

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNASRTVA

**Pregled algoritma pretraživanja A\***  
**za globalno planiranje putanje plovila na**  
**mapi troškova**

Enio Krizman  
0069083848

Zagreb, siječanj 2023.

# Sadržaj

1. Uvod .....	1
2. Planiranja putanje pomoću A* algoritma u simulacijskom okruženju .....	2
2.1. Opis primjene A* algoritma na mapi troškova (eng. Cost map) .....	3
2.2. Demonstracija pronalaska putanje pomoću A* algoritma na osnovnoj mapi .....	5
2.3. Demonstracija pronalaska putanje pomoću A* algoritma na mapi Sjevernog hrvatskog primorja .....	6
3. Zaključak i naredni koraci .....	8
4. Literatura .....	9

## 1. Uvod

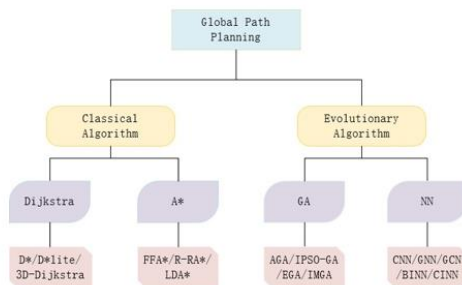
Planiranje putanje plovila neizostavni je dio svakog plovila koje sadrži uređaj za automatsko upravljanje (autopilot). Sustavi planiranja putanje plovila predstavljaju ključnu disciplinu u pomorskoj autonomiji, igrajući presudnu ulogu u sigurnom i efikasnom kretanju plovila od točke polaska do točke odredišta. Cilj planiranja putanje plovila je omogućiti upravljanje kursa pomoću autopilota bez sudjelovanja kormilara.

Razvitkom računalnih tehnologija i hardverskih, kontrolni sustav za praćenje putanje postao je jedan od najviše proučavanih zbog svoje sveobuhvatne primjene u različitim sferama automatike i robotike. Razvitkom autonomnog upravljanja različitih vozila javila se potreba za razvitkom novih rješenja i unaprjeđenjem postojećih u sustavima planiranja putanje vozila kako bi se izbjegle nesreće te omogućilo efikasnije planiranje različitih zadataka vozila. Posljednjih godina razvitak tehnologija planiranja putanje vozila većinom se svodilo na istraživanja unutar područja auto industrije zbog sve veće ekspanzije samovozećih automobila. Napredak u razvitku novih tehnologija planiranja putanje automobila potaknuo je reviziju i unaprjeđenje postojećih rješenja planiranja putanje unutar područja koja su bili pioniri autonomnog upravljanja putanjom kao što su zračna i pomorska vozila.

Planiranje putanje plovila dijeli se na globalno i lokalno. Po definiciji globalno planiranje puta je opsežna izvan mrežna metoda planiranja puta koja se temelji na pruženim informacijama o morskom okolišu (elektroničke karte) za dobivanje informacija o statičkim preprekama u području kroz koje plovilo prolazi [1]. S druge strane, lokalno planiranje putanje je mrežna metoda planiranja putanje koja se temelji na informacijama dobivenih fuzijom senzora i primjenom situacijske svijesti kako bi se izbjegle dinamičke prepreke na putanji koje globalnim planiranjem nije moguće predvidjeti.

Područje istraživanja ovog rada je primjena A\* algoritma u svrhu planiranja globalne putanje plovila. Algoritam globalnog planiranja putanje prikuplja informacije o okolišu te modelirajući okruženje uz zadane zahtjeve objavljuje preliminarni plan kretanja plovila određenom putanjom. Zahtjevi mogu biti različiti u ovisnosti o zadaci plovila, a neki od njih su ušteda goriva, izbjegavanje određenih područja, prilagodljivost vremenskim uvjetima, područja koje mora vozilo istražiti te mnogi drugi. U konačnici zadatak algoritma je generirati održiv put od početne do završne točke u poznatom radnom okruženju. Algoritam globalnog planiranja nema uvid u informacije lokalnog tipa kao što su kretanje ostalih plovila i prepreka na moru te je zato potrebna i lokalna optimizacija.

Trenutačno su glavne metode globalnog planiranja puta tradicionalni algoritmi kao što je Dijkstra algoritam i heuristički algoritam A start (\*) koji je i glavni predmet proučavanja ovog rada te novo razvijeni algoritmi kao što su genetski algoritam i algoritam neuronske mreže. Podjela algoritama globalnog planiranja putanje prikazana je na Slici 1.



**Slika 1:** Algoritmi globalnog planiranja putanje

## 2. Planiranja putanje pomoću A\* algoritma u simulacijskom okruženju

A\* algoritam je razvijen je kao unapređenje Dijkstra algoritma, s ciljem pružanja optimalnih rješenja problema pretrage putanja u grafu. Ova metoda kombinira karakteristike Dijkstra algoritma, koji jamči najkraće puteve, s heurističkom funkcijom koja ubrzava proces pretrage. Heurističke metode su pravila ili strategije koje se koriste za rješavanje problema ili donošenje odluka na brz i efikasan način, često bez potrebe za potpunim ispitivanjem svih mogućih opcija. A\* algoritam koristi procijenjenu udaljenost od trenutnog čvora do cilja kako bi poboljšao efikasnost pretrage. Osnovni koraci algoritma su inicijalizacija, postavljanje skupa otvorenih i zatvorenih čvorova, petlja pretrage i završetak pretrage.

Poput klasičnog Dijkstra algoritma i A\* algoritam je izašao početkom druge polovice prošlog stoljeća, točnije 1967. godine, te je shodno tomu uvedena su određena poboljšanja kako bi algoritam mogao biti primjenjiv i danas. Generalni problem heurističkih metoda je vrijeme koje je potrebno za obradu velike količine informacija te se zato gotovo nikad ne koriste prilikom zadataka koji zahtijevaju računanje u stvarnom vremenu (real-time). Budući da heuristička metoda ne slijedi rigorozan proces, rezultati općenito variraju. Proces je uglavnom neizvjestan, pa ima visok stupanj slučajnosti [2]. Nadalje, kao rezultat loše provedenih heurističkih metoda mogu biti putanja koja nije glatka i ne postoje kontinuitet staze.

Trenutno su glavna poboljšanja algoritma A\* povećanje broja susjednih točaka koje treba pretražiti kako bi se poboljšala glatkoća, optimiziranje heurističke funkcije za smanjenje vremena izračuna i smanjenje proračuna rastera na učinkovitost. U znanstvenom radu [3] predložen je FFA\* (Finite angle A\*) algoritam koji u odnosu na klasični A\* algoritam uvedena je sigurnosna udaljenost od putanje do prepreke tj. parametar sigurnosne udaljenosti. Dodavanje sigurnosne udaljenosti od prepreka omogućilo je sagledavanje najkraćeg puta iz druge perspektive. Naime, klasični A\* algoritam pronalazi najkraći put koji ponekad može imati veliki sigurnosni rizik zabijanja o prepreku zbog nedefinirane udaljenosti izbjegavanja. FFA\* proširio je raspon pretraživanja i povećao broj susjednih točaka u algoritmu čime je povećana glatkoća generirane rute uz povećanje sigurnosti plovidbe. S druge strane, njegovom primjenom vrijeme računanja algoritma dodatno se produljuje iako je već istaknuto da je to jedan od glavnih problem A\* algoritma. Zbog ranije navedenog u radu [4] predložen je LDA\* (Limited-Damage A\*) u kojem

se oštećenje koristi kao kriterij izvedivosti putanje uz duljinu staze kao kriteriji optimalnosti. Optimizacija se također provodi po duljini putanje kako bi se osigurala robusnost i sigurnost. Uvođenjem ranije navedenih kriterija optimizacije heurističkih funkcija nastoji se pronaći najkraći put u mrežnom okruženju radi uštede vremena. LDA\* metoda je brza za statičke prepreke i generira izvedive rute, ali izvedba nije optimalna. Optimalno planiranje rute uveden je R-RA\* algoritmom koji koristi samo dio mrežne karte za planiranje rute čime se poboljšava operativna učinkovitost algoritma i štedi vrijeme računanja, ali kao nedostatak se ističe što generirani dijelovi putanje nisu globalno optimalni. Dodatna poboljšanja koje je moguće uvesti su i zamjena heksagonalne mreže pravokutnom što smanjuje broj čvorova i čini putanju glađom.

Na tragu ideje FFA\* algoritma u kojemu je definirana sigurna udaljenost od prepreke, u ovome radu biti će ispitano djelovanje klasičnog A\* algoritma s različitim troškom čvorova s obzirom na udaljenost od prepreke u mapi. Ovakvim pristupom nastoji se postići optimizacije putanje dodavanjem više tipova čvorova u mapi pri čemu se njihov broj ne mijenja tj. vrijeme računanja algoritma ostaje isto. Detaljniji opis postupka opisan je u narednom poglavlju.

## **2.1. Opis primjene A\* algoritma na mapi troškova (eng. Cost map)**

Prvi korak implementacije A\* algoritma u simulacijskom okruženju je izrada mape. Mapa se dobiva tako da se definirani prostor unutar simulacijskog okruženja podijeli na određeni broj jednakih kvadratnih polja (tzv. grid mapa) te se poljima pridaje značenje u obliku čvorova. U osnovnom obliku čvor može poprimiti značenje prepreke ili slobodnog prolaska (binarni oblik). Kako je rad napravljen za svrhu demonstracije A\* algoritma za planiranje putanje plovila, osnovna grid mapa proširena je na mapu troškova u kojoj čvorovi slobodnog prolaska i prepreke odvojeni dodatnim čvorovima (poljima) s različitim vrijednostima troška prolaska. Definirana su dodatna tri čvora s obzirom na udaljenost od čvora prepreke. Prvi dodatni čvor mape troškova je crveni čvor koji predstavlja najmanju udaljenost od čvora prepreke (susjedni čvor), dok je žuti čvor definiran kao susjedni čvor crvenome čvoru, a zeleni čvor susjedni je čvor žutomu. Takva konfiguracija čvorova osigurava da čvor prepreke i čvor slobodnog prolaska u bilo kakvoj konfiguraciji mape uvijek budu udaljeni za 3 čvora.

Kod izračuna putanje plovila ovakva definicija čvorova i mape preduvjet je planiranja sigurne i efikasne putanje plovila. Naime, plovila ne bi smjela ploviti na određenim udaljenostima od obale ukoliko ne isplovljavaju iz luke ili uplovljavaju u luku zbog male dubine mora i velikog broja prepreka koje se mogu pojaviti kao što su ostala manja plovila, bove, hridi i ljudi. Takva područja predstavljena su crvenim čvorovima. Nadalje, žutim čvorom definirano je područje sigurne udaljenosti za kretanje plovila većim brzinama, a također se mogu pojaviti i ranije navedene prepreke. Na posljeticu, zeleni čvor predstavlja područje u kojem ne vlada toliko opasnosti i ograničenja kao u crvenom i žutom području od obale, ali se definira kao svojevrsna tampon zona između čvorova slobodnog prolaska i žutih čvorova. Osim toga, zeleni čvor je definiran kako bi se željena putanja pomakla što je više moguće ka sredini morskih prolaza tj. kanala što je jedno od pretpostavki dobrog planiranja putanje plovila.

Nakon što je definirana mapa troškova, potrebno je pomoću nje razviti A\* algoritam planiranja putanje koji će planirati putanju s obzirom na različite vrijednosti troška pojedinih čvorova. Kao i kod drugih algoritama pronalaska putanje, cilj algoritma A\* je pronaći putanju s najmanjom udaljenošću i troškom (cijenom puta). Prvo što je potrebno je definirati je trošak pomaka između susjednih čvorova te svaki čvor u zadanoj grid mapi troškova graniči sa šest susjednih čvorova. Trošak pomaka u vertikalnom i horizontalnom smjeru tako se definira kao vrijednost 1, a u dijagonalnim smjerovima kao  $\sqrt{2} \approx 1.4142$  što je zapravo euklidska udaljenost između dva čvora po Pitagorinom poučku. Nadalje, ovakvi troškovi pomaka definirani su za osnovnu mapu te je potrebno proširiti pridavanje troška pomaka čvorova s obzirom na njihov tip. Pridavanje dodatnih vrijednosti troška s obzirom na tip čvora može jednostavno izvesti tako da se ranije spomenute vrijednosti pomaka od 1 i  $\sqrt{2}$  pomnože određenom varijablom. Tako se uz trošak čvora prepreke od  $\infty$  i trošak čvora slobodnog prolaza od 1 ili  $\sqrt{2}$ , uvode troškovi crvenog, žutog i zelenog čvora koje su definirane kao umnožak troška slobodnog prolaska s varijablom čvora (zeleni, žuti i crveni troškovi). Svrha ovog rada upravo je ispitati pretragu A\* algoritma u ovisnosti o različitim vrijednostima varijabla čvorova. Primjer troškova susjedstva s obzirom na tip čvorova prikazane su na Slici 2.

$\sqrt{2}$ * zelena varijabla	1 * žuta varijabla	$\sqrt{2}$ * crvena varijabla
1 * zelena varijabla	Trenutni čvor	1 * žuta varijabla
$\sqrt{2}$ * zelena varijabla	1 * zelena varijabla	$\sqrt{2}$ * zelena varijabla

Slika 2 : Troškovi pomaka u susjedstvu trenutnog čvora s obzirom na varijablu čvora

Nakon što su definirane vrste čvorova i troškovi pomaka među njima, konačno je moguće uvesti funkcionalnost A\* algoritma pretrage za pronalaženje puta s najmanjom udaljenošću s obzirom na trošak pojedinih čvorova. Ključna ideja A\* algoritma leži u pametnom odabiru čvorova za provjeru tijekom pretrage, čime se smanjuje ukupna složenost i omogućuje efikasno pronalaženje najkraćeg puta. Heuristička funkcija koristi se kako bi se procijenila udaljenost od početnog čvora do ciljnog čvora, usmjeravajući tako pretragu prema potencijalno optimalnim putevima. Ovaj pristup čini A\* algoritam prikladnim za različite vrste problema pa tako i planiranja putanja plovila.

Algoritam A\* definira dvije vrijednosti, G-trošak i H-trošak. Trošak kretanja G-trošak predstavlja trošak kretanja od početnog čvora do određenog čvora. U mreži obično uključuje prijedenu udaljenost u vertikalnim, horizontalnim ili dijagonalnim smjerovima. Trošak se akumulira, što znači da se povećava kako algoritam istražuje različite putanje i uključuje ranije opisane vrijednosti čvora s obzirom na njihov tip. Heuristički trošak H-trošak je procjena troška određenog čvora do ciljnog čvora. Temelji se na heurističkoj funkciji koja pruža grubu aproksimaciju preostale udaljenosti. Najčešća heuristika je euklidska udaljenost ili manhattanska

udaljenost između dvije točke. A\* algoritam koristi zbroj troška kretanja i heurističkog troška za ocjenu čvorova i prioritetno njihovo istraživanje u procesu pretrage. Funkcija evaluacije je:

$$F(n) = H(n) + G(n)$$

$$G(n) = G(\text{roditelj}) + (\text{udaljenost do susjednog čvora}) * (\text{trošak s obzirom na tip čvora})$$

F(n) je ukupni trošak dolaska do cilja od početnog čvora kroz čvor n, G(n) je trošak puta od početnog čvora do čvora n koja, a H(n) je heuristička procjena troška od čvora n do cilja. Algoritam istražuje čvorove s nižim vrijednostima što osigurava pronalazak najkraćeg puta.

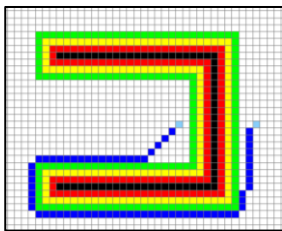
Zaključno, u ovome poglavlju opisana je izrada mape troškova i algoritam A\* za pronalazak putanje u mapi. U narednim poglavljima bit će opisana simulacija algoritma na osnovnoj mapi i mapi Sjevernog primorja te ovisnost pronalaska optimalne putanje o različitim vrijednostima varijabli troškova s obzirom na tip pojedinih čvorova.

## 2.2. Demonstracija pronalaska putanje pomoću A\* algoritma na osnovnoj mapi

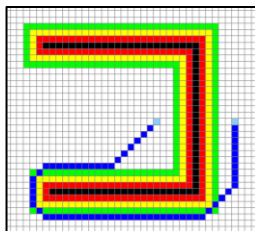
U osnovnoj mapi prepreke zatvaraju početnu točku s tri strane te je krajnja točka postavljena dijagonalno s druga horizontalno prepreke kao što je prikazana na slikama. Iz rezultata sa Slike 3 je vidljivo da kada su varijable troška velike, optimalna putanja uvijek prolazi samo kroz čvorove slobodnog prolaska što se poklapa s ranijim zaključcima jer ostali čvorovi imaju gotovo za red veći trošak. Međutim, ako se varijable troška smanje kao što je prikazano u 3. i 4. retku tablice 1, sa Slike 3 i 4 je vidljivo da će onda putanja prolaziti kroz zelene i žute čvorove jer je kroz njih najmanji trošak puta do cilja. Provedbom više eksperimenata u simulaciji dobiven je zaključak da sve dok su vrijednosti zelene i žute varijable dovoljno razmaknute, optimalna putanja će gotov uvijek prolaziti samo kroz čvorove slobodnog prolaska i zelene čvorove što je prikazano na Slici 4. Međutim, ako su vrijednosti zelene i žute varijabli bliže, putanja će prije prolaziti žute čvorove od zelenih ako je time osiguran prolazak kroz manji broj čvorova do cilja što je prikazano na Slici 5. To se događa jer trošak žutog i zelenog čvora nije dovoljno velika kao u prethodnom primjeru da algoritam prije bira prolazak kroz zeleni čvor od žutog. U prethodnim primjerima trošak crvene varijable bio je  $\infty$ , međutim u petom retku tablice pridana joj je velika konačna vrijednost jer se crveno smatra područjem od velikog rizika zabijanja o prepreku. Rezultatima je potvrđeno da će algoritam birati jedino crveni čvor ukoliko nema niti jedne druge opcije za prolazak do cilja kao što je prikazano na Slici 6 i to je željeno ponašanje algoritma.

Varijable troška	Zelena	Žuta	Crvena	Slika
Vrijednosti 1	10	15	$\infty$	3
Vrijednosti 2	1.2	1.75	$\infty$	4
Vrijednosti 4	1.2	1.5	$\infty$	5
Vrijednosti 4	1.2	1.5	100	6

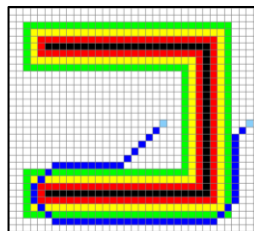
Tablica 1 : Varijable troška za testiranje algoritma u osnovnoj mapi



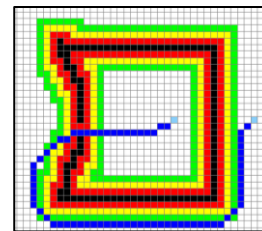
Slika 3



Slika 4



Slika 5



Slika 6

### 2.3. Demonstracija pronalaska putanje pomoću A\* algoritma na mapi Sjevernog hrvatskog primorja

Kao i u svakome eksperimentu, u osnovnoj mapi je teško predvidjeti određena ponašanja algoritma za specifičnu primjeru kao što je planiranje putanje plovila. Tako je za demonstraciju odabrana mapa Sjevernog primorja kako bi se ispitali ranije dobiveni zaključci na intuitivnijem primjeru. U ovoj mapi prepreke su zapravo obala kopna i otoka, a crveni, žuti i zeleni čvor predstavljaju udaljenost od njih.

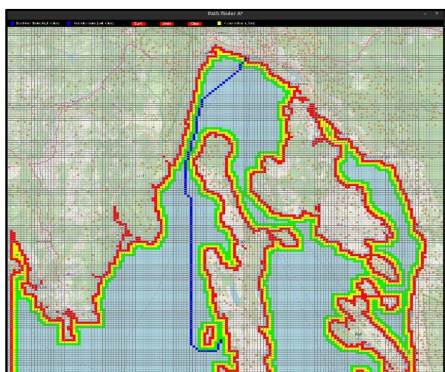
Zbog zanimljivih postavki reljefa odabrane su dvije krajnje točke za planiranje putanje iz grada Rijeke, a to su područje između Otoka Cresa i otoka Zeča te početak uvale Baška na otoku Krku. Iz rezultat s narednih slika potvrđeni su zaključci s osnovne mape. Demonstracijom algoritma na stvarnoj mapi može se uvidjeti da zapravo odabirom parametara može optimizirati putanju plovila imajući na umu željeni ishod. Naime, za slučaj putanje između Rijeke i područja između otoka Cresa i Zeča (Slike 7 i 12), kada su varijable troška žutog i zelenog čvora velike, vidljivo je da putanja nije optimalna jer se koristi veća udaljenost i broj čvorova. Kako je ranije definirano da u zelenome području ne vlada tolika opasnost po plovilo te kako zeleni čvorovi služe da se putanja pomakne više ka sredini kanala, poželjno je da putanja prolazi tim područjem do cilja ukoliko je tada manja udaljenost tj. manji broj čvorova. To je postignuto s parametrima iz trećeg retka Tablice 2 (Slike 8 i 13). Također je demonstrirano da odabirom vrijednosti zelenih i žutih varijabli koje su bliže jedna drugoj, može generirati putanje koja prolazi bliže otoku Zeču (Slike 9 i 14).

Za sva tri gore navedena primjera tj. za varijable troška prva četiri retka tablice, generiranja putanja između Rijeke i uvale Baška na otoku Krku uvijek je jednaka zbog izgleda reljefa u kojemu većina putanje prolazi kanalima između otoka (Slika 11). U tim slučajevima utjecaj varijabli troškova nije značajan jer ne postoji putanja s manjom udaljenošću kroz zelene i žute čvorove koju bi algoritam pronašao te posljedično promijenio izgled putanje. Međutim, sagledavši reljef južnog dijela otoka Krka, može se uvesti manji trošak crvenoga polja kako bi putanja prolazila između otoka Prvića i južnog dijela otoka Krka. Zbog male udaljenosti od obale do otoka Prvića, u mapi troškova to područje je označeno crvenim čvorom te putanja prolazi oko otoka Prvića do cilja. S druge strane, u stvarnosti je to područje s prihvatljivom dubinom i malom vjerojatnošću od pojave prepreka. Upravo zato pridavanjem varijabli troškova kao u petom retku tablice 2 gdje crveni čvor ima konačnu vrijednost, može se postići da putanja prolazi tim područjem čime se uvelike smanjuje udaljenost i broj čvorova putanje (Slika 12). S druge strane, uz ovakvo definirane varijable troškova, ranije opisana putanja do područja između otoka Cresa i Zeča prolaziti će blizu obale otoka Zeča (Slika 15) što može biti potencijalno opasno zbog manje dubine i nepreglednosti toga područja.

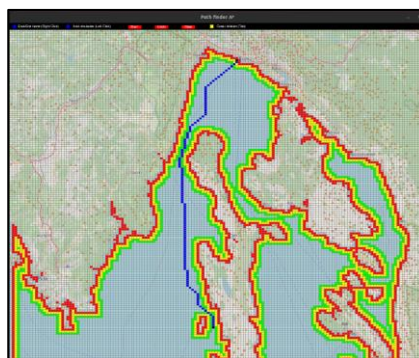


Varijable troška	Zeleno	Žuto	Crveno	Slika
Vrijednosti 1	10	15	$\infty$	7,11,12
Vrijednosti 2	1.5	2	$\infty$	8,11,13
Vrijednosti 3	1.2	1.5	$\infty$	9,11,14
Vrijednosti 4	1.5	2	10	10,12,15

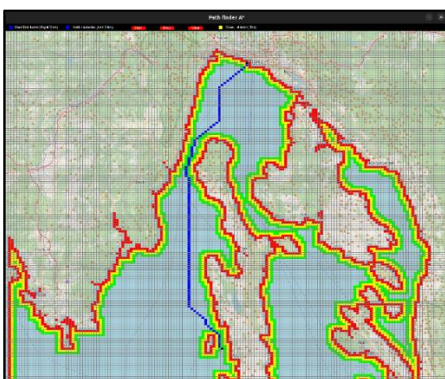
Tablica 2: Varijable troška za testiranje algoritma na mapi Sjevernog primorja



Slika 7



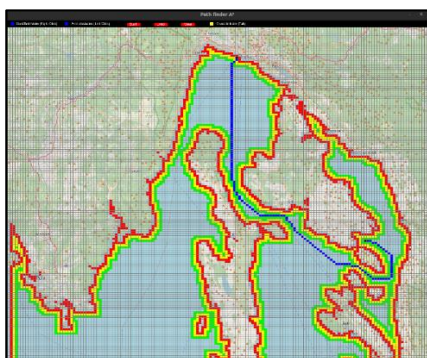
Slika 8



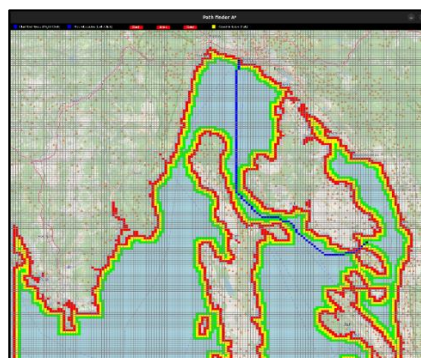
Slika 9



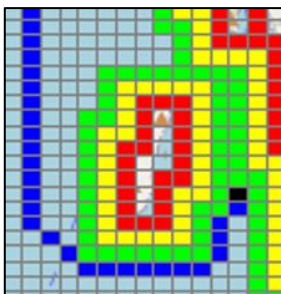
Slika 10



Slika 11



Slika 12



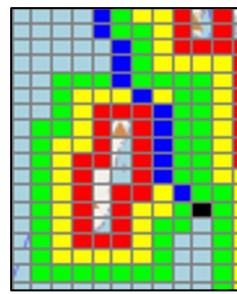
Slika 12



Slika 13



Slika 14



Slika 15

### 3. Zaključak i naredni koraci

Primjena algoritma A\* s mapom troškova za planiranje optimalne putanje plovila nije jednoznačno određeno što se upravo može vidjeti iz poglavlja 2.3. Dobiven je zaključak da odabir vrijednosti varijabli čvorova pomoću kojih se generira optimalna putanja između Rijeke i područja između Otoka Cresa i Zečca, nije istovjetan odabiru vrijednosti varijabli čvorova pomoću kojih se generira optimalna putanja između Rijeke i uvale Baška. Takav zaključak upravo govori u prilog činjenici da ne postoje univerzalno najbolji odabir parametara pri ovakvoj postavci mape troškova, već da njihovim postavljanjem možemo usmjeriti putanju željenim putem. Trenutno se generiranje mape troškova temelji isključivo na udaljenosti od obale te ne uzima u obzir dubinu mora, preglednost područja i vjerojatnost pojave prepreka što su značajne informacije za optimizaciju putanje plovila. U narednim koracima upravo bi se te informacije trebale uključiti pri dodjeli vrijednosti varijabli čvorova. Takva postavka bi za rezultat imala mapu troškova prilagođeniju stvarnome svijetu u kojoj se ne bi moralo manipulirati generiranjem putanje, već bi bilo moguće pronaći univerzalne vrijednosti varijabli čvorova pomoću kojih se u svim slučajevima pronalazi optimalna putanja.

Iz ranije navedenog može se zaključiti da je prikazani primjer dobra podloga za demonstraciju mogućnosti A\* algoritma, ali i uočavanje nedostataka klasičnog A\* algoritma na mapi troškova temeljenoj na udaljenosti od obale. Svako dodatno poboljšanje algoritma u vidu informacija o čvorovima u mapi troškova rezultira povećanjem računske složenosti procesa. Nadalje, svako smanjenje veličine polja u grid mapi također povećava računsku složenost procesa. Imajući u vidu razloge povećanja složenosti generiranja mape, u narednim koracima treba pronaći pristup generiranju mape koji bi omogućio postavljanje veličine stranice polja sumjerljive duljini plovila uz uvođenje dodatnih tipova čvorova i vrijednosti varijabli čvorova.

Jedan od prijedloga kako bi se to moglo postići je unaprjeđenje algoritma pretrage susjedstva za generiranjem mape koji bi uključivao stohastičke procese i izradom mape s različitim veličinom polja. Mapa s različitim veličinom polja temelji se na ideji da polja u blizini obale bi trebala biti manja te se povećavati s udaljenošću od obale kako bi se smanjila računaska složenost uz povećanje detaljnost mape u područjima opasnosti.

Svrha ovoga rada je bila proučavanje klasičnog A\* algoritma u mapi troškova bez naglaska na izgled i izvedivost putanje plovila. Shodno tomu, u vidu daljnjeg proučavanja tematike potrebno je ispitati navedene unaprijeđene inačice A\* algoritma s početka drugog poglavlja i njihovo ponašanje na mapi troškova kako bi se dobila šire razumijevanje problema. U konačnici, cilj cijelog procesa je razvijanje univerzalnog algoritma za izradu mape troškova i inačice A\* algoritma koja će generirati optimalnu izvedivu putanju plovila uz što je manje moguću računsku složenost.

## **4. Literatura**

- [1] B. Xing, M. Yu i Z. Liu, »A Review of Path Planning for Unmanned Surface Vehicles,« *J. Mar. Sci*, p. 1556, 6 august 2023.
- [2] Y. Y. X. S. Y. Z. F. L. Hongqiang Sang, »The hybrid path planning algorithm based on improved A\* and artificial potential field for unmanned surface vehicle formations,« *Ocean Engineering*, 2021.
- [3] C.-M. T. P. T. Joe-Ming Yang, »Path planning on satellite images for unmanned surface vehicles,« *International Journal of Naval Architecture and Ocean Engineering*, pp. 87-99, 2015.
- [4] F. P. Serhat Bayili, »Limited-Damage A\*: A path search algorithm that considers damage as a feasibility criterion,« *Knowledge-Based Systems*, pp. 501-512, 2011.
- [5] R. Antičić, »Brodsko automatsko upravljanje,« Pomorski fakultet u Splitu, Split, 2010.
- [6] L. L. W. W. Y. J. Z. Zhu, »Aplication of improved Dijkstra algorithm in intelligent ship path planning,« Navigation college, Dalian Maritime university, Dalian, China, 2021.