Nikola Bukovac, 0036470630

Izvještaj za projekt iz Heurističkih metoda optimizacija

Opis problema

Za rješavanje zadanog problema potrebno je odrediti iz skupa mogućih autobusnih stanica, one koje će biti uključene u prometnu mrežu, za svakog učenika odrediti na kojoj će od stanica, koje su mu unutar radijusa kretanja, čekati autobus, pritom pazeći da broj učenika na toj stanici ne prelazi kapacitet autobusa s obzirom da samo jedan autobus dolazi na svaku stanicu te odrediti autobusne linije pritom minimizirajući ukupnu udaljenost koju će autobusi proći.

Opis algoritma

Zadani problem je riješen pomoću eliminacijskog genetskog algoritma koji raspoređuje učenike po stanicama uz minimiziranje funkcije cilja u kojoj se pohlepnim algoritmom stvaraju autobusne linije na način da se prvo kreće od stanice najudaljenije od škole te se približavajući školi staje uz stanice čija je udaljenost od zadnje posjećene do trenutne manje nego što je udaljenost od zadnje posjećene i škole, uz dovoljno mjesta za smjestiti sve učenike sa stanice.

Prikaz rješenja, iako je modeliran sa složenim objektima moguće da je protumačiti kao polje cijelih brojeva gdje indeks polja predstavlja Id pojedinog učenika, a vrijednost polja stanicu na kojoj čeka autobus.

Funkcija cilja zbroj udaljenosti koje prolaze autobusi na svom putu od škole po svojoj ruti te nazad prema školi.

Početno rješenje se dobiva tako što se svakom učeniku iz skupa stanica do koje može doći se odrediti da dođe na stanicu koja je što bliže školi na kojoj još uvijek ima mjesta. Korištenje genetskog algoritma s populacijom 30 uvjetuje da se kreira 30 početnih rješenja i od tih se uzima ono s najmanjom vrijednosti funkcije cilja.

Kriterij zaustavljanja algoritma je limit iteracija, koji je postavljen na vrijednost od 40 000 iteracija, u svakoj iteraciji se obavi jedna evaluacija funkcije cilja.

Genetski algoritam ima nekoliko specifičnosti koje utječu na izvršavanje algoritma, a to su sama vrsta algoritma, operatori križanja, selekcije i mutacije te parametri koji određuju vjerojatnost mutacije, veličina populacije te ostali.
Vrsta korištenog genetskog algoritma je eliminacijski genetski algoritam čime se automatski dobivaju još jedna dodatna heuristika elitizma, čime se osigurava da najbolje rješenje uvijek ostaje u populaciji.
Za operator selekcije je korištena k-turnirska selekcija uz postavljanje parametra k na 9. Ova selekcija slučajno odabire 9 jedinki te potom iz populacije briše najlošiju te vraća dvije najbolje.

Za operator križanja je napravljeno križanje koje uspoređuje učenike iz selektiranih jedinki te ukoliko je moguće nastalom dijetetu pridijeljuje stanicu koja je bliža školi.
Za operator mutacije je napravljena mutacija koja ukoliko je zadovoljena vjerojatnost mutacije svakom učeniku slučajno određuje jednu od stanica do koje može doći.
Vjerojatnost mutacije je postavljena na 0.01.

Pseudokod algoritma

```
Glavni program:
      parsiraj_ulaznu_datoteku();
      pokreni_genetski_algoritam();
      kreiraj_izlaznu_datoteku();

pokreni_genetski_algoritam():
      populacija = kreiraj_početnu_populaciju(veličina_populacije);

      dok iteracija != limitIteracija:
            selektirani = napravi_k-turnir_selekciju(populacija);
            dijete = križaj_selektirane(selektirani);
            dijete = mutiraj(dijete);

            best = pronađi_najbolju_jedinku_populacije();

      vrati best;

kreiraj_početnu_populaciju(veličina_populacije):
      populacija = [];

      dok populacija.size() < veličina_populacije:
            jedinka;

            za svakog učenika jedinke:
                  dodijeli_stanicu();

            populacija.dodaj(jedinka);

      vrati populacija;


napravi_k-turnir_selekciju(populacija):
      turnir = izaberi_slučajno_k_iz_populacije(populacija);
      izbaci_najlošijeg_u_turnira_iz_populacije(populacija, turnir);

      vrati dva_najbolja_u_turniru(populacija);

križaj_selektirane(selektirani):
      dijete;
      svakom učeniku iz dijete:
            dodijeli_stanicu_najbližu_školi_od_selektiranih();

      vrati dijete;

mutiraj(dijete):
      svakom učeniku iz dijete:
            slučajno_odaberi_stanicu();

      vrati dijete;
```

Rezultati

Ime datoteke	Funkcija cilja	Funkcija cilja evaluator	Broj evaluacija
res-1m-sbr1	291.802938689273	291.80295	14623
res-5m-sbr1	281.809994368507	281.81006	40030
res-ne-sbr1	281.809994368507	281.81006	40030
res-1m-sbr2	223.277027289522	223.27701	15045
res-5m-sbr2	222.620762087102	222.62076	40030
res-ne-sbr2	222.620762087102	222.62076	40030
res-1m-sbr3	2956.7482934173	2956.7483	11698
res-5m-sbr3	2956.7482934173	2956.7483	40030
res-ne-sbr3	2956.7482934173	2956.7483	40030
res-1m-sbr4	2026.38183974616	2026.3815	12119
res-5m-sbr4	2026.38183974616	2026.3815	40030
res-ne-sbr4	2026.38183974616	2026.3815	40030
res-1m-sbr5	2163.96840240511	2163.9675	11706
res-5m-sbr5	2163.96840240511	2163.9675	40030
res-ne-sbr5	2163.96840240511	2163.9675	40030
res-1m-sbr6	1533.96223542136	1533.962	12117
res-5m-sbr6	1533.96223542136	1533.962	40030
res-ne-sbr6	1533.96223542136	1533.962	40030
res-1m-sbr7	1648.50407242761	1648.5044	9859
res-5m-sbr7	1643.60961688559	1643.6101	40030
res-ne-sbr7	1643.60961688559	1643.6101	40030
res-1m-sbr8	1049.93975503535	1049.9398	10182
res-5m-sbr8	1002.48760893394	1002.48773	40030
res-ne-sbr8	1002.48760893394	1002.48773	40030
res-1m-sbr9	528.881260286651	528.88135	5486
res-5m-sbr9	501.91535426343	501.9154	32939
res-ne-sbr9	501.91535426343	501.9154	40030
res-1m-sbr10	304.61799533688	304.61798	6937
res-5m-sbr10	292.393629475155	292.39362	33105
res-ne-sbr10	289.627656421557	289.62762	40030

Kod većine instanci vrijeme izvođenja algoritma je trajalo manje od 5 minuta, osim kod instanci 9 i 10, a glavni faktor tomu je broj iteracija algoritma koji je bio limitiran na 40 000. Limitiranje na ovaj iteracija je odabrano zato što kod skoro svih instanci se većina poboljšanja optimuma dogodi u prvih 20 000 iteracija pa kasnije rijetko dolazi do promjene, a i promjena koja se dogodi je

poprilično mala pa bi povećanjem broja iteracija došlo do malog poboljšanja rezultata, ali ne značajnog.
Veličina populacije utječe na vrijeme izvođenja algoritma pa tako veća populacija povećava vrijeme izvođenja, vjerojatnost mutacije ne utječe na vrijeme izvođenja, ali ima utjecaja na brzinu konvergencije rješenja.

Zaključak

Implementirani algoritam uspješno pronalazi rješenja koja su vjerodostojna za zadane instance problema, ali moguće da postoje i optimalnija rješenja koja bi bilo moguće pronaći korištenjem drugačijih operatora za mutacije i križanje, ili dodatnim namještavanjem opisanih parametara. Najveća mogućnost poboljšavanja rezultata bi bila kod što boljeg početnog razmještaja učenika na stanice kako bi se što više smanjila razgranatost stanica i što više se približilo školi uz prilagođenije operatore mutacije i križanja.