- 1950-1960 prvi pokušaji simuliranja biološke evolucije na računalima,

Povijesni razvoj GA

- 1950-1960 prvi pokušaji simuliranja biološke evolucije na računalima,
- 1960-1970 - pojavljuju se prve uspješne primjene i znanstveni članci na temu genetičkih algoritma,

Povijesni razvoj GA

- 1950-1960 prvi pokušaji simuliranja biološke evolucije na računalima,
- 1960-1970 - pojavljuju se prve uspješne primjene i znanstveni članci na temu genetičkih algoritma,
- 1970-1980 - John Holland i grupa kolega i studenata definiraju genetički algoritam,

Povijesni razvoj GA

- 1950-1960 prvi pokušaji simuliranja biološke evolucije na računalima,
- 1960-1970 - pojavljuju se prve uspješne primjene i znanstveni članci na temu genetičkih algoritma,
- 1970-1980 - John Holland i grupa kolega i studenata definiraju genetički algoritam,
- 1992. godine – prvi primjer genetičkog programiranja,

Povijesni razvoj GA

- 1950-1960 prvi pokušaji simuliranja biološke evolucije na računalima,
- 1960-1970 - pojavljuju se prve uspješne primjene i znanstveni članci na temu genetičkih algoritma,
- 1970-1980 - John Holland i grupa kolega i studenata definiraju genetički algoritam,
- 1992. godine – prvi primjer genetičkog programiranja,
- Pojavom osobnih računala genetički algoritmi se počinju naglo razvijati i upotrebljavati.

Primjena genetskog algoritma

- Traženje najkraćeg puta,
- problem trgovačkog putnika, (TSP)
- strategija igara,
- problemi slični transportnom problemu,
- problem rasporedivanja procesa,
- problem određivanja parametara sustava,
- optimiziranje upita nad bazom podataka,
- procjena rizika u poslovanju (banke, marketing i sl.) u situaciji gdje su podaci nepotpuni, nekonzistentni i imaju šum,
- planiranje električnih postrojenja,
- dizajn hidroelektričnih generatora,
- portretiranje osumnjičenika,
- bojanje i particioniranje grafova.

Genetski algoritam biološka pozadina

Genetski algoritmi (GA) su heurističke metode pretraživanja i optimizacije inspirirane procesima prirodne evolucije.

Genetski algoritam biološka pozadina

Genetski algoritmi (GA) su heurističke metode pretraživanja i optimizacije inspirirane procesima prirodne evolucije.

U svakom koraku takvog algoritma tretira se tzv. **populacija**, odnosno podskup skupa svih mogućih jedinki iste vrste čija veličina ovisi o problemu koji se promatra.

Populacija se sastoji jedinki, potencijalnih rješenja - **kromosoma**.

Algoritam se fokusira na optimizaciji slučajno generirane početne populacije, a ne na pronašlasku optimalnog rješenja na skupu svih mogućih jedinki.



Osnovno o genetskom algoritmu

- Kromosomi se ocjenjuju **dobrotom** pomoću **funkcije cilja ili dobrote** (*fitness function*).

Osnovno o genetskom algoritmu

- Kromosomi se ocjenjuju **dobrotom** pomoću **funkcije cilja ili dobrote** (*fitness function*).
- U svakoj iteraciji selektira se odgovarajući broj najbolje rangiranih kromosoma (*roditelja*) za daljnje generiranje novih i boljih kromosoma (*djece*).

Osnovno o genetskom algoritmu

- Kromosomi se ocjenjuju **dobrotom** pomoću **funkcije cilja ili dobrote** (*fitness function*).
- U svakoj iteraciji selektira se odgovarajući broj najbolje rangiranih kromosoma (*roditelja*) za daljnje generiranje novih i boljih kromosoma (*djece*).
- Djeca (neki članovi nove populacije) se dobivaju djelovanjem **genetskih operatora** (križanje, mutacija). Tako nastaju nova potencijalna rješenja koja predstavljaju novu generaciju populacije u GA.

Osnovno o genetskom algoritmu

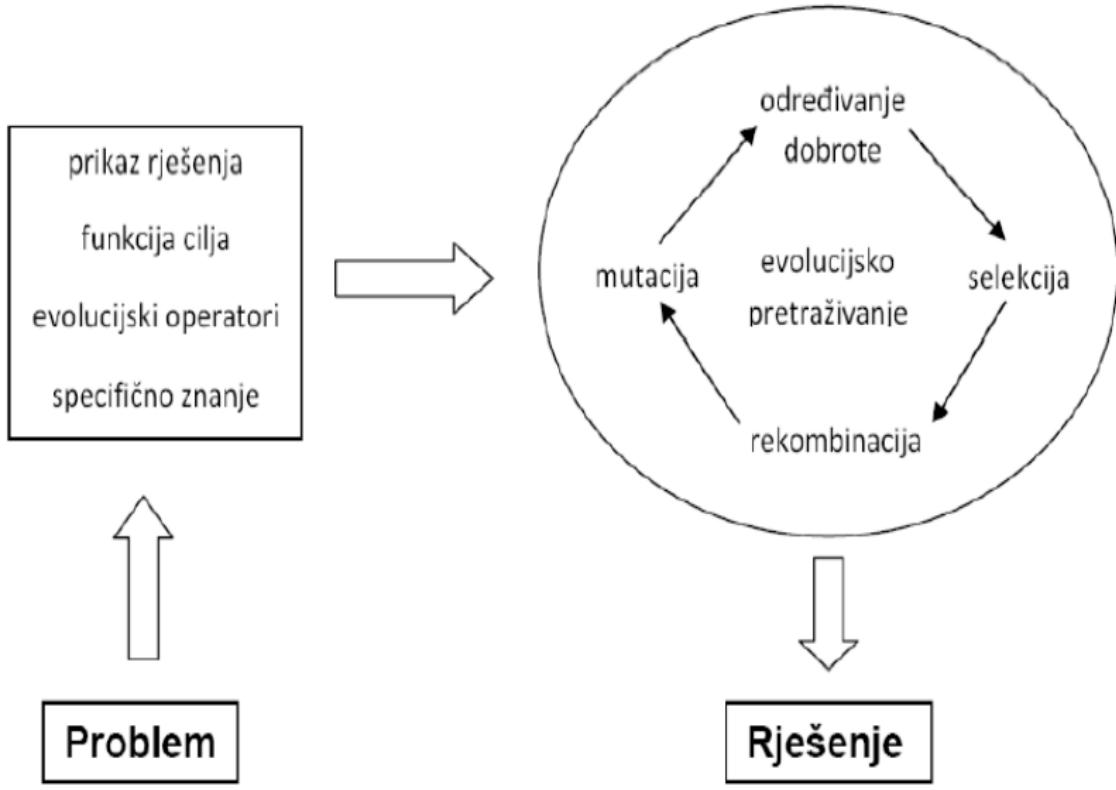
- Kromosomi se ocjenjuju **dobrotom** pomoću **funkcije cilja ili dobrote** (*fitness function*).
- U svakoj iteraciji selektira se odgovarajući broj najbolje rangiranih kromosoma (*roditelja*) za daljnje generiranje novih i boljih kromosoma (*djece*).
- Djeca (neki članovi nove populacije) se dobivaju djelovanjem **genetskih operatora** (križanje, mutacija). Tako nastaju nova potencijalna rješenja koja predstavljaju novu generaciju populacije u GA.
- Iteracija se nastavlja sve dok se ne zadovolji kriterij završetka genetskog procesa koji je definirao korisnik.

Osnovno o genetskom algoritmu

- Kromosomi se ocjenjuju **dobrotom** pomoću **funkcije cilja ili dobrote** (*fitness function*).
- U svakoj iteraciji selektira se odgovarajući broj najbolje rangiranih kromosoma (*roditelja*) za daljnje generiranje novih i boljih kromosoma (*djece*).
- Djeca (neki članovi nove populacije) se dobivaju djelovanjem **genetskih operatora** (križanje, mutacija). Tako nastaju nova potencijalna rješenja koja predstavljaju novu generaciju populacije u GA.
- Iteracija se nastavlja sve dok se ne zadovolji kriterij završetka genetskog procesa koji je definirao korisnik.
- Završetak rada znači da je pronađeno dovoljno dobro rješenje koje je vrlo blizu globalnog optimuma (odnosno maksimuma ili minimuma funkcije cilja).

Shema osnovnog GA

- [START] Generiranje proizvoljne populacije s n kromosoma (pogodno odabrana rješenja problema)
- [FUNKCIJA CILJA] Računanje vrijednosti funkcije cilja $f(x)$ za svaki kromosom x u populaciji
- [NOVA POPULACIJA] Kreiranje nove populacije ponavljanjem sljedećih koraka:
 - ① [SELEKCIJA] Prema funkciji cilja selektiraju se dva kromosoma - roditelji.
 - ② [KRIŽANJE] Križanje roditelja s odgovarajućom vjerojatnošću. Ako nema križanja, djeca su kopije roditelja.
 - ③ [MUTACIJA] Mutacija potomka u svakom lokusu (pozicija u kromosomu) s odgovarajućom vjerojatnosti.
 - ④ [PRIHVAT] Potomak ulazi u novu populaciju.
- [ZAMJENA] Na novoj se populaciji dalje provodi algoritam.
- [TEST] Ako je ispunjen uvjet završetka, **stop** i u trenutnu populaciju vrati najbolje rješenje.
- [PONAVLJAJ] Idi na drugu točku.



Kodiranje kromosoma

Kodiranje kromosoma je jedan od bitnih zadataka prilikom rješavanja problema pomoću GA. Ono ovisi o postavljenom problemu.

Binarno kodiranje

Ovo je najčešći i prvi oblik kodiranja u GA. Svaki kromosom se prezentira nizom bitova 0 ili 1. Primjer

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Problem pakiranja

Ruksak određenog kapaciteta treba napuniti sa što više predmeta zadanih veličina bez da se poveća veličina ruksaka.

Kodiranje: Svaki bit ukazuje na prisutstvo odgovarajućeg predmeta u ruksaku.

Permutacijsko kodiranje

Permutacijsko kodiranje se primjenjuje kod problema određivanja redoslijeda, kao na primjer problem trgovčkog putnika (TSP). Svaki kromosom je prikazan kao niz brojeva u odgovarajućem redoslijedu. Nužne su određene korekcije križanja i mutacije kako bi kromosomi ostali konzistentni.

TSP

Zadani su gradovi i njihove međusobne udaljenosti. Trgovčki putnik mora posjetiti sve gradove točno jedanput i vratiti se u početni grad s tim da priđe najkraći put. Treba odrediti redoslijed obilaska gradova s minimalnom sumom pređenih udaljenosti. Kodiranje: Kromosom predstavlja jedan redoslijed obilaska gradova.

Prilagođeno kodiranje

Kod ovakvog kodiranja, svaki kromosom je predstavljen nizom odgovarajućih vrijednosti (cijeli, realni brojevi, slova, dijagrami, ili neki drugi znakovi) povezanih s promatranim problemom.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGL
Chromosome C	(back), (back), (right), (forward), (left)

Često je nužno razviti nova križanja i mutacije.

Selekcija kromosoma u populaciji

Kromosomi iz populacije se na temelju dobrote selektiraju za križanje. Na taj se način čuvaju i prenose dobra svojstva (dobri geni) na sljedeću generaciju jedinki, dok se lošija svojstva odbacuju.

Selekcija kromosoma u populaciji

Kromosomi iz populacije se na temelju dobrote selektiraju za križanje. Na taj se način čuvaju i prenose dobra svojstva (dobri geni) na sljedeću generaciju jedinki, dok se lošija svojstva odbacuju.

GA, s obzirom na način selekcije dijelimo na: **generacijske i eliminacijske**.

Generacijske selekcije 1/3

Proporcionalna selekcija

Jedinke se biraju proporcionalno prema funkciji cilja. Jedinka sa najvećom vrijednosti funkcije cilja, ima najveće šanse da bude izabrana, dok najlošija jedinka ima najmanje šanse preživljavanja. Vjerojatnost preživljavanja ostalih jedinki je negdje između.

Generacijske selekcije 1/3

Proporcionalna selekcija

Jedinke se biraju proporcionalno prema funkciji cilja. Jedinka sa najvećom vrijednosti funkcije cilja, ima najveće šanse da bude izabrana, dok najlošija jedinka ima najmanje šanse preživljavanja. Vjerojatnost preživljavanja ostalih jedinki je negdje između.

Turnirska selekcija

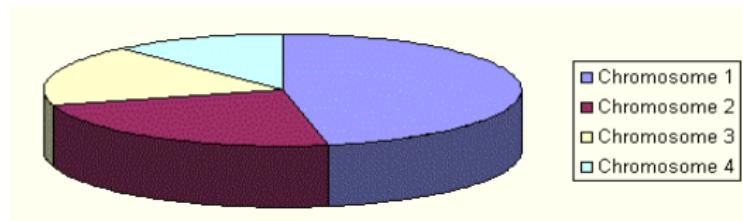
Odabire se slučajan broj jedinki iz populacije (sa ili bez zamjena, ovisno o implementaciji), a najbolje od njih postaju roditelji sljedeće generacije.

Postupak se ponavlja dok se ne popuni cijela sljedeća generacija, tj. dok broj novonastalih jedinki ne postane jednak veličini populacije. Ovaj se odabir često naziva i k-turnirski odabir, gdje k označava broj jedinki čije se vrijednosti funkcije cilja uspoređuju. Taj se broj naziva **veličina prozora**.

Generacijske selekcije 2/3

Rulet selekcija ili jednostavna selekcija

Svakom kromosomu u populaciji dodijeljen je dio kruga na temelju njegove vrijednosti funkcije cilja, kao na slici:



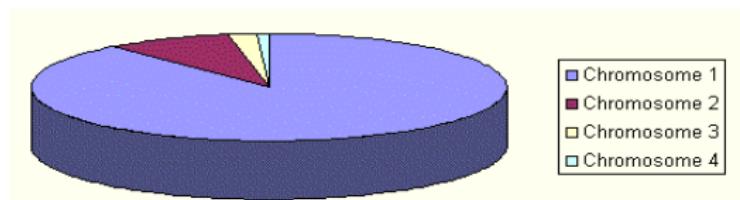
Algoritam rulet selekcije:

1. **[SUMA]** Zbrajanje vrijednosti funkcije cilja za svaki kromosom u populaciji - S .
2. **[SELEKCIJA]** Generira se proizvoljan broj $r \in [0, S]$.
3. **[PONAVLJANJE]** Prolazak kroz populaciju i zbrajaju se vrijednosti funkcije cilja od najmanje prema najvećoj. Kada je suma veća od r , izlaz je posljednji selektirani kromosom.

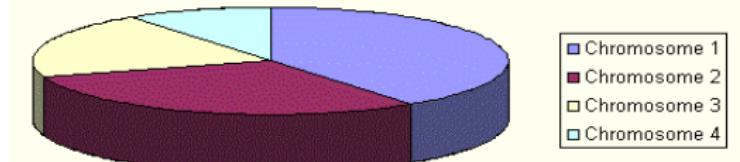
Generacijske selekcije 3/3

Selekcija po rangu

Cijela populacija se rangira tako da se svakom kromosomu promjeni vrijednost funkcije cilja (koja postaje ordinalni broj) tako da najlošije rangiran kromosom ima vrijednost funkcije cilja 1, sljedeći 2, ... i najbolje rangiran poprima vrijednost N (broj kromosoma u populaciji).



Situation before ranking (graph of fitnesses)



Situation after ranking (graph of order numbers)

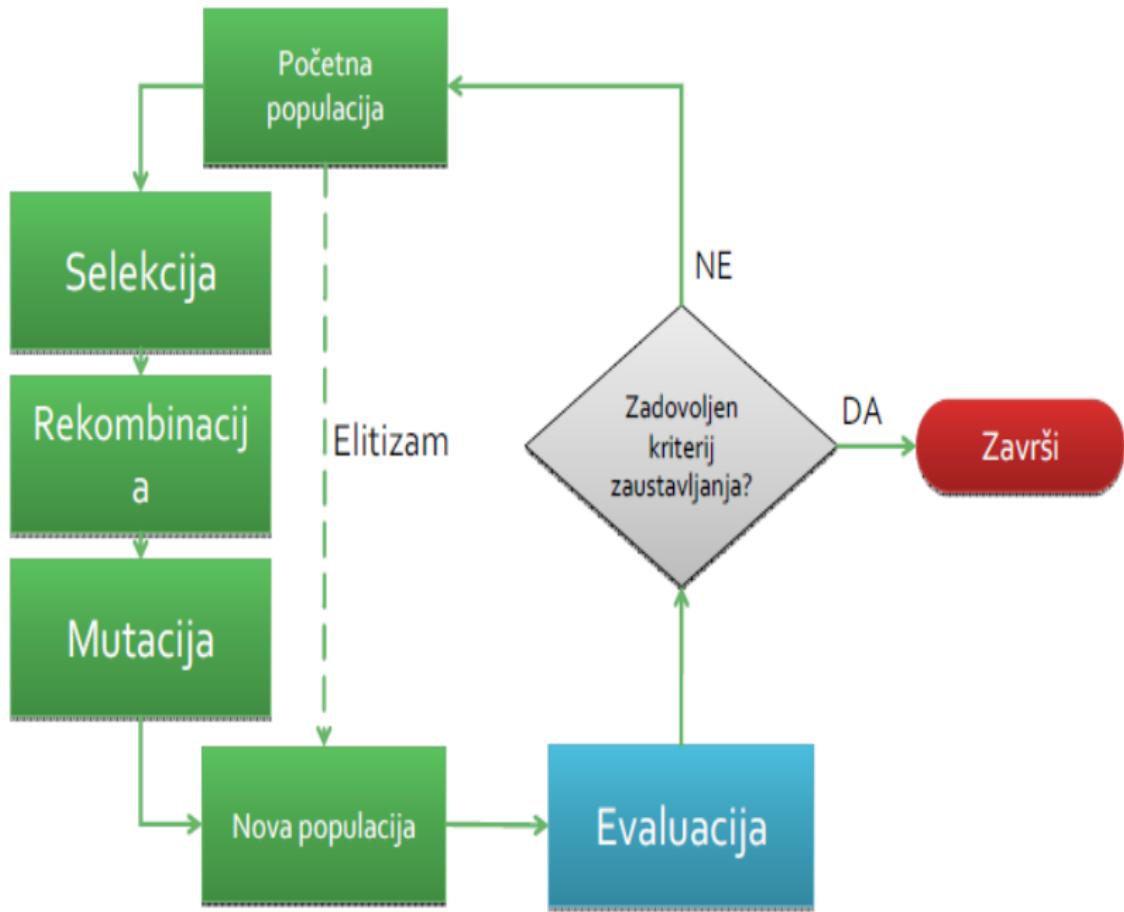
Eliminacijska selekcija

- ① Generiranje početne populacije.
- ② Selektiranje $m < VEL_{POP}$ "loših" kromosoma (vjerojatnost selekcije veća što je dobrota manja).
- ③ Brisanje selektiranih kromosoma.
- ④ Generiranje novih kromosoma od onih preživjelih pomoći genetskih operatora.
- ⑤ Dodavanje novih kromosoma i nadopunjavanje populacije.

Algoritam je jednostavan i brz za implementaciju, te daje vrlo dobre rezultate ukoliko se prilikom generiranja nove populacije eliminiraju duplikati.

Elitizam

Kod generiranja nove generacije, velika je šansa gubitka najboljih kromosoma. Kako se trenutno najbolja rješenja ne bi izgubila upotrebom genetskih operatora ili eliminacijom, tijekom selekcije javlja se potreba za zaštitom najbolje ili nekoliko najboljih jedinki od izmjena ili eliminacije. Zaštita najboljih jedinki naziva se **elitizam** i osigurava konvergenciju prema globalnom optimumu. Zaštićene jedinke zovemo **elitnim jedinkama**. Ukoliko je populacija velika, prilikom pretrage za najboljim jedinkama algoritam se usporava.



Genetski operatori

Križanje (*crossover*)

Križanjem u prirodi genetski se materijal prenosi s roditelja na djecu. Procesom križanja nastaju nove jedinke koje imaju kombinirane informacije sadržane u dvoje ili više roditelja.

Vjerojatnost križanja ukazuje na učestalost izvođenja križanja. Ako ona iznosi 0, potomci su kopije njihovih roditelja jer nema križanja. Ako je vjerojatnost križanja 1, tada je svaki potomak dobiven križanjem. Cilj križanja je proizvesti nove potomke (kromosome) koji će imati dobre dijelove roditeljskih kromosoma i time će nova možda biti bolja od stare generacije. Ipak, dobro je pustiti jedan dio populacije koji će preživiti u novoj generaciji. Općenito, vjerojatnost križanja treba biti visoka 80 - 95%. (Mada neki rezultati pokazuju da je za neke probleme najbolja vjerojatnost križanja 60%).

Najpoznatije izvedbe križanja za jednostavni prikaz rješenja binarnim brojevima:

1. Križanje u jednoj točki (*one point crossover*). Odabire se jedna točka na roditeljima (jednako udaljena od početka za oba roditelja) i mijenja se desni dio kromosoma prvog roditelja s desnim dijelom kromosoma drugog roditelja.

Roditelji	110011101010101010100000	
	1110000011010001000011	
Djeca	110011101010101001000011	
	1110000011010010100000	točka križanja

2. Križanje u dvije točke (*two point crossover*). Odabiru se dvije točke na roditeljima, pazeći da je pritom lijeva točka jednakim udaljenom od lijevog kraja i desna točka od desnog kraja kromosoma za oba roditelja. Roditelji izmjenjuju dijelove kromosoma između tih točaka.

Roditelji	110011	10101010	10100000
Djeca	11100000	01101000	10000011
Djeca	110011	00110100	10100000
Djeca	111000	10101010	01000011

točke križanja

3. Križanje rezanjem i spajanjem (*cut and splice crossover*). Na roditeljima se odabire točka koja nije jednako udaljena od početka roditeljskih kromosoma. Mijenja se desni dio kromosoma jednog roditelja s desnim dijelom kromosoma drugog. Na taj način nastaju djeca koja imaju različite duljine kromosoma.

Roditelji	1100111010101	010100000
	1110000011010001	000011
Djeca	1100111010101	000011
	1110000011010001	010100000

4. Uniformno križanje (*uniform crossover*). Uspoređuju se bit po bit oba roditelja i mijenjaju se sa fiksnom vjerojatnošću od 50%. Ako se ta vjerojatnost razlikuje za pojedine gene, tada govorimo o p-uniformnom križanju.

gen	1	2	3	4	5	6	7	8	9	...	b-3	b-2	b-1	b
p	0.1	0.5	0.9	0.1	1	1	0.5	0.1	0	0.7	0.7	0.6	0.1

4. Uniformno križanje (*uniform crossover*). Uspoređuju se bit po bit oba roditelja i mijenjaju se sa fiksnom vjerojatnošću od 50%. Ako se ta vjerojatnost razlikuje za pojedine gene, tada govorimo o p-uniformnom križanju.

gen	1	2	3	4	5	6	7	8	9	...	b-3	b-2	b-1	b
p	0.1	0.5	0.9	0.1	1	1	0.5	0.1	0	0.7	0.7	0.6	0.1

5. Segmentno križanje - točke prekida biraju se proizvoljno za svako križanje.

4. Uniformno križanje (*uniform crossover*). Uspoređuju se bit po bit oba roditelja i mijenjaju se sa fiksnom vjerojatnošću od 50%. Ako se ta vjerojatnost razlikuje za pojedine gene, tada govorimo o p-uniformnom križanju.

gen	1	2	3	4	5	6	7	8	9	...	b-3	b-2	b-1	b
p	0.1	0.5	0.9	0.1	1	1	0.5	0.1	0	0.7	0.7	0.6	0.1

5. Segmentno križanje - točke prekida biraju se proizvoljno za svako križanje.

6. Miješajuće križanje

Mutacija 1/2

Mutacija

U prirodi se mutacija definira kao slučajna promjena gena. Određeni postotak jedinki se mutira i time dobivamo potpuno novu populaciju. Vjerojatnosti promjene za gene su različite, pa geni mogu biti stabilni i nestabilni. Mutacija se koristi kako bi se održala raznovrsnost populacije i stvorile nove jedinke s novim svojstvima.

Mutacija 1/2

Mutacija

U prirodi se mutacija definira kao slučajna promjena gena. Određeni postotak jedinki se mutira i time dobivamo potpuno novu populaciju. Vjerojatnosti promjene za gene su različite, pa geni mogu biti stabilni i nestabilni. Mutacija se koristi kako bi se održala raznovrsnost populacije i stvorile nove jedinke s novim svojstvima.

Vjerojatnost mutacije p_m ukazuje na učestalost mutiranja dijelova kromosoma. Najjednostavniji primjer mutacije je vjerojatnost da se neki bit u genetskom kodu (rješenju) promijeni iz svog originalnog stanja u novo stanje. To se postiže uvođenjem varijable za svaki bit u nizu.

Ako se koristi binaran prikaz rješenja, mutacija je vjerojatnost da neki bit iz nule prijene u jedinicu ili iz jedinice u nulu. Ako je ona 1, cijeli kromosom je izmijenjen (invertiran), a ako je 0 niti jedan dio kromosoma nije izmijenjen.

Mutacija 2/2

Roditelj A	1	0	1	0	1	0	1	1	1	1	1	0
Roditelj B	0	1	1	1	0	1	0	0	0	1	1	0
Dijete A	0	1	0	0	1	0	1	1	1	1	1	0
Dijete B	1	0	1	1	0	1	0	0	1	1	1	0

Mutacija 2/2

Roditelj A	1	0	1	0	1	0	1	1	1	1	1	0
Roditelj B	0	1	1	1	0	1	0	0	0	1	1	0
Dijete A	0	1	0	0	1	0	1	1	1	1	1	0
Dijete B	1	0	1	1	0	1	0	0	1	1	1	0

Mutacija prevenira konvergenciju k lokalnom optimumu. Međutim, ne bi trebala biti preučestala, jer tada GA postaje algoritam slučajne pretrage. Vjerojatnost mutacije treba biti prilično niska 0,5-1%.

Mutacija 2/2

Roditelj A	1	0	1	0	1	0	1	1	1	1	1	0
Roditelj B	0	1	1	1	0	1	0	0	0	1	1	0
Dijete A	0	1	0	0	1	0	1	1	1	1	1	0
Dijete B	1	0	1	1	0	1	0	0	1	1	1	0

Mutacija prevenira konvergenciju k lokalnom optimumu. Međutim, ne bi trebala biti preučestala, jer tada GA postaje algoritam slučajne pretrage. Vjerojatnost mutacije treba biti prilično niska 0,5-1%.

- Jednostavna mutacija (Jednostavan GA)

Mutacija 2/2

Roditelj A	1	0	1	0	1	0	1	1	1	1	1	0
Roditelj B	0	1	1	1	0	1	0	0	0	1	1	0
Dijete A	0	1	0	0	1	0	1	1	1	1	1	0
Dijete B	1	0	1	1	0	1	0	0	1	1	1	0

Mutacija prevenira konvergenciju k lokalnom optimumu. Međutim, ne bi trebala biti preučestala, jer tada GA postaje algoritam slučajne pretrage. Vjerovatnost mutacije treba biti prilično niska 0,5-1%.

- Jednostavna mutacija (Jednostavan GA)
- Miješajuća mutacija

Mutacija 2/2

Roditelj A	1	0	1	0	1	0	1	1	1	1	1	0
Roditelj B	0	1	1	1	0	1	0	0	0	1	1	0
Dijete A	0	1	0	0	1	0	1	1	1	1	1	0
Dijete B	1	0	1	1	0	1	0	0	1	1	1	0

Mutacija prevenira konvergenciju k lokalnom optimumu. Međutim, ne bi trebala biti preučestala, jer tada GA postaje algoritam slučajne pretrage. Vjerovatnost mutacije treba biti prilično niska 0,5-1%.

- Jednostavna mutacija (Jednostavan GA)
- Miješajuća mutacija
- Potpuna mijesajuća mutacija

Parametri u GA 1/2

1. Veličina populacije

Veličina populacije $POP(t)$ = broj kromosoma u populaciji (u generaciji s rednim brojem t) ili u trenutku t . Uobičajeno $POP = \text{konst.}$

Parametri u GA 1/2

1. Veličina populacije

Veličina populacije $POP(t)$ = broj kromosoma u populaciji (u generaciji s rednim brojem t) ili u trenutku t . Uobičajeno $POP = \text{konst.}$

Prevelika populacija usporava algoritam, a premala ne stvara dovoljno novih potomaka. Istraživanja su pokazala da nakon neke granice (koja ovisi o promatranom problemu) nije korisno povećavati veličinu populacije, jer ne ubrzava proces rješavanja problema.

Dobra veličina za populaciju je 20-30, mada je ponekad i 50-100 najbolja veličina. Neka istraživanja također pokazuju da najbolja veličina populacije ovisi o kodiranju, odnosno veličini niza znakova kojima kodiramo kromosome.

Parametri u GA 2/2

2. Broj novih generacija

Parametri u GA 2/2

2. Broj novih generacija
3. Vjerojatnost mutacije p_m

Parametri u GA 2/2

2. Broj novih generacija
3. Vjerojatnost mutacije p_m
4. Vjerojatnost križanja p_c i broj jedinki za eliminaciju M za generacijski, odnosno eliminacijski GA.

Parametri u GA 2/2

2. Broj novih generacija
3. Vjerovatnost mutacije p_m
4. Vjerovatnost križanja p_c i broj jedinki za eliminaciju M za generacijski, odnosno eliminacijski GA.

Tablica uobičajenih vrijednosti parametara za GA s "malom" i "velikom" populacijom:

Parametri	oznaka	"mala" populacija	"velika" populacija
Veličina populacije	VEL_POP	30	100
Vjerovatnost mutacije	p_m	0.01	0.001
Vjerovatnost križanja	p_c	0.9	0.6
Broj jedinki za eliminaciju	M	VEL_POP/2	VEL_POP/4

Višepopulacijski GA

Novi genetski operatori:

- ubacivanje (*reinsertion*)
- migracija (*migration*)
- natjecanje (*competition*)

Višepopulacijski GA

Novi genetski operatori:

- ubacivanje (*reinsertion*)
- migracija (*migration*)
- natjecanje (*competition*)

Prilagodljivi GA (AGA)

- 1994. Srinivas i Patnaik
- Namijenjen je radu u okruženju generacijskog GA
- Cilj je da se p_m i p_c povećavaju za lošija rješenja, a smanjuju za smanjuju za bolja rješenja.

Primjer rada GA

Primjer 1

Traženje optimuma funkcije $f(x) = x^2$, $x \in [0, 31]$.

Primjer 2

Problem trgovačkog putnika (TS).

Konstrukcija BIBD-a primjenom GA

Razvili su GA za konstrukciju BIBD-a pretraživanjem bez dodatnih pretpostavki. I otkrili su nove jednostavne dizajne s parametrima $2 - (14, 4, 6)$.

Konstrukcija BIBD-a primjenom GA

Razvili su GA za konstrukciju BIBD-a pretraživanjem bez dodatnih pretpostavki. I otkrili su nove jednostavne dizajne s parametrima $2 - (14, 4, 6)$.

Definicija

Neka su $v, k, \lambda \in \mathbb{N}$ takvi da $v > k \geq 1$ i $\lambda \geq 1$. Neka je $\mathcal{P} = \{p_1, \dots, p_v\}$ i $\mathcal{B} = \{B_1, \dots, B_b\}$ disjunktni skup točaka i multiskup blokova i neka je $\mathcal{I} \subseteq \mathcal{P} \times \mathcal{B}$ relacija incidencije na tim skupovima.

Balansiran nepotpuni blok dizajn (BIBD) ili samo **blok dizajn** \mathcal{D} s parametrima $2 - (v, k, \lambda)$ je uređena trojka $\mathcal{D} = (\mathcal{P}, \mathcal{B}, \mathcal{I})$ za koju vrijedi:

- svaki blok iz \mathcal{B} je incidentan s k točaka,
- svaki par točaka iz \mathcal{D} je sadržan u λ blokova iz \mathcal{D} ,
- $1 < k < v - 1$.

Modificirani GA za konstrukciju BIBD-a

Modificirani GA za konstrukciju BIBD-a

Algoritam se temelji na 4-turnirskoj selekciji i križanju u 2 točke.

Modificirani GA za konstrukciju BIBD-a

Algoritam se temelji na 4-turnirskoj selekciji i križanju u 2 točke.

Svaki kromosom populacije je jedna $b \times k$ matrica čiji su geni zapravo retci k – člani podskup od skupa točaka $\mathcal{P} = \{0, 1, \dots, v - 1\}$. Jasno, takvi kromosomi su $2 - (v, k, \lambda)$ dizajni ako se svaki dvočlani podskup skupa točaka \mathcal{P} pojavljuje u točno λ redaka (gena) u tom kromosomu, bez dodatnih uvjeta koje treba ispitati da li su ispunjeni.

Označimo s POP broj kromosoma u populaciji.

Modificirani GA za konstrukciju BIBD-a

Algoritam se temelji na 4-turnirskoj selekciji i križanju u 2 točke.

Svaki kromosom populacije je jedna $b \times k$ matrica čiji su geni zapravo retci k – člani podskup od skupa točaka $\mathcal{P} = \{0, 1, \dots, v - 1\}$. Jasno, takvi kromosomi su $2 - (v, k, \lambda)$ dizajni ako se svaki dvočlani podskup skupa točaka \mathcal{P} pojavljuje u točno λ redaka (gena) u tom kromosomu, bez dodatnih uvjeta koje treba ispitati da li su ispunjeni.

Označimo s POP broj kromosoma u populaciji.

Prebrojavali su koliko se puta svaki dvočlani podskup od \mathcal{P} pojavljuje u retcima kromosoma zaustavljajući se na λ kada se ona dostigne. **Funkcija cilja** je definirana tako da svakom kromosomu pridružimo sumu brojeva pojavljivanja svakog para točaka.

Modificirani GA za konstrukciju BIBD-a

Algoritam se temelji na 4-turnirskoj selekciji i križanju u 2 točke.

Svaki kromosom populacije je jedna $b \times k$ matrica čiji su geni zapravo retci k – člani podskup od skupa točaka $\mathcal{P} = \{0, 1, \dots, v - 1\}$. Jasno, takvi kromosomi su $2 - (v, k, \lambda)$ dizajni ako se svaki dvočlani podskup skupa točaka \mathcal{P} pojavljuje u točno λ redaka (gena) u tom kromosomu, bez dodatnih uvjeta koje treba ispitati da li su ispunjeni.

Označimo s POP broj kromosoma u populaciji.

Prebrojavali su koliko se puta svaki dvočlani podskup od \mathcal{P} pojavljuje u retcima kromosoma zaustavljajući se na λ kada se ona dostigne. **Funkcija cilja** je definirana tako da svakom kromosomu pridružimo sumu brojeva pojavljivanja svakog para točaka.

U idealnom slučaju, kada je kromosom dizajn, suma iznosi

$$\binom{v}{2} \cdot \lambda = \frac{v(v-1) \cdot \lambda}{2},$$

što je maksimalna (optimalna) vrijednost za funkciju cilja.

U svakoj iteraciji, odabrana su dva najbolja kromosoma od četiri koji su proizvoljno određeni. Oni zatim postaju roditelji u križanju s dvije točke, kao što je prikazano za biravninu (7, 4, 2):

$$\begin{array}{c} \left[\begin{array}{cccc} 5 & 0 & 2 & 4 \\ 2 & 6 & 0 & 3 \\ \hline 6 & 1 & 4 & 0 \\ 2 & 1 & 5 & 6 \\ 3 & 0 & 5 & 6 \\ 0 & 2 & 1 & 5 \\ \hline 0 & 2 & 6 & 3 \end{array} \right] \quad \left[\begin{array}{cccc} 4 & 6 & 1 & 2 \\ 6 & 2 & 3 & 4 \\ \hline 0 & 4 & 5 & 3 \\ 5 & 4 & 3 & 0 \\ 1 & 0 & 4 & 6 \\ 2 & 0 & 6 & 3 \\ \hline 6 & 4 & 1 & 2 \end{array} \right] \\ \text{parent A} \qquad \text{parent B} \end{array}$$

$$\begin{array}{c} \left[\begin{array}{cccc} 5 & 0 & 2 & 4 \\ 2 & 6 & 0 & 3 \\ \hline 0 & 4 & 5 & 3 \\ 5 & 4 & 3 & 0 \\ 1 & 0 & 4 & 6 \\ 2 & 0 & 6 & 3 \\ \hline 0 & 2 & 6 & 3 \end{array} \right] \quad \left[\begin{array}{cccc} 4 & 6 & 1 & 2 \\ 6 & 2 & 3 & 4 \\ \hline 6 & 1 & 4 & 0 \\ 2 & 1 & 5 & 6 \\ 3 & 0 & 5 & 6 \\ 0 & 2 & 1 & 5 \\ \hline 6 & 4 & 1 & 2 \end{array} \right] \\ \text{child A} \qquad \text{child B} \end{array}$$

Dopuštene su dvije vrste mutacija, odnosno mutacijskih operatora.

Jedno dijete može imati zamijenjen gen koji je odabran proizvoljno kao k-člani podskup od \mathcal{P} , dok drugo dijete može imati zamijenjen samo jedan element proizvoljno odabranog gena.

Oba operatora mutacije će se dogoditi s jednakom vjerojatnošću p_m , uslijed eksperimentalnog iskustva.

Algoritam GABIBD($v, k, \lambda, POP, c_{max}$)

```

generate initial population
 $c \leftarrow 0$ 
while ( $f_{best} < numPairs$  and  $c < c_{max}$ )
    mix the positions of chroms
     $i \leftarrow 0$ 
    while ( $i < POP$ )
        select two better among the
         $i$ th -  $(i + 3)$ th chrom
        create two new chrom through
        crossover and mutate it
         $i \leftarrow i + 4$ 
        evaluate the fitness of chromosomes
         $c \leftarrow c + 1$ 
end

```

Optimizacija ulaznih parametara

Nakon nekoliko provedenih eksperimenata optimizirali su ulazne parametre kako bi poboljšali algoritam.

U tijeku testiranja 100 puta su pokretali algoritam za svaku odabranu trojku (v, k, λ) dok nisu pronašli rješenje.

Najprije su ustanovili da su rezultati značajno bolji ako se zamjeni pozicija kromosoma (dva proizvoljno odabrana kromosoma mijenjaju poziciju u populaciji) prije njihovog odabira i tako dva susjedna kromosoma u jednoj iteraciji, nisu više susjedi u sljedećoj iteraciji. Pokazalo se da je optimalno oko 1000 zamjena za bilo koje dopustive parametre dizajna.

Zatim su nizom testova dobili očekivane rezultate što se tiče veličine populacije odnosno POP : veća populacija, bolji rezutati. Isprobali su za populaciju veličine od 40-600 i pokazalo se da je za POP optimum 60 s obzirom na definiciju funkcije cilja.

GA je općenito vrlo osjetljiv na vjerojatnost mutacije, njihov glavni cilj je pronaći takav p_m koji omogućuje najmanji mogući broj iteracija koje vode do prvog rješenja. Kako su postupno povećavali vjerojatnost mutacije, primijetili su (tablica) sve bolje rezultate i u prosječnom broju iteracija do prvog dizajna, avi i u broju rješenja, $nsol$.

(v, k, λ)	p_m	95	98	100
(11,5,2)	avi	3856	3386	2657
	nsol	45	59	76
(19,3,1)	avi	19394	19062	18800
	nsol	9	11	18
(21,3,1)	avi	48462	85136	65931
	nsol	7	35	63

Usporedba dviju različitih početnih populacija

Populacije se generiraju proizvoljno.

(v, k, λ)	<i>mini</i>	<i>avi</i>	<i>mini2</i>	<i>avi2</i>
(6,3,2)	10	48	0	64
(7,3,1)	14	49	1	109
(7,4,2)	16	69	1	228
(8,4,3)	74	586	173	1619
(11,5,2)	148	8479	700	7001

Rezultati

Proveli su dva niza testiranja. Jedno s ciljem mjerenja raspona algoritma i drugo s ciljem povećanja nekih već poznatih rezultata. Kod prvih eksperimenata, za svaki od odabralih parametara dizajna proveli su 100 pokretanja algoritma.

U svakom od pokretanja, granica za iteracije, c_{max} je postavljena na 100000, a vjerojatnost mutacije je postavljena na 100 posto.

Uzorak za testiranje sastoji se od nekoliko BIBD klasa, čiji rezultati su u sljedećoj tablici:

2- (v, k, λ)	mini	avi	nsol
(8,4,3)	74	586	108
(8,4,15)	189	891	100
(10,3,2)	107	677	101
(10,5,4)	1545	39810	92
(12,4,3)	5106	74106	47
(21,3,2)	5166	52609	82
(7,3,1)	14	49	108
(13,4,1)	170	1963	104
(21,5,1)	42233	98238	5
(7,4,2)	16	69	108
(11,5,2)	148	8479	106
(9,3,1)	50	189	107
(13,3,1)	245	4005	105
(15,3,1)	416	12672	104
(19,3,1)	5305	51143	85
(21,3,1)	6372	65931	63
(25,3,1)	17724	94821	17
(27,3,1)	57064	99375	2

U drugom nizu testiranja, c_{max} je povećana na milijun, dok su ostali parametri ostali nepromijenjeni kao u prethodnom nizu testiranja. Rezultati su prikazani u sljedećoj tablici:

D	$nsol$	$niso$	$ndes$	$ Aut(D) $
2-(13,5,5)	20	20	20	20×1
2-(14,4,6)	74	74	20	20×1

Dobiveni dizajni s parametrima $2 - (13, 5, 5)$ su tada već poznati.

Teorem 1

Postoji barem 78 dizajna s parametrima $2 - (14, 4, 6)$. Postoji barem 24 jednostavnih dizajna s tim parametrima. Njih 74 imaju trivijalnu grupu automorfizama.

Martinjak,I. Pavčević, M. Modified Genetic Algorithm for BIBD Construction, Proceedings of the ITI 2009 31st Int. Conf. on Information Technology Interfaces , Cavtat, 2009. 647-652.

Tabu search algoritam za konstrukciju blok dizajna (V.Krčadinac, dokt.dizertacija 1/2)

Tabu search algoritam za konstrukciju blok dizajna (V.Krčadinac, dokt.dizertacija 1/2)

Nove donje ocijene za broj neizomorfnih dizajna

R.br.	(v, k, λ)	Handbook	Novo	R.br.	(v, k, λ)	Handbook	Novo
69	(19, 4, 2)	423	10^4	361	(19, 4, 4)	424	5000
70	(13, 4, 3)	3702	10^4	416	(11, 5, 10)	3337	10^4
102	(15, 5, 4)	207	10^4	447	(14, 4, 6)	4	2500
123	(16, 5, 4)	11	10^4	520	(29, 4, 3)	1	200
124	(13, 5, 5)	30	10^4	525	(15, 5, 8)	104	10^4
128	(16, 6, 5)	15	300	529	(15, 6, 10)	118	10^4
147	(25, 4, 2)	17	1000	606	(16, 5, 8)	12	5000
148	(17, 4, 3)	1	10^4	607	(13, 5, 10)	31	10^4
149	(13, 4, 4)	2462	10^4	688	(17, 4, 6)	1	250
220	(20, 4, 3)	1	2500	742	(16, 6, 11)	1	10^4
221	(20, 5, 4)	1	400	785	(18, 4, 6)	1	100
238	(31, 4, 2)	1	50	788	(18, 6, 10)	4	1000
239	(21, 4, 3)	1	2500	825	(15, 5, 10)	1	1000
242	(11, 4, 6)	1	10^4	835	(16, 7, 14)	1	1000
246	(17, 5, 5)	1	5000	1000	(20, 5, 8)	1	1000
247	(11, 5, 8)	4394	10^4	1036	(16, 6, 13)	1	1000
284	(16, 6, 7)	1	5000	1037	(14, 6, 15)	1	5000
340	(24, 4, 3)	1	1000	1094	(17, 5, 10)	1	1000

Tabu search algoritam za konstrukciju blok dizajna (V.Krčadinac, dokt.dizertacija 2/2)

Broj konstruiranih dizajna u 50 poziva tabu search algoritma:

(v, k, λ)	Br.								
(7, 3, 1)	50	(10, 4, 4)	50	(6, 3, 16)	50	(25, 9, 3)	11	(12, 4, 6)	46
(6, 3, 2)	50	(16, 4, 1)	42	(16, 8, 7)	0	(15, 5, 4)	48	(12, 6, 15)	50
(7, 3, 2)	50	(9, 3, 3)	50	(11, 5, 8)	50	(21, 7, 3)	0	(9, 4, 15)	50
(9, 3, 1)	50	(9, 4, 6)	47	(9, 4, 9)	50	(7, 3, 13)	50	(13, 4, 5)	48
(8, 4, 3)	50	(8, 4, 9)	50	(7, 3, 10)	50	(16, 4, 2)	49	(11, 5, 14)	45
(6, 3, 4)	50	(13, 3, 1)	50	(13, 4, 3)	50	(16, 6, 5)	6	(9, 3, 8)	50
(11, 5, 2)	50	(13, 4, 2)	50	(13, 5, 5)	50	(9, 3, 6)	50	(21, 5, 2)	0
(7, 3, 3)	50	(13, 6, 5)	40	(16, 6, 4)	0	(9, 4, 12)	50	(21, 6, 3)	0
(10, 4, 2)	50	(7, 3, 7)	50	(15, 3, 1)	50	(15, 7, 9)	32	(21, 10, 9)	0
(9, 4, 3)	50	(6, 3, 12)	50	(15, 6, 5)	11	(13, 3, 2)	48	(16, 6, 7)	41
(13, 4, 1)	50	(10, 5, 8)	49	(12, 3, 2)	49	(13, 4, 4)	46	(10, 3, 6)	50
(6, 3, 6)	50	(19, 9, 4)	24	(12, 6, 10)	50	(13, 6, 10)	50	(10, 4, 12)	49
(10, 5, 4)	50	(11, 5, 6)	50	(23, 11, 5)	6	(10, 5, 16)	48	(15, 7, 12)	47
(7, 3, 4)	50	(14, 7, 6)	10	(7, 3, 11)	50	(19, 9, 8)	0	(18, 6, 5)	0
(9, 3, 2)	50	(16, 6, 3)	49	(9, 3, 5)	50	(11, 5, 12)	48	(22, 11, 10)	0
(8, 4, 6)	49	(7, 3, 8)	50	(10, 5, 12)	45	(22, 8, 4)	0	(15, 5, 6)	50
(15, 7, 3)	28	(12, 4, 3)	50	(8, 4, 15)	50	(14, 7, 12)	50	(16, 4, 3)	50
(6, 3, 8)	50	(6, 3, 14)	50	(17, 8, 7)	0	(27, 13, 6)	0	(16, 8, 14)	29
(11, 5, 4)	50	(9, 3, 4)	50	(7, 3, 12)	50	(21, 9, 6)	0	(31, 6, 1)	26
(7, 3, 5)	50	(7, 3, 9)	50	(10, 3, 4)	49	(10, 4, 10)	47	(31, 10, 3)	0
(16, 6, 2)	20	(21, 5, 1)	41	(10, 4, 8)	49	(25, 5, 1)	35	(31, 15, 7)	0
(12, 6, 5)	50	(8, 3, 6)	49	(11, 3, 3)	48	(9, 3, 7)	50	(11, 5, 16)	45
(7, 3, 6)	50	(8, 4, 12)	50	(11, 4, 6)	49	(20, 10, 9)	0	(22, 7, 4)	0
(6, 3, 10)	50	(10, 4, 6)	50	(11, 5, 10)	49	(16, 5, 4)	43	(9, 3, 9)	50
(10, 3, 2)	50	(15, 7, 6)	1	(18, 9, 8)	0	(16, 6, 6)	23	(25, 10, 6)	0