

Heurističke metode optimizacija

Usmjeravanje školskih autobusa

Vinko Kolobara
January 15, 2018

1 OPIS PROBLEMA

Problem usmjeravanja školskih autobusa je varijacija problema usmjeravanja vozila u kojemu nemamo predefinirane postaje, nego moramo odrediti i postaje koje ćemo posjetiti, te formirati autobusne linije što ovisi o mjesto stanovanja pojedinog učenika (na koju stanicu ide koji učenik) i kapaciteta autobusa.

Cilj rješavanja problema je:

1. odrediti skup stanica koje će se posjetiti,
2. odrediti za svakog učenika do koje stanice treba hodati,
3. odrediti autobusne linije, tako da se minimizira ukupna udaljenost koju autobusi prelaze.

Pritom vrijede neka **ograničenja i pretpostavke**:

1. svakom učeniku mora biti dodijeljena stanica čija je udaljenost manja od maksimalne dozvoljene udaljenosti,
2. svi autobusi imaju isti kapacitet koji ne smije biti premašen,
3. svaku postaju posjećuje samo jedan autobus, što znači da svi učenici koji idu na istu stanicu ne mogu ići različitim autobusima.

Početna i završna točka svih autobusnih linija mora biti škola.

2 OPIS ALGORITMA

Za rješavanje navedenog problema, korišten je hibridni genetski algoritam.

2.1 PRIKAZ RJEŠENJA

Svaki kromosom sastoji se od:

- zapisa autobusnih linija, permutacijski vektor cijelih brojeva koji predstavlja redoslijed obilaska stanica. Dekodira se u više linija tako da, kada se vidi da je premašen kapacitet autobusa, počinje nova linija, a sve stanice koje nisu posjećene se jednostavno preskaču.
- zapis dodjele stanica studentima, vektor cijelih brojeva koji označava koji student ide na koju stanicu.
- broj studenata po stanicama, pomoćni vektor cijelih brojeva koji olakšava dodjeljivanje stanica.

Table 2.1: Primjer kromosoma, kapacitet autobusa 3

1	3	2	0	5
---	---	---	---	---

Zapis autobusnih linija, dekodirat će se u dvije linije: 1, 3 -> 0

0	1	1	1	2
---	---	---	---	---

Dodjela stanica studentima, student 1 ide na stanicu 0, student 2 na stanicu 1 ...

1	3	1	0	0
---	---	---	---	---

Broj studenata po stanicama, stanica 0 ima 1 studenta, stanica 1 3 studenta i stanica 2 1 studenta

2.2 FUNKCIJA CILJA

Funkcija cilja je predstavljena zbrojem duljina svih linija (euklidska udaljenost). Kako nam je želja što manja ukupna udaljenost, ovo je problem minimizacije.

2.3 GENERIRANJE POČETNOG RJEŠENJA

Stvaranje početnog rješenja odvija se u dvije faze. U prvoj fazi, potrebno je dodijeliti stanice studentima, a druga faza odnosi se na određivanje autobusnih linija tako odabranom podskupu stanica.

2.3.1 DODJELJIVANJE STANICA STUDENTIMA

Za ovaj dio koristi se GRASP i to tako da se svakom studentu nasumično dodijeli jedna od α stanica najbližih školi (nije nam bitno koliko jadni studenti pješače).

2.3.2 ODREĐIVANJE LINIJA

I za ovaj dio se koristi GRASP, i to tako što se prvo postave linije tako da je jedna stanica jedna linija. Nakon toga spajamo linije uz heuristiku:

$$h(i, j) = distanceToSchool(i) + distanceToSchool(j) - euclideanDistance(i, j)$$

Nakon toga, temeljem heuristike spajamo nasumično jedan od α najboljih parova.

2.4 KRITERIJ ZAUSTAVLJANJA

Kriterij zaustavljanja algoritma je 5 minuta bez unaprjeđivanja rješenja, ili 60 minuta ukupnog trajanja izvođenja.

2.5 TIJEK ALGORITMA

Algoritam je blaga varijacija na klasični eliminacijski genetski algoritam što se može vidjeti iz pseudokoda.

Algorithm 1 Eliminacijski genetski algoritam

```
procedure HGA
  stvori početnu populaciju pomoću GRASP-a
  evaluiraj početnu populaciju
  while !kriterijZaustavljanja do
    if iter mod 10000 then
      ubaci novo GRASP rješenje u populaciju
    else
      turnirska selekcija
      križaj i mutiraj novu jedinku
      evaluiraj novu jedinku
      zamijeni najgoru jedinku turnira s novom jedinkom
    end if
  end while
  if nema napretka then
    stvori novu populaciju
    dodaj najbolje rješenje iz stare u novu populaciju
  end if
end procedure
```

2.6 GENETSKI OPERATORI

Genetski algoritam je vođen genetskim operatorima:

- Selekcija
- Križanje
- Mutacija

2.6.1 SELEKCIJA

Operator selekcije zadužen je za biranje jedinki iz populacije koje će ići dalje u reprodukciju (križanje i mutaciju), ali i odabir jedinke koja će biti zamijenjena rezultatom reprodukcije. Korištena selekcija za ovaj problem je k-turnirska selekcija.

K-turnirska selekcija radi tako što se nasumično iz populacije odabere k jedinki, najbolje 2 se odaberu za roditelje, dok se najlošija kasnije zamijeni rezultatom reprodukcije roditelja. U ovoj selekciji je bitan parametar k, što je k veći, to je i selekcijski pritisak veći.

2.6.2 KRIŽANJE

Operator križanja omogućuje kombiniranje 2 rješenja u jedno novo rješenje koje će imati značajke oba roditelja. Time omogućujemo finije pretraživanje prostora stanja. Važno je naglasiti, da se križanje ne mora nužno izvesti prilikom svake iteracije, tu uvodimo parametar *crxRate* koji označava vjerojatnost izvršenja križanja. Zaključak vezan za taj parametar je da *crxRate* treba biti što veći.

Za potrebe projekta implementirano je više vrsta križanja: *OrderedCrossover*, *CycleCrossover*, *AssignmentCrossover*. Križanje koje se izvodi u pojedinoj iteraciji je određeno nasumično. Moguće je i dodati različite vjerojatnosti ovisno o križanju, ali takav pristup nije pokazao značajnija poboljšanja za navedena križanja.

ORDEREDCROSSOVER je vrsta križanja u kojoj se nasumični podniz uzme od jednog roditelja, dok se ostali elementi unose onim redom kojim se pojavljuju u drugom roditelju.

ASSIGNMENTCROSSOVER je varijanta uniformnog križanja u kojem se s malom vjerojatnosti geni boljeg roditelja mijenjaju za gene lošijeg roditelja, tj. dodjela stanica studentima se uglavnom zadrži od boljeg roditelja, ali neki postotak može prijeći od lošijeg čime možda možemo dobiti bolja rješenja.

CYCLECROSSOVER je vrsta križanja gdje tražimo cikluse u roditeljima. Nađemo ciklus u prvom roditelju, prepíšemo elemente ciklusa u dijete (na istim pozicijama kao u roditelju). Ostale elemente prepíšemo iz drugog roditelja.

2.6.3 MUTACIJA

Operator mutacije omogućuje izbjegavanje lokalnih optimuma, tako da se nasumično izmijene geni jedinke. Važnost mutacije je time suprotna od križanja, omogućuje kaotičnije pretraživanje prostora stanja u nadi da će nas odvesti u bolji lokalni (ili globalni) optimum. I mutacija se, kao i križanje, ne izvodi pri svakoj iteraciji, već tu uvodimo parametar *mutRate* koji označava vjerojatnost izvršenja mutacije. Zaključak izveden iz ovog rada je da *mutRate* treba biti relativno malen.

Kao i za križanja, implementirano je više operatora mutacije: *ScrambleRoutesMutation*, *ScrambleAssignmentMutation*, *InversionRoutesMutation*, *StopsAssignmentMutation*.

SCRAMBLEROUTESMUTATION uzima nasumični podniz autobusnih linija i izmiješa ga.

SCRAMBLEASSIGNMENTMUTATION sa jako malom vjerojatnošću određuje hoće li pojedini gen mutirati i to tako da skloni studenta dodijeljenog s neke stanice i dodijeli mu drugu stanicu.

INVERSIONROUTESMUTATION uzima nasumični podniz autobusnih linija i invertira ga.

STOPSASSIGNMENTMUTATION bira jednu stanicu čija je posjećenost veća od 1, nakon toga ukloni nasumičan broj studenata sa stanice i dodijeli tim studentima novu stanicu.

3 ANALIZA REZULTATA

U genetskom algoritmu je bitno dobro namjestiti parametre, inače se algoritam može ponašati neželjeno. Najvažniji parametri su vjerojatnosti križanja i mutacije čiji utjecaj je jako velik. Npr. velika vjerojatnost mutacije, može upropastiti moguće dobro rješenje nastalo križanjem mnogo češće nego što će generirati kvalitetna rješenja. Slično vrijedi i za vjerojatnost križanja, ako stavimo premalu vjerojatnost križanja, prepušteni smo samo širokom pretraživanju stanja što donekle nalikuje na nasumično pretraživanje čime ne možemo jako daleko doći.

Prikazani parametri u tablici 3.1 su se pokazali kao jako dobrima za problem raspoređivanja školskih autobusa.

Table 3.1: Prikaz parametara

Naziv parametra	opis parametra	vrijednost
popSize	veličina populacije	30
crxRate	vjerojatnost križanja	0.995
mutRate	vjerojatnost mutacije	0.1
iterNoProgress	broj iteracija bez napretka nakon kojeg se osvježava populacija	100000

Algoritam je izvršen na 10 instanci u trajanju od 1 minute, trajanju od 5 minuta te neograničenom trajanju. Dobiveni rezultati su vidljivi u tablici 3.2.

Table 3.2: Rezultati

instanca	lmin	5min	ne
sbr1	142.23	142.23	140.40
sbr2	130.17	123.19	120.68
sbr3	2775.98	2754.67	2670.32
sbr4	1819.78	1819.78	1677.41
sbr5	2116.22	2090.02	2013.27
sbr6	1630.44	1427.86	1334.69
sbr7	1313.27	1305.82	1278.69
sbr8	869.65	818.63	810.12
sbr9	325.99	325.99	325.77
sbr10	149.55	149.55	143.88

4 ZAKLJUČAK

U ovom izvještaju se mogla vidjeti primjena genetskog algoritma na problem raspoređivanja školskih autobusa. Ovaj problem je varijacija na problem usmjeravanja vozila, uz razliku što postaje koje treba posjetiti nisu statične, već treba odrediti i optimalnu dodjelu postaja studentima. Samim time je problem znatno teži od usmjeravanja vozila, kao i problema trgovačkog putnika i pripada razredu **NP-teških** problema.

Genetski algoritam je u prošlosti primjenjen na velik broj optimizacijskih problema, i to uspješno, što me navelo na korištenje istog na ovaj problem. Zbog jednostavnosti, omogućuje izvođenje velikog broja iteracija u kratkom vremenu. Iz rezultata se isto može vidjeti da jako brzo dođe do konačnog rješenja, u pravilu mu povećanje vremena nije toliko bitno, već u 5 minuta dođe do zadovoljavajućih vrijednosti funkcije cilja. Ovo može biti i problem, i značiti da rano zapne u lokalnom optimumu, ali obzirom na dobivena rješenja, izgleda zadovoljavajuće.

Moguća poboljšanja algoritma su novi operatori križanja, koji će uzimati u obzir i autobusne linije, pokušati na pametan način kombinirati linije, ali i dodjele stanica studentima, kako bi dobili što kraću i što popunjeniju liniju. Isto tako, mogući su i drugi operatori mutacija, koji će također pratiti optimalne linije, dodjeljivanja stanica studentima i ovisno o tome raditi ne tako nasumično pretraživanje stanja. Dodatno, moguće je i dodati više vrsta generiranja početnog rješenja, čime bi se uvelike povećala raznolikost početne populacije što povećava vjerojatnost genetskom algoritmu za pronalazak boljih rješenja.