



UNIVERZITET U BANJOJ LUCI  
PRIRODNO-MATEMATIČKI FAKULTET  
MATEMATIKA I INFORMATIKA, INFORMATIKA  
UVOD U VJEŠTAČKU INTELIGENCIJU



***METRIC DIMENSION PROBLEM***

Seminarski rad

Student:  
Nikola Đajić (7/19)

Predmetni profesor:  
doc. dr Marko Đukanović

Banja Luka, maj 2025

# S A D R Ź A J

<b>1. UVOD .....</b>	<b>2</b>
<b>2. PROBLEM METRIČKE DIMENZIJE GRAFOVA.....</b>	<b>3</b>
<b>3. PREGLED LITERATURE.....</b>	<b>5</b>
<b>4. GENETSKI ALGORITAM .....</b>	<b>6</b>
4.1. IMPLEMENTACIJA GENETSKOG ALGORITMA NA MDP .....	6
<b>5. ACO – OPTIMIZACIJA KOLONIJE MRAVA.....</b>	<b>8</b>
5.1. IMPLEMENTACIJA ACO ALGORITMA NA MDP .....	9
<b>6. REZULTATI .....</b>	<b>10</b>
6.1. POREĐENJE ALGORITAMA.....	12
<b>7. ZAKLJUČAK.....</b>	<b>14</b>
<b>LITERATURA .....</b>	<b>15</b>

# 1. UVOD

Grafovi predstavljaju jedan od osnovnih modela za analizu i reprezentaciju kompleksnih sistema u raznim oblastima, uključujući računarstvo, biologiju, društvene mreže i telekomunikacije. Jedan od problema teorije grafova jeste problem metričke dimenzije (eng. *Metric Dimension Problem – MDP*), koji se odnosi na određivanje minimalnog skupa čvorova, koje nazivamo rješavajućim skupom, tako da se svi preostali čvorovi u grafu mogu jednoznačno identifikovati na osnovu svoje udaljenosti do čvorova tog skupa. Ovaj problem je NP-težak, što znači da, ako pretpostavimo da hipoteza o SETH-u važi, ne postoji efikasan algoritam koji ga može riješiti za sve instance u polinomijalnom vremenu. Zbog toga se koriste različite heurističke i optimizacione metode, kako bi se pronašla optimalna ili približna rješenja.

U ovom radu implementirani su genetski algoritam (eng. *Genetic Algorithm – GA*) i algoritam optimizacije kolonijama mrava (eng. *Ant Colony Optimization – ACO*). Dobijeni rezultati su upoređeni sa algoritmima varijabilna pretraga susjedstva (eng. *Variable Neighborhood Search – VNS*) i cjelobrojno linearno programiranje (eng. *Integer Linear Programming – ILP*). Izvorni kod dostupan je putem javnog GitHub repozitorijuma, na linku: <https://github.com/NikolaDjajic/MDP>.

U nastavku rada predstavljena je formalna definicija problema, pregled relevantne literature, opis primijenjenih algoritama, kao i dobijeni rezultati.

## 2. PROBLEM METRIČKE DIMENZIJE GRAFOVA

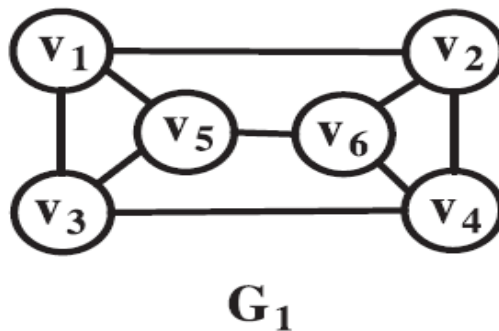
Metrička dimenzija grafova prvi put je formalno uvedena u radu Harary i Melter (1976), a kasnije su je detaljno analizirali Chartrand i saradnici 2000. godine. Problem metričke dimenzije predstavlja značajan aspekt teorije grafova, sa primjenama u različitim oblastima kao što su navigacija robota, lociranje grešaka u mrežama, identifikacija čvorova u komunikacionim sistemima i bioinformatika.

Neka je  $G=(V, E)$  neprazan, cikličan i povezan graf. Za skup čvorova  $W \subseteq V$ , kažemo da  $W$  „rješava“ graf  $G$  ako za svaki par različitih čvorova  $u, v \in V$  postoji čvor  $w$ , takav da  $w \in W$  i  $d(u, w) \neq d(v, w)$ . Drugim riječima, svaka dva čvora u grafu se mogu razlikovati na osnovu svojih udaljenosti do bar jednog čvora iz skupa  $W$ .

Za dati skup čvorova  $S=\{s_1, s_2, \dots, s_k\}$ , vektor rastojanja za čvor  $v$  definiše se kao:

$$r(v, S) = (d(v, s_1), d(v, s_2), \dots, d(v, s_k)),$$

pri čemu je  $d(v, s_i)$  najkraće rastojanje između čvorova  $v$  i  $s_i$ . Skup  $S$  je skup rješenja (eng. *Resolving Set*) ako su svi vektori  $r(v, S)$  jedinstveni za svako  $v \in V \setminus S$ . Najmanji takav skup označava se kao minimalni rješavajući skup, a njegova kardinalnost predstavlja metričku dimenziju grafa  $G$ , označenu kao  $\beta(G)$ .



Slika 1. Primjer grafa  $G_1$

Na Slici 1, prikazan je graf  $G_1$ , nad kojim će biti prikazan problem metričke dimenzije. Graf  $G_1$  posjeduje 6 čvorova i 9 ivica. Rastojanje između dva povezana čvora je uvijek jednako jedan.

Skup  $S_1 = \{v_1, v_2, v_3\}$  je jedan od rješenja, zato što:

- $r(v_1, S_1) = (0, 1, 1)$ ;
- $r(v_2, S_1) = (1, 0, 2)$ ;
- $r(v_3, S_1) = (1, 2, 0)$ ;
- $r(v_4, S_1) = (2, 1, 1)$ ;
- $r(v_5, S_1) = (1, 2, 1)$ ;

- $r(v_6, S_1) = (2, 1, 2)$ .

$r(v_i, S)$  je rastojanje od čvora  $v_i$  do svakog od čvorova iz skupa  $S$ , te je isto što i:  $r(v_i, \{v_1, v_2, v_3\})$ .

Kao što se vidi, ne postoje dva identična skupa što znači da pomoću  $S_1$  možemo jednoznačno identifikovati svaki čvor koji nije u  $S_1$  na osnovu njegove udaljenosti od čvorova  $S_1$ . Skup  $S_1$  sadrži 3 čvora pa je i kardinalnost skupa jednaka 3.

Međutim skup  $S_1$  nije minimalni rješavajući skup, zato što postoji i  $S_2 = \{v_1, v_3\}$  koji je takođe skup rješenja.

- $r(v_1, S_2) = (0, 1)$ ;
- $r(v_2, S_2) = (1, 2)$ ;
- $r(v_3, S_2) = (1, 0)$ ;
- $r(v_4, S_2) = (2, 1)$ ;
- $r(v_5, S_2) = (1, 1)$ ;
- $r(v_6, S_2) = (2, 2)$ .

$S_2$  predstavlja bolje rješenje u odnosu na  $S_1$ , zato što je kardinalnost skupa jednaka 2. Takođe  $S_2$  nije jedinstveno minimalno rješenje pošto postoji i skup  $S_3 = \{v_2, v_4\}$  kojem je kardinalnost jednaka isto 2 kao i skupu  $S_2$ .

### 3. PREGLED LITERATURE

U ovom poglavlju prikazani su relevantni radovi koji se bave problemom metričke dimenzije (Tabela 1).

*Tabela 1. Pregled relevantne literature*

Harary i Melter [1]	Prvi put formalno uvedena metrička dimenzija.
Chartrand i saradnici [2]	Detaljnija analiza metričke dimenzije.
Kratica i saradnici [3]	Predložen je prvi metaheuristički pristup problemu metričke dimenzije grafova. Genetski algoritam predstavljen u ovom radu koristi binarno kodiranje i standardne genetske operatore prilagođene problemu.
Chartand i saradnici [2]; Currie i Oellermann [4]	Formulacije problema metričke dimenzije pomoću ILP-a
Kratica i saradnici [5]	ILP formulacija za problem minimalnog dvostrukog skupa razrješenja <sup>1</sup> .
Peters-Fransen i Oellermann [6]; Cáceres i saradnici [7]	Bavili su se određivanjem donjih i gornjih granica za metričku dimenziju Karatezijanovog produkta grafova.
Yero i saradnici [8]	Istražili metričku dimenziju korona produkta grafova, te dali teorijske granice
Hernando i saradnici [9] [10]	Istraživali vezu između metričke dimenzije i drugih grafičkih invarijanti, kao što su prečnik, broj čvorova i stepen čvorova.
Chappell i saradnici [11]	Uspostavili relaciju između metričke i particione dimenzije grafa.
Bailey i Meagher [12]	Analizirali su metričku dimenziju Grassmannovih grafova.
Bailey i Cameron [13]	Ispitali metričke dimenzije Johnsonovih i Knezerovih grafova.
Javaid i saradnici [14]; Husnine i Kousar [15]	Istraživali metričke dimenzije generalizovanih Petersonovih grafova.
Cáceres i saradnici [16]; Rebatel i Thiel [17]	Pokazali da neki beskonačni grafovi imaju beskonačnu metričku dimenziju.
Beerliova i saradnici [18]	Primijenili su MDP u domenu telekomunikacija.
Chartrand i saradnici [2]	Predložili su primjenu MDP u hemiji, za identifikaciju jedinstvenih pozicija u molekulskim strukturama.
Khuller i saradnici [19]	Primijenili su MDP u navigaciji robota.

<sup>1</sup> Za čvorove  $x$  i  $y$  grafa  $G$  kaže se da dvostruko razrješavaju čvorove  $u$  i  $v$  grafa  $G$  ako važi:  $d(u, x) - d(u, y) \neq d(v, x) - d(v, y)$ . Skup čvorova  $D$  grafa  $G$  je dvostruko rješavajući skup ako svaka dva različita čvora grafa  $G$  budu dvostruko razrješena nekim dva čvora iz  $D$ .

## 4. GENETSKI ALGORITAM

Genetski algoritam pripada heurističkim metodama (metode koje ne garantuju optimalno rješenje). Inspiracija za ovaj algoritam leži u Teoriji evolucije Čarlsa Darvina.

Neki od pojmova koje bi trebalo da definišemo kada pričamo o genetskom algoritmu:

- Jedinka = svaka jedinka je moguće rješenje problema.
- Populacija = skup jedinki koji predstavlja podskup prostora pretrage.
- Prilagođenost (eng. fitness) = ocjena kvaliteta date jedinke.
- Genetski kod = način na koji se predstavlja jedna jedinka.
- Geni = dijelovi genetskog koda.

Takođe, postoje i genetski operatori, a to su:

### *Selekcija*

Ovim procesom se biraju jedinke nad kojima će se poslije vršiti ukrštanje i mutacija. Jedinke se biraju na osnovu individualne prilagođenosti, one sa većom prilagođenošću su u prednosti da budu odabrane za nastavak. Neke od načina selekcije su ruletska, turnirska, eliminaciona i druge.

### *Ukrštanje*

Slično kao i u samoj evoluciji, ukrštanje predstavlja miješanje gena jedinki. Kao rezultat ove operacije dobijamo novu jedinku koja nosi potencijalno dobre gene svojih roditelja.

### *Mutacija*

Ovaj proces predstavlja mijenjanje gena. Time se omogućava vraćanje dobrih gena koji su se izgubili prilikom prethodnih radnji. Posljedica izvršavanja prethodno definisanih radnji jeste nastanak nove generacije.

### 4.1. IMPLEMENTACIJA GENETSKOG ALGORITMA NA MDP

Implementacija genetskog algoritma za MDP u ovom radu nešto se razlikuje od klasične implementacije genetskog algoritma, te je možemo nazvati i „izmijenjenim genetskim algoritmom“. Početnu populaciju u početku smo birali nasumično (eng. *random*), tako da svaki čvor ima 50% šanse da se nađe u početnom skupu. Zatim smo koristili pohlepnu (eng. *greedy*) tehniku za generisanje početne populacije. Čvorovi se sortiraju po opadajućem stepenu, nakon čega se iterativno dodaju u skup sve dok ne dobijemo validan rješavajući skup.

Za selekciju je korištena težinska selekcija, ona predstavlja vrstu selekcije koja bira jedinke (rješenja) za sljedeću generaciju na osnovu njihove *fitness* vrijednosti. U slučaju jedinki s istim *fitness* vrijednostima one dobijaju istu težinu, a izbor između njih se obavlja nasumično. Implementirano je i jednopoziciono i dvopoziciono ukrštanje, te se ispostavilo da je dvopoziciono dovelo do boljih rezultata.

Mutacija je izmijenjena tako da ukoliko gen mutira on može samo da mutira nabolje, to jeste da taj čvor koji je bio u datoj jedinki ispadne iz nje. Ovim pristupom neće se moći vratiti neki gen ukoliko je odbačen, ali će se jako brzo smanjivati broj čvorova u jednikama što je nama i potrebno, s činjenicom da se kreće sa jako lošim početnim rješenjem. Ovakav način mutacije dao je neuporedivo bolje rezultate i brže vrijeme izvršavanja algoritma.

Pomoću *fitness* funkcije procjenjuje se kvalitet datog rješenja. To se vrši tako što se provjerava da li postoje dva čvora u grafu koja imaju identična rastojanja do svih čvorova iz rješavajućeg skupa. Ukoliko ne postoje vraćamo dužinu tog rješenja kao *fitness* vrijednost, međutim ako postoji onda isto tako vraćamo dužinu rješenja *plus 100 \* broj nerješenih parova*. Time dajemo jako veliku *fitness* vrijednost rješenjima koja ne predstavljaju rješavajući skup.

Sve najkraće puteve između čvorova smo unaprijed izračunali kako bismo smanjili vrijeme izvršavanja algoritma.

Glavne razlike između ovog genetskog algoritma i onog koji se koristi u radu za poređenje sa implementiranim VNS algoritmom su:

- U *fitness* funkciji kod nas se traži da li je to rješavajući skup, ukoliko nije dodajemo veliku penalizaciju, dok u radu ukoliko nije rješavajući skup dodaju se nasumični čvorovi sve dok skup ne bude rješavajući.
- U radu se upotrebljava *fine-grained* turnirska selekcija, dok se u našem genetskom algoritmu koristi težinska selekcija.
- U našem genetskom algoritmu postoji posebna mutaciju, koja je ranije objašnjena, dok se u radu pažnja posvećuju “zamrznutim” bitovima (oni koji se ne mijenjaju tokom evolucije rješenja).



## 5. ACO – OPTIMIZACIJA KOLONIJE MRAVA

ACO je heuristički optimizacioni algoritam inspirisan ponašanjem stvarnih mrava u prirodi, posebno njihovom sposobnošću da pronalaze najkraći put između izvora hrane i mravinjaka. Algoritam je prvi put predstavio Marco Dorigo 1992. godine u okviru svoje doktorske disertacije.

Osnovne ideje i pojmovi:

- Mrav = predstavlja agenta pojedinačnog rješenja problema;
- Kolonija = skup svih mrava koji se kreću kroz prostor pretrage;
- Feromoni = vještačka “hemijska” oznaka koja se koristi za memorisanje prethodnih dobrih rješenja. Mravi više preferiraju puteve s većom koncentracijom feromona, i
- Heuristička informacija = dodatna informacija o kvalitetu rješenja, npr. inverzna udaljenost kod problema najkraće putanje.

Glavni procesi u ACO algoritmu:

### *Kretanje mrava kroz prostor rješenja*

Svaki mrav gradi rješenje korak po korak, birajući naredni korak na osnovu kombinacije količine feromona i heurističke vrijednosti (npr. dužine puta). Ova selekcija je probabilistička, što znači da bolje opcije imaju veću šansu da budu izabrane, ali i lošije se povremeno istražuju – radi očuvanja raznolikosti.

### *Ažuriranje feromona*

Nakon što svi mravi završe konstruisanje svojih rješenja, feromoni se ažuriraju. Dobri putevi (bolja rješenja) dobijaju više feromona, dok stari feromoni postepeno isparavaju. Ova ravnoteža između ojačavanja dobrih puteva i isparavanja loših sprečava preuranjenu konvergenciju. Postoji nekoliko metoda, a neke od njih su:

- Ažuriranje feromona kod najboljeg mrava(ovaj pristup smo mi koristili).
- Ažuriranje feromona kod k najboljih mrava.
- Ažuriranje feromona kod svih mrava.

### *Isparavanje feromona*

Ova operacija smanjuje vrijednosti feromona na svim putevima kako bi se stari, neefikasni putevi vremenom zaboravili. Time se podstiče istraživanje novih rješenja.

## 5.1. IMPLEMENTACIJA ACO ALGORITMA NA MDP

Implementacija ACO algoritma na MDP u ovom radu započinje tako što se svakom čvoru u grafu dodijeli ista količina feromona. Pored toga izračunava se heuristička vrijednost svakog čvora kao njegova *betweenness centrality* – što je čvor “važniji” za povezivanje ostalih, to mu heuristika daje veću privlačnost. Postavljaju se parametri tako da feromon i heuristika posjeduju različitu važnost (*alpha* i *beta*), te se potom određuje koliko brzo feromoni isparavaju (eng. *evaporation*), a zatim se određuje i ukupna količinu feromona koja se dodjeljuje uspješnim rješenjima (Q).

$$p_{ij} = \frac{(\tau_{ij}^{\alpha}) \cdot (\eta_{ij}^{\beta})}{\sum_{j \in \Lambda} (\tau_{ij}^{\alpha}) \cdot (\eta_{ij}^{\beta})}$$

Vjerovatnoća prelaska mrava iz čvora  $i$  u čvor  $j$ .

$p_{ij}$  – vjerovatnoća da mrav iz čvora  $i$  odabere čvor  $j$  kao sljedeći.

$\tau_{ij}$  – količina feromona na ivici između čvorova  $i$  i  $j$ .

$\eta_{ij}$  – heuristička vrijednost ivice  $i \rightarrow j$ .

$\alpha$  – parametar koji određuje uticaj feromona. Ako je  $\alpha = 0$ , feromoni se ignorišu.

$\beta$  – parametar koji određuje uticaj heuristike. Ako je  $\beta = 0$ , heuristika se ignoriše.

$\Lambda$  – skup svih čvorova koje mrav iz čvora  $i$  može posjetiti.

Takođe bira se broj mrava i broj iteracija koje će simulirati. Vrijednosti svih ovih parametara zavise od datog grafa.

Svaki mrav gradi svoje rješenje iterativnim izborom čvorova: u svakom koraku posmatra preostale čvorove i bira ih na osnovu kombinovane ocjene feromona i heurističke vrijednosti. Čvorovi s jačim feromonskim tragom i višom heurističkom vrijednošću imaju veću vjerovatnoću da budu odabrani.

Kvalitet svakog rješenja procjenjuje se pomoću iste logike kao u genetskom algoritmu: mjeri se veličina odabranog skupa, uz snažnu kaznu za svaki par čvorova koji ostane nerješiv. Time se osigurava da se prvo traže bolja rješenja, a potom se smanjuje njihova veličina.

Nakon što svi mravi naprave svoja rješenja u jednoj iteraciji, traži se najbolji među njima. Nismo koristili dodatnu strategiju za razbijanje jednakih vrijednosti *fitness*-a. Zatim isparava dio feromona sa svih čvorova kako bismo održali raznovrsnost pretrage, i na čvorove iz najboljeg rješenja dodajemo novu dozu feromona – ta doza je veća ako je rješenje bolje. Kao i kod genetskog algoritma svi najkraći puteve između čvorova unaprijed su izračunati.

## 6. REZULTATI

Svaki algoritam pokretan je po nekoliko puta, a kriterijum zaustavljanja je da ukoliko u određenom broju iteracija (koje smo mijenjali u zavisnosti od samog grafa) ne dođe do poboljšanja prekinemo sa radom ili ukoliko istekne vremensko ograničenje od 10 minuta.

*Tabela 2. Eksperimentalni rezultati*

Grafovi	VNS			GA		ACO		PULP1		PULP2	
Naziv	V	E	$d_{avg}$	$\beta(G)$	t	$\beta(G)$	t	$\beta(G)$	t	$\beta(G)$	t
Graf-10	10	18	1,8	<b>4</b>	0.01	<b>4</b>	0.01	<b>4</b>	0.01	<b>4</b>	0.04
Graf-11	11	20	1,81	<b>4</b>	0.01	<b>4</b>	0.01	/	/	<b>4</b>	0.04
Graf-12	12	40	3,3	<b>3</b>	0.01	<b>3</b>	0.01	<b>3</b>	0.01	<b>3</b>	0.06
Graf-14	14	44	3	<b>3</b>	0.01	<b>3</b>	0.01	<b>3</b>	0.01	<b>3</b>	0.05
Graf-17	17	38	2,2	<b>2</b>	0.03	<b>2</b>	0.02	<b>2</b>	0.01	<b>2</b>	0.07
johnson8-2-4	28	420	15	<b>6</b>	0.07	<b>6</b>	0.14	<b>6</b>	0.1	<b>6</b>	6.6
MANN-a9	45	918	40	<b>16</b>	0.8	<b>16</b>	13.2	17	1.2	<b>16</b>	21.9
hamming6-4	64	704	22	<b>7</b>	1	7	11	7	0.7	/	600
hamming6-2	64	2000	57	<b>17</b>	1.7	22	4	20	3.1	/	600
johnson8-4-4	70	2000	53	<b>10</b>	1	11	14.7	12	1.55	/	600
keller4	171	9000	110	<b>12</b>	4	15	74.5	14	14.5	/	600
hamming8-4	256	21000	164	<b>14</b>	14.3	19	121	17	60	/	600
Csp50	53	176	6.6	<b>8</b>	0.4	9	2	11	1.4	<b>8</b>	1.5
Csp100	106	722	13.6	11	10	15	45.3	16	61.8	<b>10</b>	592
Csp150	401	1505	7.5	<b>13</b>	6.47	18	37.4	20	12.1	/	600
Gcol1	100	2487	49.7	<b>9</b>	1.4	10	4.8	10	2	/	600
Gcol5	100	2450	49	<b>8</b>	8.86	9	9.4	9	1.9	/	600
Gcol11	100	2467	49.3	<b>9</b>	1.2	10	4.2	10	2.27	/	600
Gcol15	100	2528	50.5	<b>9</b>	1.2	10	3.5	10	2.4	/	600
Gcol18	100	2472	49.4	<b>8</b>	1.72	9	26.6	10	2.3	/	600
Gcol20	100	2420	48.4	<b>9</b>	1.37	10	5.1	10	2.5	/	600
Gcol23	300	22393	149.2	<b>12</b>	21.8	13	73.4	13	89.3	/	600
Gcol24	300	22446	149.6	<b>12</b>	15.5	14	74.5	13	66.1	/	600
Gcol26	300	22601	150.6	<b>12</b>	16	14	65.8	13	70	/	600
Gcol30	300	22543	150.2	<b>12</b>	17.3	14	52.4	13	45	/	600

Tabela sadrži sljedeće podatke:

- $|V|$  – broj čvorova,
- $|E|$  – broj grana,
- $d_{avg}$  – prosječan stepen čvora,
- $\beta(G)$  – metrička vrijednost grafa(vrijednost koju tražimo),
- $t$  – vrijeme izvršavanja algoritma u sekundama.

U Tabeli 2. su prikazani rezultati pet algoritama – VNS, GA, ACO, PULP1 i PULP2 – primijenjenih na različite grafove. Algoritmi GA (genetski algoritam) i ACO (algoritam mravlje kolonije) su implementirani u ovom radu, dok su algoritmi VNS, PULP1 i PULP2 implementirani na osnovu ideja iz rada [20].

Parametre genetskog algoritma podešavali smo eksperimentalno u zavisnosti od veličine grafa. Za manje grafove koristili smo manju populaciju, manji broj generacija i nižu stopu mutacije kao i viši faktor zaustavljanja. Kod većih grafova povećavali smo broj jedinki i generacija, kao i stopu mutacije, kako bismo obezbijedili veću raznovrsnost i izbjegli lokalne optimume u znatno većem prostoru pretrage.

Parametre ACO algoritma takođe smo prilagođavali zavisno od veličine grafa, na osnovu nekoliko testiranja. Za manje grafove koristili smo manji broj mrava i iteracija, uz veći uticaj heurističke komponente. Kod većih grafova povećavali smo broj mrava i iteracija, smanjivali značaj heuristike u korist feromona, te ubrzali isparavanje tragova kako bismo omogućili bolje istraživanje kompleksnijeg prostora rješenja.

U oba slučaja smo puštali po nekoliko puta algoritme sa različitim vrijednostima parametara, a kao rezultat uzimali najbolje rješenje, ukoliko ih ima više onda je birano ono sa najbržim vremenom izvršavanja.

Specifikacije uređaja na kojem su navedeni eksperimenti rađeni navedene su u Tabeli 3.

*Tabela 3. Specifikacije uređaja*

CPU	Intel(R) Core(TM) 5 120U
CPU frekvencija	1.40GHz
CPU jezgra	10
RAM	16 GB DDR5
OS	Windows 11
Programski jezik	Python

## 6.1. POREĐENJE ALGORITAMA

### *Manje instance*

Na manjim grafovima svi algoritmi postižu odlične rezultate. GA i ACO se pokazuju kao konkurentni sa VNS, PULP1 i PULP2, ostvarujući iste vrijednosti ciljne funkcije, uz slično vrijeme izvršavanja. To ukazuje da su implementacije ispravne i daju kvalitetna rješenja na jednostavnijim instancama. Primjenom ACO algoritma rješenje za instanci graf-11 nije pronađeno u zadatom vremenskom opsegu.

### *Srednje instance*

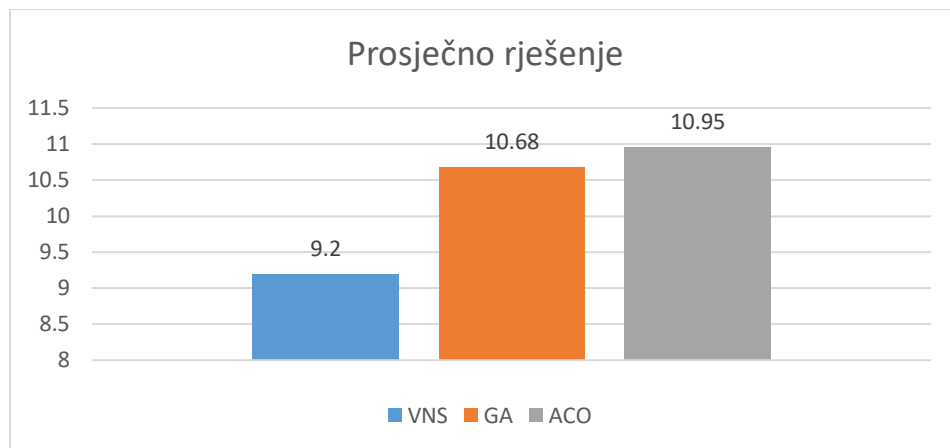
Kod srednje velikih instanci, GA i ACO ostvaruju rješenja malo lošija nego VNS, uz duže vrijeme izvršavanja. ACO pokazuje nešto slabije rezultate od GA na nekim instancama (npr. hamming6-4). Kod instanci kao što su Csp100 i Csp150, GA se pokazuje kao uspješniji u poređenju s ACO, dok PULP metode nisu uspjele da pronađu rješenje u zadatom vremenskom ograničenju od 10 minuta.

### *Velike instance*

Na velikim instancama, GA i ACO uspjevaju da pronađu rješenja, ali su ona lošijeg kvaliteta u odnosu na VNS. ACO i GA imaju slično vrijeme izvršavanja, ali ACO daje nešto bolje rezultate.

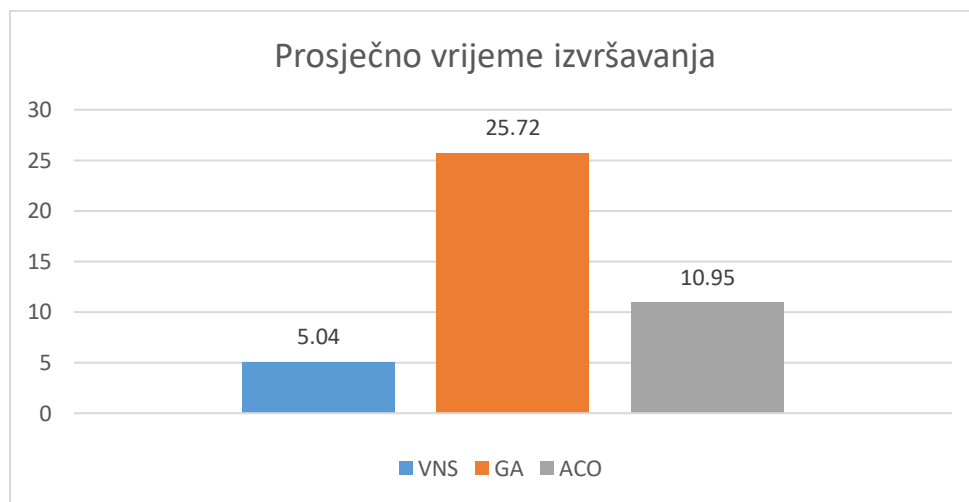
Važno je dodati da VNS algoritam, koji smo implementirali na osnovu rada [20], posjeduje nedostatak koji se ogleda u tome da u nekim slučajevima ne dobijemo skup koji je rješavajući ili najbolje rješenje za taj graf. Razlog tome je način na koji je napisan, to jeste ovaj algoritam ne čuva najbolje rješenje već na kraju zadatog broja iteracija vraća posljednje rješenje. Tu ponekad dolazi do greške, te postoji mogućnost da dobijeni skup ne predstavlja najbolje dobijeno rješenje, već rješenje generisano u posljednjoj iteraciji.

Oba PuLP rješavača su davali najbolje rezultate, dok su mogli da pronađu rješenje u roku od 10 minuta. Kao što se može vidjeti u tabeli to se desilo za samo nekoliko manjih grafova. Takođe PULP1 se pokazao dosta brži od PULP2 algoritma.



*Slika 2. Grafički prikaz prosječnog rješenja za algoritme.*

Na Slici 2. prikazan je grafički prikaz prosječnog rješenja za algoritme. Posmatrajući prosjek jasno se uočava da se VNS algoritam pokazao najboljim. Takođe vidimo da su ACO i GA dali slične rezultate.



*Slika 3. Grafički prikaz prosječnog vremena izvršavanja algoritama.*

Na Slici 3. prikazan je grafik koji predstavlja prosječno vrijeme izvršavanja algoritama. Ovdje takođe vidimo dominaciju VNS algoritma koji se pokazao najbržim, ali i to da je genetski uzimao dosta više vremena u odnosu na ACO a davao je slične rezultate.

## 7. ZAKLJUČAK

U ovom radu istražen je NP-težak problem metričke dimenzije grafova čiji je cilj pronaći najmanji rezolventni skup čvorova tako da se svaki par čvorova može jednoznačno razlikovati prema udaljenosti do članova tog skupa.

Implementirali smo dva PuLP rješavača, te jedan VNS algoritam, koji se bave rješavanjem ovog problema čija je ideja implementacije preuzeta iz rada [20], a zatim i naše dvije ideje za rješavanje istog problema, genetski algoritam i algoritam mravljih kolonija.

Kao što smo vidjeli PuLP rješavačem postignuta su odlična rješenja, međutim ograničen je u smislu mogućnosti rješavanja problema na jako malim grafovima. VNS se pokazao kao najbolji algoritam u pogledu rješenja i vremena izvršavanja. Nedostatak VNS-a je što ponekad kao rezultat izvršavanja algoritma nismo dobijali rješavajući skup.

Genetski i ACO postigli su poprilično ujednačene rezultate, iako je genetski bio dosta bolji kod manjih grafova, gdje je ACO imao dosta problema. Isto tako se ACO pokazao kao dosta bolja opcija za veće grafove gdje su rezultati bilo malo lošiji od VNS algoritma. Moramo dodati da je GA uzimao dosta više vremena.

Rješavanje problema ACO algoritma sa određenim istancama manjih grafova, kao i ubrzavanje genetskog algoritma tako što će počinjati sa dosta boljim početnim rješenjima je nešto što bi se moglo usavršiti u budućim radovima.

## LITERATURA

- [1] Harary, F., Melter, R., 1976. On the metric dimension of a graph. *Ars Combinatoria* 2, 191–195.
- [2] Chartrand, G., Eroha, L., Johnson, M., Oellermann, O., 2000. Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics* 105, 99– 113.
- [3] Kratica, J., Kovačević -Vujčić, V., Čangalović, M., 2009. Computing the metric dimension of graphs by genetic algorithms. *Computational Optimization and Applications* 44, 343–361.
- [4] Currie, J., Oellerman, O., 2001. The metric dimension and metric independence of a graph. *Journal of Combinatorial Mathematics and Combinatorial Computing* 39, 157–167.
- [5] Kratica, J., Čangalović, M., Kovačević -Vujčić, V., 2009b. Computing minimal doubly resolving sets of graphs. *Computers & Operations Research* 36, 2149–2159.
- [6] Peters-Fransen, J., Oellermann, O., 2006. The metric dimension of cartesian products of graphs. *Utilitas Mathematica* 69, 33–41.
- [7] Cáceres, J., Hernando, C., Mora, M., Pelayo, I., Puertas, M., Seara, C., Wood, D., 2007. On the metric dimension of cartesian products of graphs. *SIAM Journal in Discrete Mathematics* 21, 423–441.
- [8] Yero, I., Kuziak, D., Rodriguez-Velazquez, J., 2011. On the metric dimension of corona product graphs. *Computers and Mathematics with Applications* 61, 2793–2798.
- [9] Hernando, C., Mora, M., Pelayo, I., Seara, C., Wood, D., 2007. Extremal graph theory for metric dimension and diameter. *Electronic Notes in Discrete Mathematics* 29, 339–343.
- [10] Hernando, C., Mora, M., Pelayo, I., Seara, C., Wood, D., 2010. Extremal graph theory for metric dimension and diameter. *Electronic Journal of Combinatorics* 17, 1– 28.
- [11] Chappell, G., Gimbel, J., Hartman, C., 2008. Bounds on the metric and partition dimensions of a graph. *Ars Combinatoria* 88, 349–366.
- [12] Bailey, R., & Meagher, K., 2011. On the metric dimension of Grassmann graphs. Technical Report. arXiv:1010.4495.
- [13] Bailey, R., Cameron, P., 2011. Base size, metric dimension and other invariants of groups and graphs. *Bulletin of the London Mathematical Society* 43, 209–242.
- [14] Javaid, I., Rahim, M., Ali, K., 2008. Families of regular graphs with constant metric dimension. *Utilitas Mathematica* 75, 21–33.
- [15] Husnine, S., Kousar, I., 2010. A subfamily of a generalized Petersen graph  $p(n, 3)$  with constant metric dimension. *Utilitas Mathematica* 81, 111–120.



- [16] Cáceres, J., Hernando, C., Mora, M., Pelayo, I., Puertas, M., 2009. On the metric dimension of infinite graphs. *Electronic Notes in Discrete Mathematics* 35, 15– 20.
- [17] Rebatel, F., Thiel, E., 2011. Metric bases for polyhedral gauges. *Lecture Notes in Computer Science* 6607, 116–128.
- [18] Beerliova, Z., Eberhard, F., Erlebach, T., Hall, A., Hoffmann, M., Mihalák, M., Ram, L., 2006. Network discovery and verification. *IEEE Journal on Selected Areas in Communications* 24, 2168–2181.
- [19] Khuller, S., Raghavachari, B., Rosenfeld, A., 1996. Landmarks in graphs. *Discrete Applied Mathematics* 70, 217–229.
- [20] Nenad Mladenović, Jozef Kratica, Vera Kovačević-Vujčić, Mirjana Čangalović, Variable neighborhood search for metric dimension and minimal doubly resolving set problems, *European Journal of Operational Research*, <https://doi.org/10.1016/j.ejor.2012.02.019>