

**SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE**

**DOKTORSKI STUDIJ: ELEKTROTEHNIKA I
INFORMACIJSKA TEHNOLOGIJA**

KVALIFIKACIJSKI DOKTORSKI ISPIT

**PLANIRANJE PUTANJE ZA AUTONOMNE
MOBILNE ROBOTE**

Toma Sikora

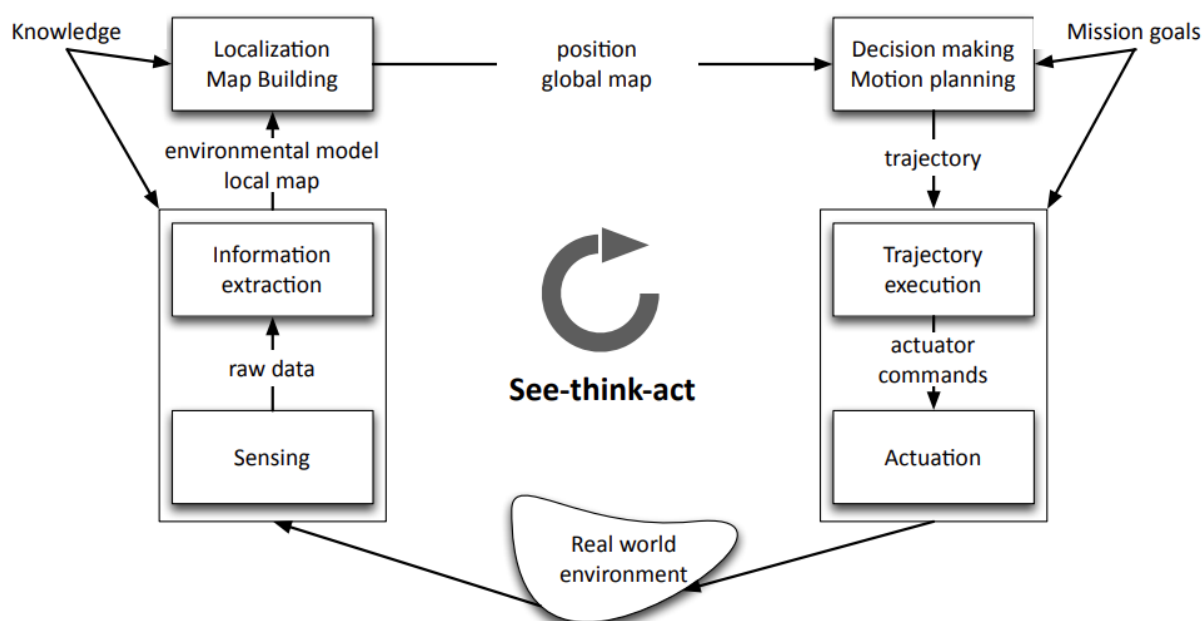
Split, srpnja 2024.

1.	Uvod.....	4
1.1.	Opis problema.....	4
1.2.	State-Of-The-Art rješenja	6
1.3.	Struktura rada	7
2.	Pristupi planiranju	8
2.1.	Potpunost, optimalnost i/ili efikasnost.....	8
2.2.	Planiranje na grafu ili u prostoru	9
2.3.	Razina planiranja	9
2.4.	Prepreke	10
3.	Algoritmi.....	11
3.1.	Dijkstra i A*	11
3.2.	Travelling Salesman Problem.....	11
3.3.	Potencijalna polja	13
3.4.	“Bug” i slični algoritmi.....	13
3.5.	Uzorkovanje prostora	14
3.6.	Model Predictive Control	17
3.7.	Cross Entropy Method.....	18
3.8.	Podržano učenje.....	20
3.8.1.	Podržano učenje temeljene na modelu	21
3.9.	Planiranje difuzijom i generativnim modelima	25
3.9.1.	Modeli difuzije	25
3.9.2.	Generative Adversarial Network mreže	26
3.10.	Ostale metode dubokog učenja	27
4.	Planiranje u statičkim okolinama.....	29
4.1.	Jedan objekt interesa.....	29
4.2.	Više objekata interesa	30
5.	Planiranje u dinamičnim okolinama	33
5.1.	Predviđanje kretanja	33
5.2.	Jedan objekt interesa.....	35
5.3.	Više objekata interesa.....	36
6.	Zaključak.....	38
	LITERATURA.....	39
	POPIS OZNAKA I KRATICA	46

1. Uvod

1.1. Opis problema

Za uspješno funkcioniranje u prostoru, ljudi i životinje koriste se brojnim pretpostavkama razvijenim evolucijom i učenjem kroz život. To može biti intuicija o interakciji sa objektima, putanji kojom se stvari kreću ili namjerama drugih elemenata u sceni. U procesu automatizacije bilo kojeg mobilnog robota, njegov kreator polazi od puno užeg vida svijeta. Sliku o stanju svijeta gradi samo iz dostupnih senzora (kao što su kamere, LiDAR-i, mikrofoni, žiroskopi i magnetometri), a utječe na njega samo preko dostupnih aktuatora (kao što su kotači, zglobovi, hvataljke i propeleri). Ograničen dostupnom računalnom snagom, robot treba pomoću senzora percipirati svijet oko sebe, planirati svoje ponašanje s obzirom na predodređene generalne smjernice i izvršavati svoj plan utjecajem na okolinu kroz svoje aktuatora. Takva se paradigma često naziva *see-think-act* ili *plan-sense-act* i njena je ilustracija dana na sljedećoj slici.



Slika: Ilustracija See-Think-Act ciklusa operacije autonomnog robotskog sustava¹

Problem planiranja za robota obuhvaća generiranje plana budućeg ponašanja robota za neki vremenski horizont s obzirom na dostupne informacije o svijetu oko sebe i krajnjim ciljem misije koju robot želi postići. Ovisno o robotskom sustavu, taj proces na ulazu ima informacije koje dolaze od senzora, a plan kao rezultat procesa može definirati precizne upute aktuatorima na nižem nivou, a može i definirati globalne smjernice koje onda upravljački modul na nižoj razini obrađuje i prati.

¹ Izvor: https://stanfordasl.github.io/PoRA-I/aa174a_aut2324/pdfs/lecture/lecture_1.pdf.

Sam plan dakle predstavlja funkciju koja mapira trenutno znanje o svijetu na buduća stanja u definiranom vremenskom horizontu. Pod pretpostavkom da možemo modelirati razvoj situacije s obzirom na naš plan, sa matematičke strane, takva se funkcija može optimizirati po brojnim kriterijima. Neki od njih su potrošena energija, broj obavljenih misija ili vrijeme obavljanja misije. Takav se pristup zove Model Predictive Control (MPC).

U stvarnom se svijetu ipak susrećemo sa brojnim preprekama koje otežavaju taj proces. Naš model možda ne obuhvaća svu kompleksnost interakcije robota s okolinom što nam otežava predviđanje, ili možda uopće nemamo model dovoljno dobar za raditi predviđanje. Isto tako kompleksna okruženja koja uključuju druge agente/objekte za sobom povlače modeliranje međusobne interakcije i ponašanja. U tim slučajevima nesigurnosti i dinamičkih okolnosti, dovodi se u pitanje samo postojanje optimalnog plana. Optimalnost plana tada ovisi o budućem raspletu događaja što je potencijalno nemoguće uzeti kao parametar optimizacije.

Ipak, od takvih problema ne treba odustati. Iako planiranje u dinamičkim okolinama predstavlja iznimno težak skup problema, nove metode upućuju na dovoljno dobra rješenja. Posebno treba spomenuti metode učenja, koje upućuju na velik prostor potencijalnog napretka. Difuzijski i generativni algoritmi, učenje modela okoline, algoritmi temeljeni na vjerojatnosti i ostali, predstavljaju izvor inspiracije za stvaranje novih moćnih pristupa planiranja.

1.2. State-Of-The-Art rješenja

Pristupi planiranja za robote podijelili su se kroz godine na nekoliko visoko specijaliziranih područja. Primarna podjela ovisi o arhitekturi mobilnog robotskog sustava. Najpopularnije arhitekture danas su zračni roboti (Unmanned Aerial Vehicle - UAV), podna vozila (Unmanned Ground Vehicle - UGV), površinski i podvodni sustavi (Unmanned Surface Vehicle – USV, Unmanned Underwater Vehicle - UUV), četveronožni roboti i humanoidi. Primjeri spomenutih arhitektura dani su na sljedećoj slici.



Slika: primjeri arhitektura autonomnih mobilnih robota²

Stupanj interakcije robota s okolinom jedan je od ključnih elemenata u odabiru algoritma planiranja. Kod zračnih robota, površinskih i podvodnih robota on je uglavnom poprilično nizak, te je planiranje i upravljanje lako odvojiti u nezavisne module. S druge strane, planiranje kod četveronožnih robota i humanoida duboko je integrirano s procesom samog upravljanja zbog visokog stupnja interakcije s okolinom. U tom području koriste se kombinacije tradicionalnih metoda planiranja predviđanja po modelu (MPC) i podržanog učenja (RL), kao naprimjer Deep Tracking Control algoritam u (Jenelten, He, Farshidian, & Hutter, 2024). Sličan je pristup korišten i kod zračnih robota kod kojih je bitna visoka agilnost. Primjer takvog algoritma je Actor Critic Model Predictive Control, razvijen u (Romero, Song, & Scaramuzza, 2023).

Planiranje obilaska više statičnih objekata na višoj razini proučava se u (Chen, Frey, Arm, & Hutter, 2023). Tim istraživača razvio je Safe Multi Goal planer za zahtjevne okoline. Razmotrili su četveronožnog robota kojem je zadatak obići više ciljeva interesa iz preddefiniranih konfiguracija. Finalni algoritam generira plan u nekoliko ugniježdenih petlji koristeći LazyPRM* i ekstenzije Traveling Salesman Problem algoritma.

² Izvor: https://stanfordasl.github.io/PoRA-I/aa174a_aut2324/pdfs/lecture/lecture_1.pdf.

Što se tiče algoritama planiranja za izbjegavanje prepreka, statičnih i dinamičnih, State-Of-The-Art rješenja su visoko optimizirani algoritmi kao MADER i rMADER, planer trajektorije za dinamička okruženja s više agenata (Tordesillas & How, 2021), (Kondo, et al., 2023).

Bitne napretke može se pronaći i u području upravljanja robotskim manipulatorima. Ti su sustavi dugo vremena imali problema sa deformabilnim i mekanim objektima u kućanstvu. Takvi su zadaci danas rješivi zahvaljujući difuzijskim algoritmima koji mogu imitirati ljude u obavljanju takvih zadataka. Primjeri takvih algoritama su Diffusion Planning (Chi, et al., 2023).

1.3. Struktura rada

Ovaj je rad podijeljen u sljedeća poglavlja. Prvo poglavlje dalo je uvod u područje planiranja za autonomne mobilne robote u dinamičkim okruženjima. To uključuje opis samog problema, tradicionalnu strukturu takvih sustava i trenutna najnaprednija rješenja. Drugo poglavlje predstavlja raspravu o kompromisima s kojim se suočavamo pri odabiru algoritma i arhitekture modula za planiranje u ovakvim situacijama. Osim toga, drugo poglavlje sadržava i pregled algoritama korištenih u tom području kroz povijest. Treće i četvrto poglavlje donose pregled velikog broja radova vezanih uz planiranje u statičkim i dinamičkim okolnostima s jednim ili više ciljeva. Na kraju, peto poglavlje donosi zaključak pregleda područja danog u ovom radu.

2. Pristupi planiranju

Planiranje za mobilne robote predstavlja interdisciplinarno područje na presjeku polja matematike, fizike, elektronike i računarstva. Osim toga, specifičan zadatak sa sobom donosi i druga razmatranja. Na primjer polje autonomne vožnje za cestovna vozila uključuje sociološku studiju interakcije sudionika u prometu. Ovo poglavlje prolazi kroz najbitnija pitanja koja se nameću pri odabiru arhitekture i algoritma planiranja za autonomne mobilne robote.

2.1. Potpunost, optimalnost i/ili efikasnost

Razvoj svakog algoritma za planiranje mora uzeti u obzir kompromis između potpunosti, optimalnosti i efikasnosti. Potpuni algoritmi oni su koji uvijek nalaze rješenje, pod uvjetom da je dostupno barem jedno rješenje za problem koji se rješava. Potpunost algoritma matematički je uvjet za korisnost tog algoritma. Kada imamo potpuni algoritam planiranja, sigurni smo da ćemo dobiti barem jedno rješenje, ako ono uopće postoji.

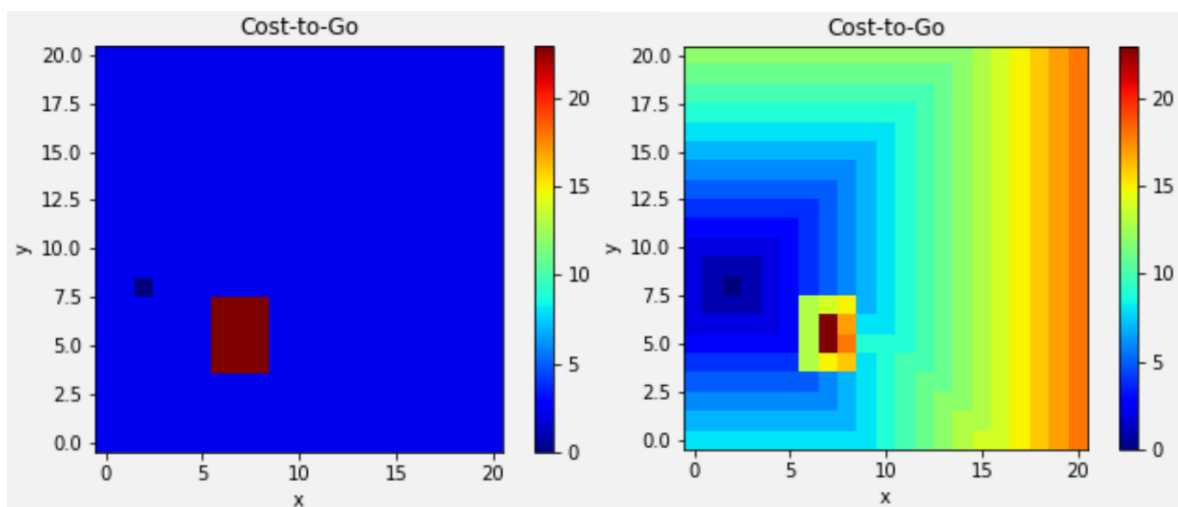
Optimalni su algoritmi oni koji uvijek pronalaze najbolje dostupno rješenje problema. Matematički gledano, za problem planiranja za robotski sustav oni promatraju prostor svih mogućih putanja koje vode od početnog do ciljnog stanja robota, uspoređuju putanje po definiranim kriterijima i odabiru najbolju. Iako teoretski savršeni, način na koji se provode optimalni algoritmi uglavnom sadrže kompleksno i iscrpno računanje te postaju korisni tek uz preinake.

S obzirom da se algoritmi planiranja izvršavaju na računalnu, bilo na robotu ili u komunikaciji s robotom, ograničeni smo dostupnim resursima tog sustava. Efikasnost algoritma nam govori o omjeru kvalitete rješenja i resursa koje trošimo. Efikasni algoritmi pronalaze dobra rješenja s malo utrošenih resursa. U polju planiranja za mobilne robote, efikasnost se uglavnom postiže informiranjem algoritma planiranja vanjskim znanjem kao što su heuristike.

Kako su mobilni roboti uglavnom ograničeni računalnom snagom zbog nosivosti bitno je postići dobar kompromis između optimalnosti i efikasnosti.

2.2. Planiranje na grafu ili u prostoru

Izravno vezano uz problem efikasnosti odluka je o prostoru u kojem se traži rješenje problema planiranja. Mobilni se roboti uglavnom kreću u tri prostorne i tri rotacijske dimenzije, pa bi prirodno bilo tražiti kontinuirane putanje u tim dimenzijama. Kako je to za duže vremenske horizonte računski iznimno neefikasno (primjer algoritma koji to ipak radi je MPC), većina pristupa planiranju pronalazi aproksimaciju tog prostora drugim domenama, na primjer grafovima (na primjer A*). Primjeri tih aproksimacija su diskretizacija kao što je vidljivo na sljedećoj slici, ručna izrada grafa za planiranje ili izgradnja grafa algoritmom uzorkovanja kao što su PRM* ili RRT*.



Slika: aproksimacija 2D prostora diskretizacijom kao povezani graf u kojem se agent može kretati do susjednih ćelija. Lijevo je prikazan prostor s preprekom u crvenom. Desno je prikazana cijena putanje do svakog od čvorova grafa³.

2.3. Razina planiranja

S većom autonomijom autonomnih mobilnih robota pojavili su se problemi u vezi horizonta planiranja. Naime, provođenje skupe optimizacije plana za dalek vremenski horizont nema smisla, ako postoji ikakva nesigurnost u parametre i razvoj stanja okruženja. Kako bi to izbjegli, proces planiranja u većini situacija podijeljen je u dvije razine: globalna i lokalna. Globalno planiranje izvodi se na aproksimaciji domene u kojoj se kreće robot, te se u njoj nižom frekvencijom provodi planiranje od trenutne pozicije robota do ciljnih točaka. Na toj izračunatoj gruboj putanji u petlji veće frekvencije sustav lokalnog planiranja optimizira putanju oko prepreka i lokalnih elemenata. Uz ovakvu se podjelu računarski napor planiranja značajno smanjuje uz zanemariv gubitak u vidu optimalnosti.

³ Izvor: <https://underactuated.mit.edu/dp.html#Sutton98> , 17.5.2024.

Još je jedan način podjele pristupa planiranju na razine, geometrijsko planiranje i planiranje temeljeno na varijablama upravljanja. Geometrijsko planiranje podrazumijeva stvaranje putanje kojoj su pojedini elementi stanja u domeni prostora u kojem se robot kreće. Na primjer, za plovilo bi to bio trodimenzionalni prostor x i y koordinata i kuta kursa plovila. Za podvodnog robota to je 6D prostor s tri prostorne x , y , z i tri kutne dimenzije. S druge strane, planiranje temeljeno na varijablama upravljanja generira plan čiji su elementi vrijednosti kontrolnih varijabli za pojedini vremenski trenutak. Dimenzije tog stanja ovise o samom robotskom sustavu, pa su tako dimenzije za UAV s četiri propelera vektor s četiri vrijednosti, za brodove s propelerom i kormilom vektor s dvije vrijednosti, a za četveronožne robote s 3 zgloba po nozi vektor s 12 vrijednosti.

Slično kao i globalno i lokalno, ta je podjela u praksi usko povezana s planiranjem na grafu ili u prostoru. Kod kompleksnih robotskih sustava, planiranje se uglavnom radi na dvije razine: globalno i lokalno. Globalno planiranje odvija se na grafu koji predstavlja geometrijsku aproksimaciju prostora u kojem se kreće robot i kao rezultat daje suboptimalne putanje koje povezuju trenutno stanje robota sa ciljanim stanjima robota. Rezultat globalnog planiranja uglavnom je putanja u prostoru u kojem se kreće robotski sustav. S druge strane, lokalno se planiranje u praksi koristi za povećanje rezolucije globalnog plana s obzirom na trenutnu okolinu samog robota. To uglavnom obuhvaća elemente kao što su pokretne prepreke ili dinamička ograničenja robotskog sustava. Rezultat procesa lokalnog planiranja uglavnom je plan akcije u prostoru kontrolnih varijabli samog robotskog sustava.

2.4. Prepreke

Kako bismo sačuvali fizički integritet mobilnog robota, moramo uzeti u obzir prepreke u okolini. Naravno, problem koji one predstavljaju također ovisi o sustavu (robotima u svemir predstavljaju manji problem od cilja postizanja dugotrajne autonomije) i o prirodi samih prepreka. one mogu biti statične (na primjer zidovi, namještaj, industrijske instalacije), a mogu biti i dinamične (na primjer ljudi, životinje, vozila, drugi roboti).

Postoji više načina na koji možemo prepreke uključiti u planiranje. Prvi je pristup već u prostoru planiranja definirati zabranjena područja koja sadrže prepreke (kod kompaktnih robota to se uglavnom radi napuhivanjem prepreke za radijus robota). Takav se pristup koristi sa velikim statičkim preprekama (Wang, et al., 2023). Drugi je pristup prepreke uključiti na nižim razinama nakon planiranja. Zadatak je onda sustava upravljanja slijediti isplaniranu putanju s minimalnim devijacijama reagiranjem na prepreke u prostoru. Ovakav je pristup prigodan kada se pojavljuju manje dinamične prepreke (Tordesillas & How, 2021), (Romero, Song, & Scaramuzza, 2023).

Situacije u kojima se elementi okoline kreću predstavljaju poseban problem za proces planiranja putanje mobilnih robota. Ukoliko se te kretnje ne uzmu u obzir, rezultat planiranja vrlo brzo može postati neizvediv, ponekad i opasan.

3. Algoritmi

Algoritmi koji se koriste u procesu planiranja za autonomne mobilne robote počeli su se razvijati i prije samih robotskih sustava. Ovo će poglavlje predstaviti najbitnije algoritme na području planiranja za mobilne robote zajedno sa domenama u kojima se oni koriste, od Dijkstra i A* do generativnih neuronskih mreža.

3.1. Dijkstra i A*

Vjerojatno najpoznatiji algoritam za pronalazak optimalnog puta između dvije točke na grafu je A* algoritam. Stvoren je 1968. za Shakey robota kao ekstenzija Dijkstrinog algoritma uz heurističku funkciju (Hart, Nilsson, & Raphael, 1968). Na težinskom grafu, A* algoritam pronalazi optimalan put između dva čvora oslanjajući se na najmanju očekivanu putanju u odabiru idućeg čvora (zbrajanjem dosadašnje duljine puta i očekivanog ostatka puta do krajnjeg čvora po definiranoj heurističkoj funkciji). Dijkstrin je algoritam (Dijkstra, 2022) preteča A* algoritma u kojem se ne koristi heuristička funkcija, pa se A* zove i informirani Dijkstrin algoritam.

Algoritam je dan danas korišten u brojnim aplikacijama zahvaljujući potpunosti i optimalnosti. Negativna strana ovog algoritma (ali i svih tog tipa) njegova je visoka prostorna kompleksnost, pa primjenjivost ovisi o veličini grafa kojim modeliramo našu okolinu. Naravno, u praksi će performanse algoritma također izravno ovisiti o kvaliteti heuristike koju koristimo.

U svijetu mobilnih robota, A* algoritam je primjenjiv ako možemo prostor u kojem se robot kreće prikazati relativno malim i statičnim težinskim grafom na kojem možemo pronalaziti plan. Zanimljive nadogradnje A* algoritma su D* i Focused D* (Stentz, 1995) gdje je dopušteno mijenjanje udaljenosti između parova točaka u grafu, te algoritmi ažuriraju optimalne putanje u skladu s novim informacijama.

3.2. Travelling Salesman Problem

Što se tiče algoritama obilaska više točaka u prostoru, povijesno je najznačajniji problem trgovačkog putnika (Travelling Salesman Problem, TSP) formuliran 1930. Na skupu točaka s definiranim udaljenostima među svim parovima točaka, on se sastoji od pronalaska optimalne sekvence posjećivanja svih točaka. S obzirom na primjenjivost on predstavlja jedan od najčešće razmatranih problema u području optimizacija i pripada familiji NP potpunih problema. Unatoč tomu, postoje razni pristupi za omogućiti rješavanje problema s velikim brojem točaka kao što su relaksacije problema ili heuristike. Planiranje putanje mobilnog robota s definiranim statičnim objektima odgovara TSP problemu. Problem planiranja putanje za obilazak dinamičkih objekata je prema tome, kompleksniji.

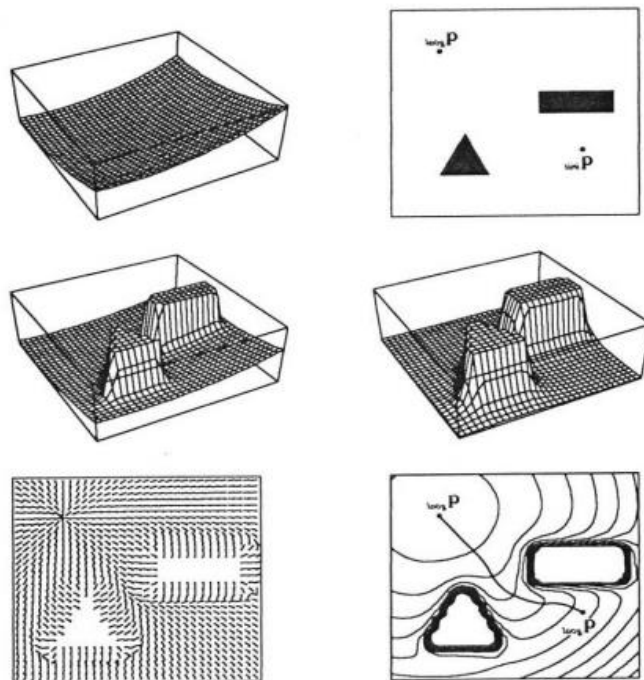
Dvije varijante TSP problema, Generalized Travelling Salesman Problem (GTSP) (Noon & Bean, A Lagrangian Based Approach for the Asymmetric Generalized Traveling Salesman Problem, 1991) i TSP with Neighbourhoods (TSPN) (Gentilini & Shimada, 2012), razmatraju ciljeve na nešto drugačiji način. GTSP kao skupove točaka, TSPN kao kontinuirana susjedstva. Pri aplikaciji u stvarnom svijetu, takvi pristupi imaju više smisla s

obzirom da se ciljevima u zadacima inspekcije uglavnom treba samo prići. Pristupi potpunom rješenju GTSP problema uključuju Branch and Cut (Pintea, Pop, & Chira, 2017), Lagranževe relaksacije (Noon & Bean, A Lagrangian Based Approach for the Asymmetric Generalized Traveling Salesman Problem, 1991 i konverzije u ekvivalentni TSP problem (Noon & Bean, An Efficient Transformation Of The Generalized Traveling Salesman Problem, 1993). Osim toga, u literaturi postoje brojni primjeri korištenja heuristika u rješavanju GTSP i TSPN problema. Primjeri su primjena mravljeg algoritma (Yang, Shi, Marchese, & Liang, 2008) i genetskih algoritama (Snyder & Daskin, 2006). Bitno je napomenuti da ovi problemi ne uključuju računalno intenzivan postupak provjere prepreka.

Multi-Goal Path Planning (MTP) je verzija TSP problema u robotici, gdje su čvorovi lokacije u prostoru u kojem se kreće robot, a duljine veza između njih udaljenosti. Rezultat takvog je algoritma putanja bez kolizija s okolinom i često uključuje dinamička ograničenja samog robota u proračun, Multi-Goal Motion Planning (MGMP). Rješenja ovog problema dana su na primjer u (Tree, 2007) s Minimum Spanning Tree (MST) algoritmom i (Plaku, Kavvaki, & Vardi, 2010) Synergistic Combination of Layers of Planning (SyCLOP) algoritmom.

3.3. Potencijalna polja

Za razliku od dosad predstavljenih iscrpnih algoritama, potencijalna polja (Howie Choset, 2024) predstavljaju intuitivan i jednostavan pristup lokalnom planiranju i izbjegavanju prepreka. Temelji se na prikazu prostora kao polje u kojem se oko prepreka stvara negativan potencijal sile odbijanja, a oko cilja/ciljeva pozitivan potencijal sile privlačenja. Na takvom prostoru robot određuje smjer kretanja gradijentom potencijala oko sebe. Primjer računanja potencijalnog polja u dvodimenzionalnom prostoru sa dvije prepreke dan je na sljedećoj slici.



Slika: primjer procesa planiranja metodom potencijalnih polja⁴. Trokut i pravokutnik predstavljaju statične prepreke.

3.4. “Bug” i slični algoritmi

Kao i u drugim područjima razvoja algoritama i na području planiranja pronalazimo rješenja inspirirana prirodom. Takozvani “Bug” algoritmi (Howie Choset, 2024) za lokalno planiranje u robotici skup su heurističkih metoda koje koristimo za navigaciju u okolinama sa preprekama. Kao i kod insekata u prirodi, osnovni princip praćenje je prepreke do pronalaska mogućnosti za nastavak ka cilju.

“Bug1” algoritam sastoji se od sljedećih koraka:

⁴ Izvor: https://www.cs.cmu.edu/~motionplanning/lecture/Chap4-Potential-Field_howie.pdf.

1. Kretanje direktno ka cilju dok ne naiđemo na prepreku ili dođemo do cilja.
2. Praćenje obrisa prepreke dok je ne obiđemo, pamteći točku u kojoj smo bili najbliže cilju.
3. Povratak na točku iz 2. i nastavak prema cilju u koraku 1.

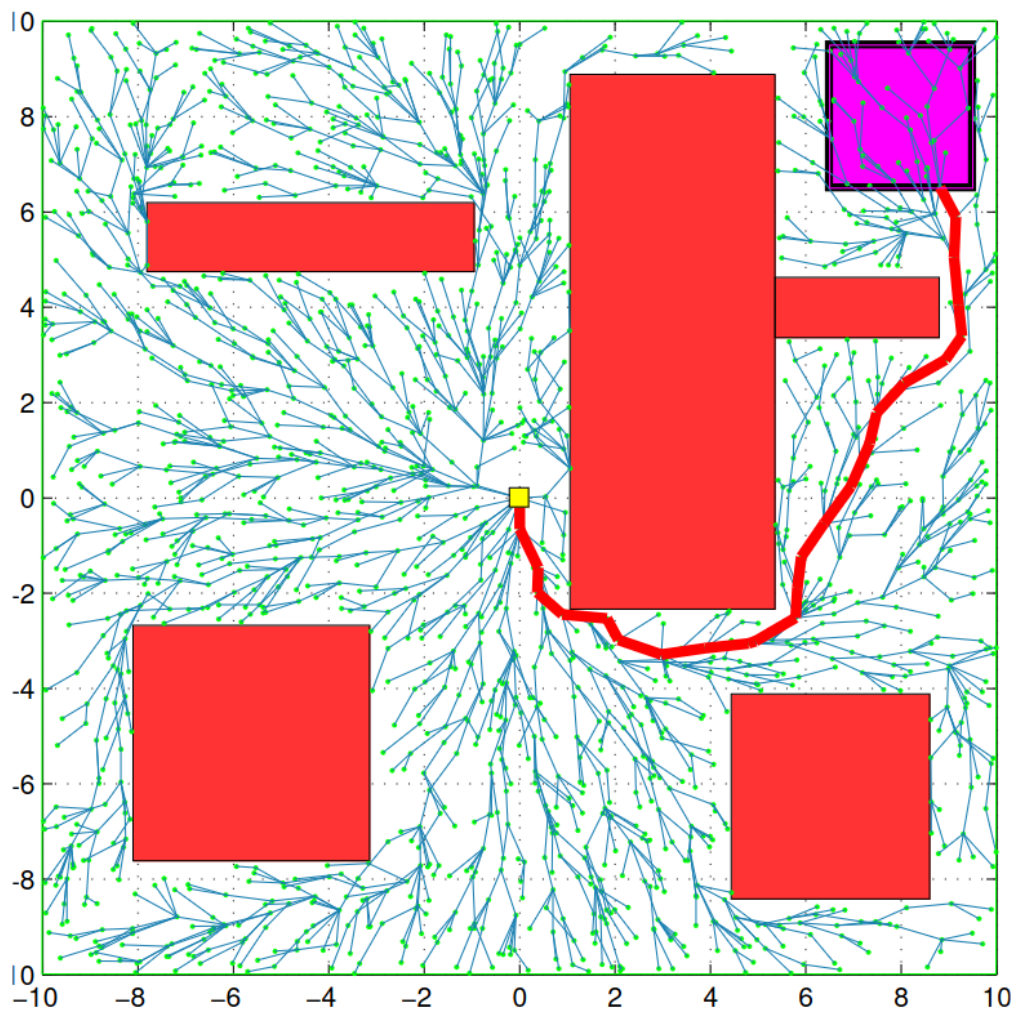
“Bug2” poboljšava ovaj pristup tako što robot prati ravnu liniju između početka i cilja i nastavlja njom svaki put kad se u obilasku prepreke vrati na nju. Ovi algoritmi jednostavni su za implementaciju i izvršavanje tako da predstavljaju dobar pristup za robotske sustave ograničene računalnim resursima. Osim toga, ovisno o okruženju, putanje ovih algoritmi izrazito su neoptimalne.

3.5. Uzorkovanje prostora

S obzirom da se mobilni roboti kreću u prostorima više dimenzija, uglavnom se mogu aproksimirati točkastom masom u tri prostorne i tri rotacijske dimenzije, iscrpno pronalaženje kontinuirane optimalne putanje nije moguće. Prirodan način za jednostavniji pronalazak plana kretanja bio bi uzorkovanje tog prostora s obzirom na prepreke te kreiranje grafa. Suočeni nedostatkom računalne moći za pronalazak optimalne putanje u takvom grafu, znanstvenici su razvili algoritme planiranja temeljene na nasumičnom uzorkovanju prostora, Probabilistic Roadmap (PRM) i Rapidly Exploring Random Tree (RRT). Glavna je ideja ovakvih algoritama smanjiti veličinu prostora pretrage postepenom izgradnjom grafa prostora odabirom nasumičnih točaka u prostoru za dodati u graf.

PRM je predstavljen 1996. (Kavraki, Svestka, Latombe, & Overmars, 1996) i algoritam funkcionira u dvije faze. Prva je faza izgradnje grafa prostora dodavanjem nasumičnih točaka u prostoru i povezivanjem sa susjednim točkama već u grafu, pazeći da veza ne prelazi preko prepreke. U drugoj fazi u graf dodajemo početnu i finalnu točku, spajamo na najbliže susjede, te pronalazimo putanju Dijkstrinim algoritmom. Vjerojatnost potpunosti algoritma je sve veća dodavanjem više točaka u graf, dok je optimalnost osigurana nadogradnjom u algoritmu PRM* (Karaman & Frazzoli, 2011).

Sličan algoritam RRT, razvijen 2001. (Lavalle & Kuffner, 2001) nasumično gradi stablo u prostoru kretanja robota. Algoritam kreće od početne pozicije robota te nasumično odabire točke prostora. Pod uvjetom da je moguć spoj sa stablom (nema prelaska kroz prepreku), točka se dodaje u stablo. Za pronalazak putanje do finalne točke testira se možemo li dodanu točku izravno spojiti s njom. Kao i kod PRM, algoritam je potpun s neograničenim brojem točaka, te ima optimalnu verziju RRT* s nadogradnjom iz (Karaman & Frazzoli, 2011). Osim toga, algoritam je lako nadogradiv mijenjanjem prostora vjerojatnosti uzorkovanja uz dodatne informacije. Primjer primjene algoritma RRT* dan je u sljedećoj slici.



*Slika: primjer optimalne putanje od početne točke (žuto) do cilja (ljubičasto) u stablu koje izgradi RRT*⁵*

U (Chen, Frey, Arm, & Hutter, 2023), autori su razvili algoritam za planiranje obilaska više statičnih objekata. Algoritam se zove Safe Multi Goal planer za zahtjevne okoline koji na izlazu definira putanju kretnje četveronožnog robota na visokoj razini (globalno planiranje). Robotu je zadatak obići više ciljeva interesa i na funkcionira kroz iteraciju tri koraka:

1. Pronalazak optimalnog redoslijeda ispitivanja ciljeva, algoritmom rješavanja Traveling Salesman Problem (TSP).
2. Odabir optimalnog prilaza inspekcije za svaki od ciljeva, algoritmom Iterative Dynamic Programming (IDP).
3. Planiranje optimalne putanje obilaska kroz definirane poze prilaza informiranim algoritmom LazyPRM* temeljenom na Open Motion Planning Library (OMPL) (Ompl: Sucas, Moll, & Kavraki, 2012).

⁵ Izvor: <https://arxiv.org/pdf/1105.1186> 16.5.2024.

Za provjeru kolizija algoritam koristi aproksimaciju robota kao kvadar te provjerom kolizije kvadra sa objektima u sceni. Ovaj pristup uspješno pronalazi i izvršava optimalne putanje u simulaciji i stvarnom svijetu.

Informed RRT* (Gammell, Srinivasa, & Barfoot, Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic, 2014) evolucija je RRT* algoritma koja nakon svakog pronalaska novog rješenja, ograničava prostor pretrage na područje višedimenzionalne elipse oko te putanje. Rezultat ovoga je dramatično ubrzanje algoritma. Fast Marching Tree* (FMT*) asimptotski je optimalan algoritam (Janson & Pavone, 2015) koji gradi na idejama PRM* i RRT*, ali konvergira brže.

Iz FMT* i Informed RRT* algoritama autori (Gammell, Barfoot, & Srinivasa, Batch Informed Trees (BIT*): Informed asymptotically optimal anytime search, 2020) razvili su Batch Informed Trees (BIT*). Algoritam pri svakoj iteraciji uzorkovanja prostora u petlji pokušava pronaći putanju počevši od ekstremno ograničene elipse između početne i ciljne točke, te postepenim opuštanjem. Evolucije ovog algoritma su RABIT* i ABIT* (Strub & Gammell, Adaptively Informed Trees (AIT*): Fast Asymptotically Optimal Path Planning through Adaptive Heuristics, 2020), (Strub & Gammell, Advanced BIT* (ABIT*): Sampling-Based Planning with Advanced Graph-Search Techniques, 2020).

PRM, RRT, FMT i ostali spomenuti algoritmi predstavljaju robusne i efikasne pristupe modeliranju prostora u kojem se kreće robot. Ipak, takvi algoritmi, temeljeni na nasumičnom uzorkovanju, predstavljaju jako uzak vid prostora kroz točke u prostoru. Kada pričamo o metodama planiranja kretnji za mobilne robote, to za rezultat ima neefikasne trajektorije za sustave upravljanja niže razine da prate.

3.6. Model Predictive Control

Model Predictive Control (MPC) napredna je metoda za upravljanje uglavnom linearnim procesima različitih razina složenosti. Koristi se od 1980-ih u kemijskim postrojenjima i naftnim pogonima, gdje su modeli sustava iznimno predvidivi, a danas su se razvile brojne ekstenzije i generalizacije kao što Nonlinear MPC, Robust MPC, Explicit MPC, iCEM i PETS (Richards & How, 2006), (Wang & Ba, 2019), (Pinneri, 2021), (Chua, 2018).

Osnovni algoritam leži na pretpostavci da imamo dovoljno dobar model našeg sustava i okoline po kojem možemo predviđati razvoj stanja. S takvim modelom onda optimiziramo buduću trajektoriju našeg sustava po definiranim kriterijima optimizacije (energija, vrijeme itd.). Jednadžba J koju se minimizira (*cost function*) izgleda ovako:

$$J = \sum_{i=1}^N w_{x_i} (r_i - x_i)^2 + \sum_{i=1}^M w_{u_i} \Delta u_i^2$$

Jednadžba: tipična cost funkcija Model Predictive Control algoritma

gdje je:

N : broj koraka koji definira budući horizont optimizacija,

M : broj kontrolnih varijabli,

w_i : težinski parametri optimizacije,

r_i, x_i : referentna i predviđena vrijednost u trenutku i ,

u_i : kontrolne varijable.

Optimizacija se radi za ograničen budući vremenski horizont, te se izvršava samo početak dobivene optimizirane trajektorije. Za rad u stvarnom vremenu, metoda onda ponavlja optimizacijski postupak pri svakom koraku, povećavajući robusnost samog pristupa. S obzirom da je takav pristup računalno neefikasan, postoje razne metode za optimizaciju.

Kako u jednadžbi optimizacije možemo uračunati elemente kao ograničenja sustava i kriterije optimizacije, ovakav je pristup iznimno fleksibilan. Također, sam model sustava ne mora biti definiran na tradicionalan način. Noviji pristupi koriste algoritme učenja za aproksimaciju samog sustava neuronskom mrežom, na kojoj se onda temelji optimizacija. Slično tako, optimizacija se ne odvija nužno u zatvorenoj formi, već se može ograničiti broj potencijalnih trajektorija iz kojeg se onda odabire najbolja.

Što se tiče robotskih sustava, inačice algoritma MPC uglavnom se koriste na nižoj razini bližoj upravljanju samog sustava (Wang, et al., 2023). Algoritam na ulazu prima globalnu trajektoriju koju treba pratiti te optimizira slijed naredbi za aktuatora kako bi najbliže pratio zadanu trajektoriju. Kod robotskih sustava veće složenosti, kao što su robotski manipulatori ili robotske ruke, ideje algoritma MPC koriste se i u fazi planiranja (Pinneri, 2021), (Chua,

2018). Definira se cilj na višoj razini, na primjer finalna pozicija manipulatora, a metoda onda pronalazi najbolju aktuaciju u određenom vremenskom horizontu za dovesti sustav u tu poziciju.

3.7. Cross Entropy Method

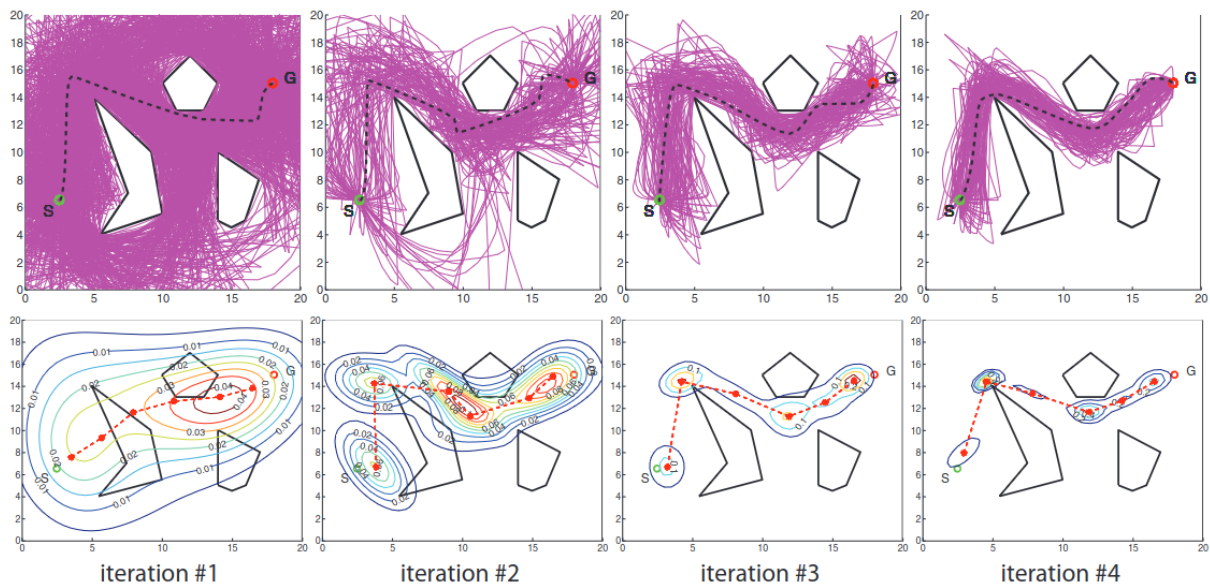
Dosad spomenute metode pronalaze optimalna ili suboptimalna rješenja za probleme koji su statični i potpune informacije. Takva su rješenja korisna i za probleme na koje se odnosi ovaj rad, planiranje u dinamičkim okolinama. Pa ipak, nesigurnu, dinamičnu i šumovitu prirodu uobičajenu za takve okoline vjerojatno bi trebalo uključiti u optimizacijski postupak. Jedan od algoritama kojim to možemo postići je Cross Entropy Method (CEM). On predstavlja jednostavan, efikasan i generalni Monte Carlo pristup utemeljen na teoriji vjerojatnosti i statistike. Njegova efikasnost proizlazi velikim dijelom iz činjenice da za optimizaciju ne koristi računanje gradijenta, već *cross-entropy* metriku.

Algoritam se odvija izmjenjivanjem dva koraka:

1. Uzorkovanje distribucije vjerojatnosti.
2. Minimizacija *cross-entropy* vrijednosti između te distribucije i ciljne distribucije kako bi idući put dobili bolje uzorke.

Metodu je razvio R. Rubinstein 1999. za procjenu vjerojatnosti za rijetke događaje (Rubinstein & Kroese, 1999), a pokazala se korisna za rješavanje brojnih kombinatorijskih optimizacijskih problema. Analiza konvergencije CEM algoritma istražena je u (Margolin, 2005).

Na području planiranja i upravljanja u robotici, algoritam CEM je korišten pri rješavanju brojnih podproblema. U (Kobilarov, 2012) autori koriste ga za planiranje kretanja za nelinearne robotske sustave u ograničenim okolinama. Ideja je koristiti CEM za aproksimaciju prostora vjerojatnosti koji se koristi u uzorkovanju točaka za RRT algoritam. Dok je originalno taj prostor uniforman i samim time uzorkovanje njega neefikasno, CEM je korišten za pronalazak informacija o područjima koja su bogatija točkama koje s visokom vjerojatnosti pripadaju optimalnoj putanji. Prostor vjerojatnosti definiran na taj način se onda koristi za uzorkovanje u algoritmu RRT. Pristup je testiran 2D i 3D problemima, te značajno povećava efikasnost RRT* algoritma. Primijenjen na dvodimenzionalnom prostoru sa statičnim preprekama, rezultat CEM metode prikazan je na sljedećoj slici.



Slika: primjena CEM algoritma pri RRT* planiranju za problem duplog integratora. Gore su prikazane pretražene putanje za četiri iteracije CEM algoritma (iscrtkana putanja je optimalna). Dolje je prikazana distribucija vjerojatnosti područja visoke vrijednosti.

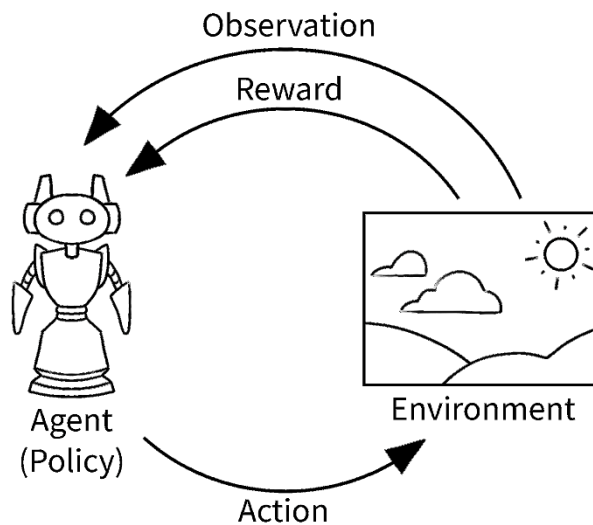
S druge strane, CEM je korišten i izravno u optimizaciji trajektorija i planiranju sa naučenim modelima. Na primjer, u (Chua, 2018), (Danijar Hafner, 2019), (Wang & Ba, 2019), (Pinneri, 2021) je korišten za generiranje familije trajektorija koja se predaje MPC algoritmu na odabir. U (Chua, 2018) autori koriste neuronske mreže za aproksimirati dinamiku interakcije okoline i agenta. Uz taj model onda uzorkuju buduće trajektorije pomoću CEM metode i odabiru najboljeg kandidata simulacijom razvoja situacije MPC pristupom.

U (Danijar Hafner, 2019) autori predstavljaju Deep Planning Network (PlaNet), algoritam učenja dinamike interakcije okoline i agenta, te vrši planiranje u latentnom prostoru u stvarnom vremenu. Pri tom naučen model predviđa vrijednost funkcije nagrade više koraka unaprijed. Koristeći samo informacije u obliku piksela, algoritam uspješno rješava probleme kontakta manipulatora, rijetke nagrade i parcijalne preglednosti.

U (Pinneri, 2021), autori uz nekoliko modifikacija, autori koriste CEM za upravljanje robotskih sustava visoke složenosti, kao što su četveronožni roboti i robotske ruke. Metoda ovdje predstavlja evolucijsku strategiju koja pronalazi kvalitetne kandidate u prostoru mogućih budućih trajektorija određenog vremenskog horizonta. Kao i prije ti su kandidati onda testirani MPC pristupom te je odabran najbolji. Glavni doprinosi algoritmu u članku su u korištenju obojenog šuma umjesto Gaussovog šuma pri uzorkovanju i zadržavanje nekih uzoraka putanje u memoriji za iduću iteraciju optimizacije. S obzirom da originalna CEM metoda odbacuje generirane trajektorije pri svakom koraku, ovo predstavlja povećanje efikasnosti algoritma.

3.8. Podržano učenje

Podržano učenje predstavlja drugačiji pristup problemima upravljanja robotom u okolini učenjem kroz iskustvo. Originalno se metoda temeljila na Bellmanovoj jednadžbi i optimizaciji putem dinamičkog programiranja (Kaelbling, Littman, & Moore, 1996), a u međuvremenu je u robotici narasla u interdisciplinarno područje na spoju optimalnog upravljanja i strojnog učenja. Tradicionalna ilustracija interakcije okoline i agenta u podržanom učenju dana je na sljedećoj slici.



Slika: apstrakcija petlje agenta i okoline u podržanom učenju⁶.

Probleme podržanog učenja karakteriziraju interakcija agenta i okoline i signal nagrade. Interakcija agenta i okoline prikazuje se kao Markovljev lanac (Markov Decision Process, MDP), što je stohastički, vremenski diskretan proces. Sam agent prikazan je funkcijom koja mapira trenutno stanje okoline na signale akcije. Generalna je ideja algoritama pronaći optimalno ponašanje agenta u okolini maksimizirajući signal nagrade.

⁶ Izvor: https://www.gymlibrary.dev/content/basic_usage/ 16.5.2024.

Primjer algoritma kojim se može rješavati takav problem je Q learning algoritam računanjem varijacije Bellmanove funkcije:

$$Q_{new}(s_t, a_t) = (1 - \alpha)Q_{old}(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in \Gamma(s_t)} Q(s_{t+1}, a))$$

Jednadžba: Bellmanova funkcija ažuriranja vrijednosti za Q-learning algoritam.

Gdje je:

Q_{new} : nova vrijednost za par stanje-akcija,

Q_{old} : stara vrijednost za par stanje-akcija,

S_t, a_t : stanje i akcija poduzeta u vremenu t ,

R_t : nagrada dobivena u trenutku t ,

α : faktor učenja,

γ : faktor opadanja utjecaja buduće nagrade.

Sa strane strojnog učenja, takav postupak se može prikazati kao nenadzirano učenje iz razloga što se ne definiraju izričiti parovi ulaz-izlaz nad kojima bi se izvršavao *gradient descent*, već je fokus na pronalasku ravnoteže među fazama algoritma optimiziranja ponašanja (*exploit*) i istraživanja (*explore*). S druge strane, iz perspektive teorije optimalnog upravljanja proces učenja možemo shvatiti kao optimizacijski postupak pronalaska buduće trajektorije po funkciji nagrade (kriterij optimizacije). Tu se mogu povući paralele s algoritmom MPC.

Kroz godine razvijeni su brojni algoritmi podržanog učenja. Neki od najpoznatijih tradicionalnih algoritama su Monte Carlo metode, Temporal Difference i Q learning, State–Action–Reward–State–Action (SARSA), dok su u zadnje vrijeme uključene neuronske mreže u Deep Q network, Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO) i State Actor Critic (SAC).

Polje podržanog učenja može se podijeliti na metode koje koriste poznati model interakcije agenta i okoline (*model based*) i one koje ne koriste taj model već ga uče same (*model free*).

3.8.1. Podržano učenje temeljene na modelu

Ove metode pretpostavljaju poznate informacije o modelu interakcije agenta i okoline. To uključuje znanje o trenutnom stanju okoline i načinu na koji agent kroz aktuator može utjecati na okolinu. Takav se model tradicionalno može otkriti postupkom identifikacije, a novijim ga je metodama moguće i aproksimirati učenjem (Wang & Ba, 2019).

Takvi su pristupi uglavnom računalno intenzivniji od *model free* metoda, ali ovisno o primjeni pokazuju dobre rezultate (Wang & Ba, 2019). Uz model, pitanje odabira trajektorije rješava

se na razne načine, na primjer CEM algoritmom kao što je predstavljeno u (Chua, 2018), (Danijar Hafner, 2019), (Wang & Ba, 2019), (Pinneri, 2021).

U (Danijar Hafner, 2019), autori kombiniraju ideje *model based* podržanog učenja i MPC ili *online* planiranja za probleme lokomocije četveronožnih i humanoidnih robota. Algoritam koji su razvili, *model based policy planning* (POPLIN) koristi neuronske mreže kao optimizacijski algoritam za planiranje situacije pri svakom koraku. Autori testiraju dva pristupa: planiranje u prostoru situacije i planiranje u prostoru parametara same mreže. Dodavanjem šuma u parametre mreže, dobiju se drugačije trajektorije koje se dalje uspoređuju, na primjer MPC algoritmom. Ovaj je algoritam efikasniji i jednostavniji od tradicionalnih pristupa (Wang & Ba, 2019) jer preskače korake CEM optimizacije. Također, autori koriste neuronsku mrežu za distiliranje znanje MPC algoritma.

U (Williams, Aldrich, & Theodorou, 2015), autori su predstavili Model Predictive Path Integral (MPPI) algoritam optimizacije trajektorije koji se temelji na stohastičkom uzorkovanju putanja za robotski sustav. Kako algoritam ne koristi derivacije dinamike sustava ni funkcija nagrade, prirodno uzima u obzir nelinearna svojstva određenih sustava. Zbog efikasnosti koristi se grafička kartica za uzorkovanje tisuća putanja u stvarnom vremenu.

U (Nagabandi, Kahn, Fearing, & Levine, 2018), autori su predstavili još jedan *model based* pristup upravljanju robotskim sustavima. Koristeći neuronske mreže srednje veličine, aproksimiraju dinamiku interakcije okoline i agenta, na temelju čega koriste MPC za postizanje efikasnog upravljanja. Osim toga, pokazali su da je takav model moguće koristiti za inicijalizirati *model free* treniranje na specifičnom zadatku.

3.8.2. Podržano učenje bez modela

Cilj je metoda bez modela interakcije, s druge strane, izravno pronaći funkciju preslikavanja iz trenutnog stanja okoline na ponašanje agenta. Iako to predstavlja kompleksniji problem, ovakav se pristup pri korištenju pokazao efikasniji i često daje bolje rezultate.

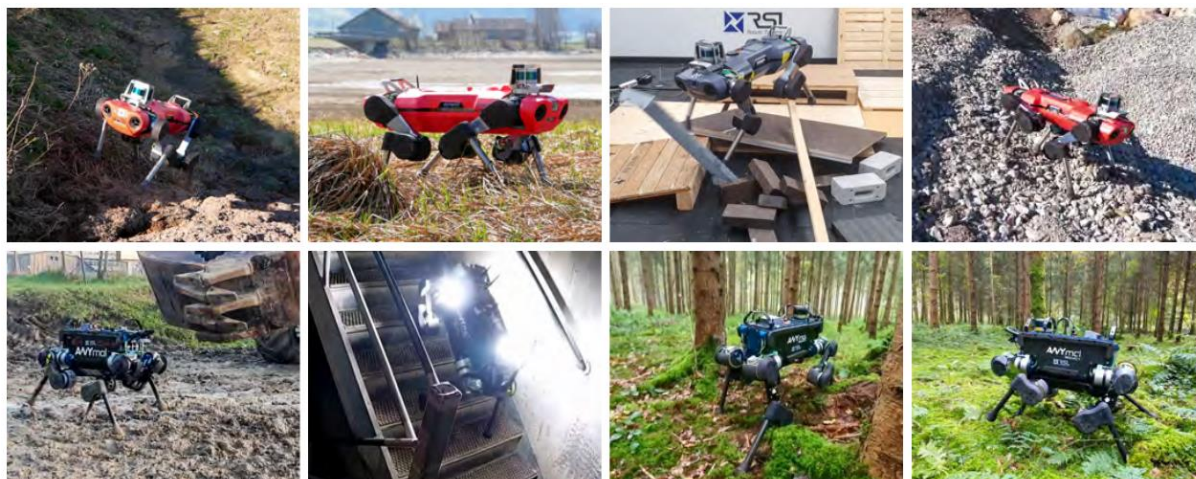
Velika je mana ovog pristupa računalno intenzivan proces treniranja neuronske mreže u dovoljno iteracija. Kod jednostavnijih problema i algoritama, učenje se može odvijati u stvarnom vremenu, dok se agent kreće u prostoru. Kod kompleksnijih algoritama i problema potrebno je unaprijed imati aproksimirani model interakcije na kojem se može trenirati algoritam. Taj proces treniranja uključuje iterativno generiranje seta za treniranje simuliranjem epizoda agenta u okolini, te treniranje upravljačkog algoritma u skladu sa setom.

Drugi je nedostatak ovog pristupa inherentna nestabilnost samog procesa učenja za kompleksne primjene. Pošto je novi set za treniranje generiran pri svakoj iteraciji procesa treniranja, prostor u kojem želimo optimizirati upravljački algoritam se stalno mijenja. Te su probleme djelomično riješili tek nedavno algoritmima kao što su TRPO i PPO (Schulman, Levine, Abbeel, Jordan, & Moritz, 2015), (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), obuzdavanjem promjene ponašanja neuronske mreže u svakoj od iteracija.

Unatoč problemima, podržano se učenje pokazalo kao jako efektivan alat u velikom broju problema. Pokazalo je nadljudske performanse u igranju igara kao što su šah, dame i Go

(Silver, et al., 2017), te efikasno upravljanje kompleksnih robotskih sustava (Romero, Song, & Scaramuzza, 2023), (Lee, Hwangbo, Wellhausen, Koltun, & Hutter, 2020).

Najpoznatiji primjer primjene algoritama podržanog učenja je (Lee, Hwangbo, Wellhausen, Koltun, & Hutter, 2020) gdje su autori razvili robustan upravljački algoritam za četveronožni robot Anymal⁷ temeljen na Proximal Policy Optimization (PPO) algoritmu. Upravljačka shema se temelji na neuronskoj mreži koja daje ciljne pozicije za stopala robota kontroleru niže razine koji onda prati te naredbe. Samojoj neuronskoj mreži na ulaz se dovode samo podaci proprioceptivnih senzora (koji mjere stanje samog robota, ne okoline). Algoritam je treniran u simulatoru kompliciranim procesom inspiriranim evolucijskim algoritmima (Wang, Lehman, Clune, & Stanley, 2019) i na kraju testiran na stvarnom sustavu bez prilagodbe. Performanse pristupa su pokazale da je pristup robustan prethodno nepremostivim problemima kao što su nestabilna uporišta i mekano tlo. Primjeri okolina u kojima je testiran istrenirani algoritam vidljiv je u sljedećoj slici.



Slika: primjeri prethodno nepremostivih okolina za četveronožne robote (Lee, Hwangbo, Wellhausen, Koltun, & Hutter, 2020).

Isti tim istraživača nastavio je razvoj na algoritmu i napravio veliki korak algoritmom Deep Tracking Control (DTC) (Jenelten, He, Farshidian, & Hutter, 2024). U njemu kombiniraju tradicionalne pristupe optimalne kontrole, konkretno metode optimizacije trajektorije (Trajectory Optimization, TO), i prethodno spomenutog podržanog učenja. DTC predstavlja najbolje od oba pristupa, nasljeđujući robusnost prethodno opisanog algoritma, ali i prednosti TO pristupa (okoline rijetkih signala nagrade kao što su rupe u tlu). U ovom pristupu TO algoritam generira buduće putanje nižom frekvencijom, dok naučena neuronska mreža prati tu putanju visokom frekvencijom.

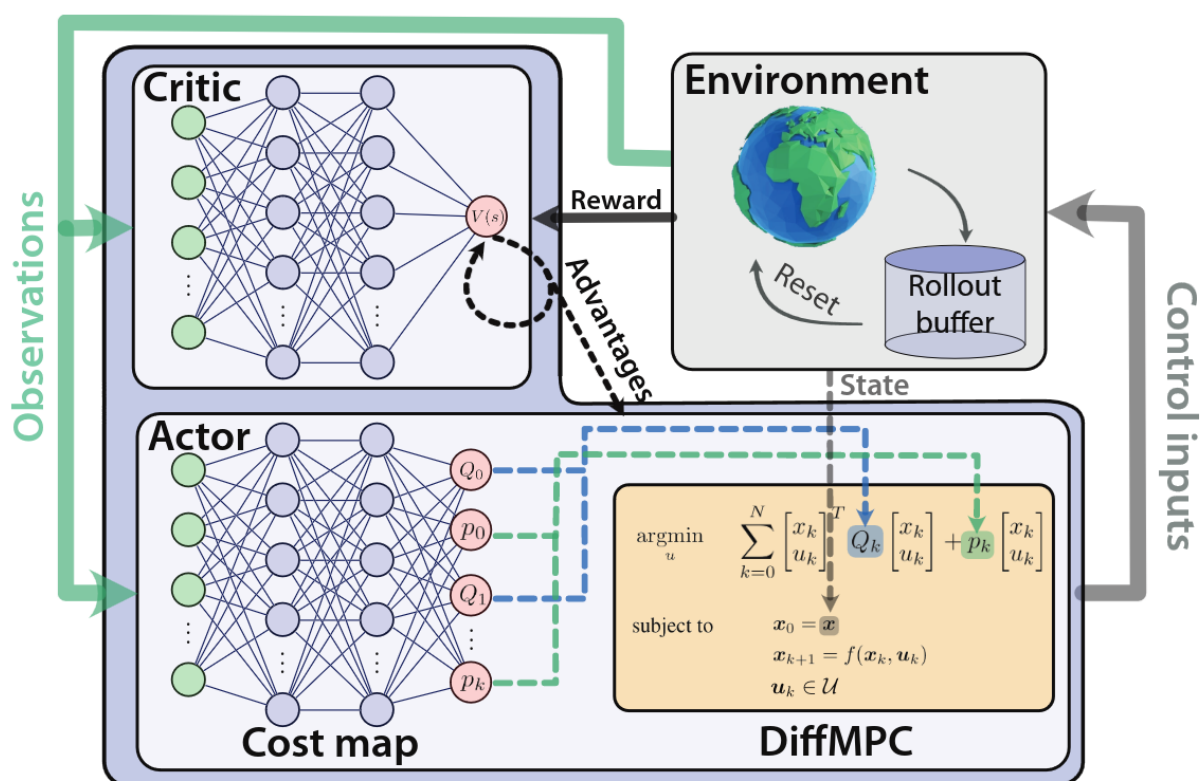
Sličan napredak postigao je i tim istraživača u (Song, Romero, Müller, Koltun, & Scaramuzza, 2023), (Romero, Song, & Scaramuzza, 2023) na problemu upravljanja zračnih

⁷ <https://www.anybotics.com/robotics/anymal/>

robota u kompetitivnom okruženju⁸. Razvijaju visoko agilne i dinamične metode upravljanja pomoću metoda podržanog učenja i optimalne kontrole.

U nizu članaka (Song, Romero, Müller, Koltun, & Scaramuzza, 2023), (Romero, Song, & Scaramuzza, 2023), (Xing, Bauersfeld, Song, Xing, & Scaramuzza, 2023), autori su koristili podržano učenje za postići razine upravljanja ljudskih eksperata u utrkivanju zračnim dronovima. Ograničeni ekstremno malom nosivosti, na raspolaganju imaju kameru, malo računalo i komunikaciju s propelerima. Slično već spomenutim pristupima, koristili su simulator u kojem su detaljno opisali dinamiku vozila. Trik kojim su uspješno povezali performanse u stvarnom svijetu i u simulatoru bio je ubacivanje empirijski izmjerene šuma u simulator. Algoritam je testiran protiv svjetskih prvaka utrkivanja zračnih dronova među kojima je postigao najbolje rezultate.

Kao i kod četveronožnih robota, zadnji pristup predstavljen u (Romero, Song, & Scaramuzza, 2023) koristi kombinaciju ideja tradicionalnog upravljanja, konkretno algoritma MPC, i metode podržanog učenja za utrkivanje zračnim dronovima. Algoritam se zove Actor Critic Model Predictive Control i zadržava poželjne elemente MPC algoritma (mogućnost optimizacije u bliskoj budućnosti) i podržanog učenja (treniranje neuronske mreže upravljanja u *end-to-end* procesu). Pokazali su kako ovakav pristup može izvršavati kompleksne manevre i robusan je situacijama izvan distribucije na kojoj je treniran. Arhitektura algoritma prikazana je u sljedećoj slici.



Slika: arhitektura sustava Actor Critic Model Predictive Control (Romero, Song, & Scaramuzza, 2023).

⁸ <https://www.drl.io/>

Za kraj, zanimljiva *meta* tematika u modernoj primjeni podržanog učenja odabir je šuma na izlazu *policy* mreže koja upravlja robotskim sustavom. Njom se bavi tim istraživača u (Eberhard, Hollenstein, Pinneri, & Martius, 2022) gdje istražuju spektar šumova od potpuno bijelog šuma do crvenog ili tzv. Brownovog šuma. Naime, boja šuma označava stupanj vremenske korelacije tog signala. Bijeli je šum vremenski ne koreliran, dok je crveni koreliran. U primjenama na robotske sustave pokazalo se korisno ubaciti šum u upravljački signal na izlazu iz neuronske mreže. Tradicionalan odabir je bio bijeli šum, ali autori pokazuju da najbolji odabir ovisi o robotskom sustavu i samom zadatku, te da je u prosjeku najbolji izbor ružičasti šum.

3.9. Planiranje difuzijom i generativnim modelima

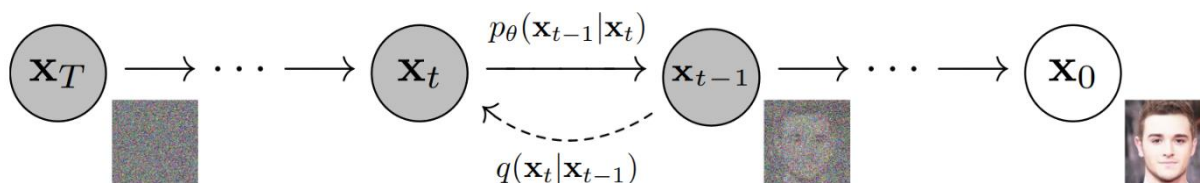
Najnoviji napredci u području planiranja za robotske sustave dolaze korištenjem difuzijskih i generativnih modela. Kod zadataka kao što je autonomna vožnja ili manipulacija mekanim i deformabilnim objektima još uvijek nisu dostupni modeli koji bi opisivali interakciju elemenata u okolini. Za autonomnu vožnju model vozila pri cestovnim brzinama prilično je dobro shvaćen i opisan. Područje u kojem nastaju problemi interakcija je među agentima u prometu. Isto tako kod manipulacije deformabilnim objektima, na primjer u kućanstvu, modeliranje interakcije manipulatora i objekta predstavlja problem. Kod takvih se zadataka učenje imitacijom iz ljudskih demonstracija difuzijskim i generativnim modelima pokazalo efikasnim.

3.9.1. Modeli difuzije

Modeli difuzije vrše iterativan proces uklanjanja šuma sa podataka. Najpoznatiji primjer toga su generatori slika kao što je Stable Diffusion (Rombach, Blattmann, Lorenz, Esser, & Ommer, 2022), koji uspješno generiraju nepostojeće slike visoke rezolucije iz signala čistog šuma. Proces je prikazan na slici dolje.

Proces treniranja kod takvih modela izgleda ovako:

1. Generiranje seta podataka za treniranje iterativnim dodavanjem šuma na set slika dok ne postanu neprepoznatljive.
2. Treniranje modela kojem je zadatak uklanjati jedan korak dodanog šuma na ulaznoj slici, tj. vršiti prethodni korak unazad.



Slika: prikaz procesa difuzije za slike. Dodavanja šuma s desna na lijevo, uklanjanje šuma s lijeva na desno⁹.

Mreža trenirana na taj način trebala bi moći iz slike potpunog šuma generirati novi primjer iz distribucije slika korištenih za treniranje. Ovaj je proces moguće trenirati i uz uvjetovanje informacijama kao što su tekstualni podaci za usmjeravanje procesa difuzije (Rombach,

⁹ Izvor: <https://arxiv.org/pdf/2006.11239> 16.5.2024.

Blattmann, Lorenz, Esser, & Ommer, 2022), (Bansal, et al., 2024), (Ramesh, Dhariwal, Nichol, Chu, & Chen, 2022).

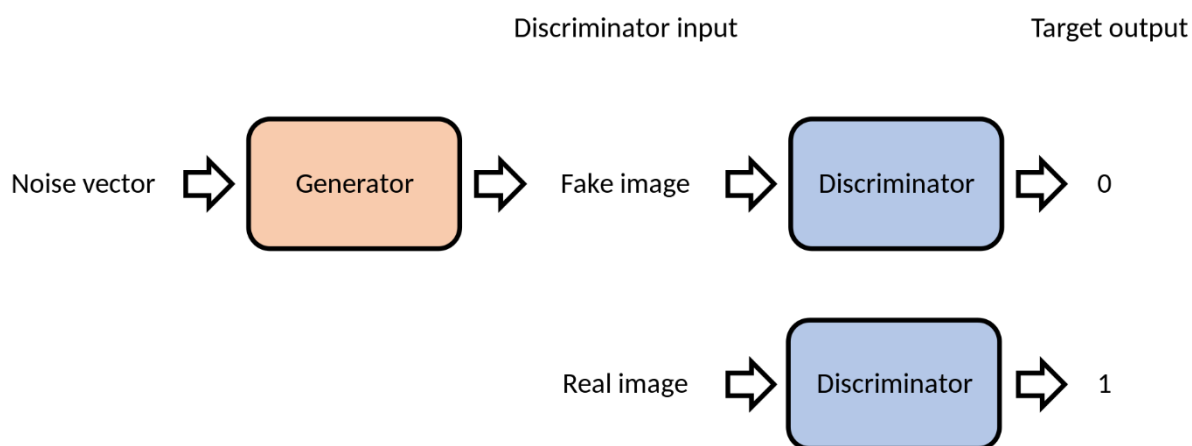
Primjena u robotici je sve više, na primjer u (Chi, et al., 2023) autori uče zadatke kao slaganje odjeće ili listanje stranica s manipulatorom kroz proces difuzije.

3.9.2. Generative Adversarial Network mreže

Poseban su tip generativnih modela Generative Adversarial Network (GAN) modeli. GAN modeli na malo drugačiji način rješavaju sličan zadatak: doći od šuma do primjera iz naučene distribucije. Umjesto da postepeno uklanjaju šum sa slike, GAN modeli istodobno treniraju dvije mreže: generator i diskriminator.

Generator mreža duboka je neuronska mreža naučena da iz vektora šuma generira primjer iz željene distribucije. Arhitektura generatora može biti na primjer dio autoenkodera, gdje model nauči sažeti ulaz u latentni prostor niže dimenzije. Enkoder dio mreže visokodimenzionalni ulaz (sliku) takozvane *sparse* reprezentacije mapira u niskodimenzionalni latentni prostor. Dekoder s druge strane tu latentnu reprezentaciju koristi za rekreiranje originalnog ulaza. Tokom treniranja oba se dijela modela uče, enkoder i dekode, dok se za generiranje primjera iz naučene distribucije koristi dekode sa nasumičnim primjerima iz latentnog prostora.

S druge strane diskriminator mreža uči ocijeniti je li podatak na ulazu umjetno generiran ili iz stvarnog seta za treniranje. Ugrubo, na početku procesa treniranja mreža diskriminator uspješno diskriminira prave od umjetnih primjera, dok generator postepeno uči stvarati kvalitetnije primjere. Treniranje se završava kada diskriminator više ne može razaznati stvarne od generiranih uzoraka. Skica procesa je dana u sljedećoj slici:



Slika: skica funkcije GAN modela¹⁰.

¹⁰

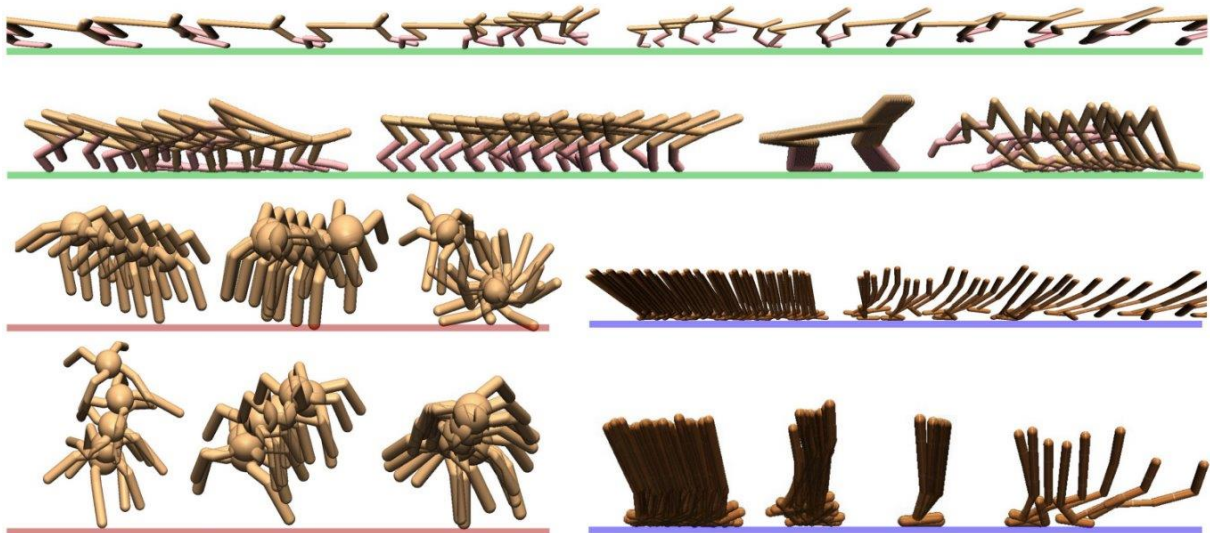
https://en.wikipedia.org/wiki/Generative_adversarial_network#/media/File:Generative_Adversarial_Network_illustration.svg , 16.5.2024.

3.10. Ostale metode dubokog učenja

Osim GAN mreža i difuzijskih modela, na području upravljanja i planiranja za mobilne robote pokazalo se korisno još nekoliko metoda dubokog učenja. Iako se to područje krenulo razvijati relativno davno, ove su metode tek nedavno postale obećavajuće. Pristup mnogih je metoda u ovoj grupi, kombinirati evolucijske i genetske algoritme s neuronskim mrežama kako bi se stvorili generalniji i robusniji modeli (Stanley & Miikkulainen, 2002), (Wang, Lehman, Clune, & Stanley, 2019), (Meier & Mujika, 2022), (Eysenbach, Gupta, Ibarz, & Levine, 2018).

Prva značajna kombinacija predstavljena je u (Stanley & Miikkulainen, 2002), gdje autori rade na postepenom povećanju kompleksnosti neuronske mreže dok to donosi pozitivne rezultate i poboljšanje metrika u algoritmu Neuroevolution of augmenting topologies (NEAT). Nastavak u istom duhu predstavljaju (Meier & Mujika, 2022), gdje se treniranje neuronskih mreža metodama podržanog učenja nadograđuje evolucijskim modelom s ciljem takozvanog *open-ended* učenja. Tu algoritmi kao Paired Open Ended Trailblazer (POET) koriste više neuronskih mreža koje se kroz treniranje križaju i mutiraju susreći se s sve težim problemima. Taj je pristup korišten i u treniranju kontrolera već spomenutog četveronožnog robota u (Lee, Hwangbo, Wellhausen, Koltun, & Hutter, 2020).

U (Eysenbach, Gupta, Ibarz, & Levine, 2018), predstavljen je još jedan zanimljiv pristup koji koristi metode nenadziranog dubokog učenja za profiliranje vještina u lokomociji robotskih sustava. Naime, kako je iznimno teško definirati dovoljno informiranu funkciju nagrade u podržanom učenju za kompleksne sustave, na primjer humanoidne robote, autori su za mjeru uspjeha pokušali u procesu učenja koristiti samo entropiju naučene *policy* funkcije. Na taj su način potaknuli istraživanje cijele domene mogućih pokreta i postigli rađanje kompleksnih sekvenci kretnji. Te kretnje predstavljaju vještine koje se onda koriste u rješavanju drugih problema, na primjer hodanja, trčanja ili dizanja na noge. Razvijeni algoritam zove se Diversity is all you need (DIAYN) i primjer vještina naučenih nenadziranim putem vidljiv je u sljedećoj slici.



Slika: primjer korištenja nenadziranih metoda dubokog podržanog učenja za lokomociju biološki inspiriranih robotskih arhitektura. Algoritam DIAYN iz (Eysenbach, Gupta, Ibarz, & Levine, 2018).

4. Planiranje u statičkim okolinama

Nakon što smo upoznali široki izbor algoritama korištenih na području planiranja za mobilne robote, ovo će poglavlje predstaviti specifične projekte koji koriste upravo njih za rješavanje problema planiranja u statičkim okolinama. Takve su postavke dobro pokriven literaturem i još uvijek ostavljaju prostora potrazi za optimalnim rješenjem. Pregled najpopularnijih algoritama korištenih u području planiranja za mobilne robote dan je u (Karur, Sharma, Dharmatti, & Siegel, 2021), (Tang & Ma, 2021). Velika većina algoritama već je predstavljena u drugom poglavlju, a oni uključuju Dijkstra, A*, D*, RRT*, LPA* i biološki inspirirane algoritme. Predstavljena su područja primjene, kao i pozitivne i negativne strane svakog od njih.

4.1. Jedan objekt interesa

Inicijalna verzija problema planiranja za mobilne robote razumno sadrži jedan statičan objekt ili točku u prostoru koja predstavlja cilj. Rezultat takvog algoritma putanja je od početne točke do tog cilja u geometrijskom prostoru ili prostoru kontrolnih varijabli našeg sustava. Ovisno o potrebama, rješenja takvog problema mogu se dobiti spomenutim algoritmima. U geometrijskom prostoru na primjer A*, ako nam je prostor aproksimiran grafom, algoritmima uzorkovanja prostora (PRM*, RRT*, FMT*, BIT* i drugi) ako nije. U prostoru kontrolnih varijabli našeg sustava možemo koristiti MPC ako nam je model sustava poznat, a ako nije na primjer metode podržanog učenja.

U (Ellefsen, Lepikson, & Albiez, 2017), autori se bave automatizacijom robotske inspekcije struktura u stvarnom svijetu. U tu svrhu razvili su Multiobjective Evolutionary Optimization (MOEA) algoritam. Cilj je algoritma obići strukturu prolazeći kroz niz definiranih pozicija kako bi se dovršila inspekcija. Na danom modelu strukture definiranom trokutima, algoritam pronalazi 3D putanju koja maksimizira pokrivenost same strukture i minimizira utrošenu energiju robotskog sustava. Pristup je testiran na podvodnom robotskom sustavu (UUV) sa rezultatima prikazanim u sljedećoj slici.



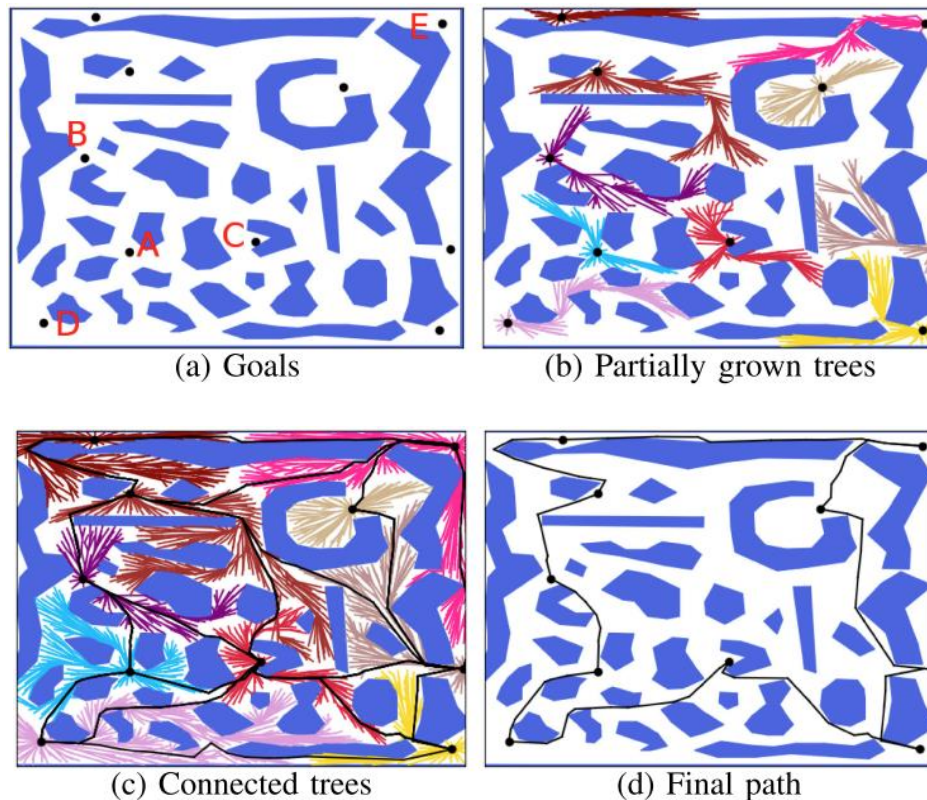
Slika: putanje isplanirane MOEA algoritmom sa različitim inicijalnim parametrima inspekcije. Struktura prikazana u crvenom, putanja u crnom.

Sličnim problemom inspekcije struktura bavi se (Shi, Mehrooz, & Jacobsen, 2021). Autori koriste UAV dronove koji prate planirane putanje za obilazak. Ulaz u algoritam je *point cloud* reprezentacija strukture koja se onda simplificira u pojednostavljeni model. Iz tog se modela stvara konveksna ljuska koja predstavlja graf korišten za finalno planiranje putanje. S obzirom da je cilj ovakvog algoritma prekriti cijeli volumen objekta, problem možemo shvatiti kao evoluciju TSP-a i zove se Coverage Path Planning (CPP). Tradicionalno je rješavan iteracijama dva algoritma: Art Gallery Problem (AGP), gdje se računa minimalan broj pozicija za pokriti volumen, i TSP. Autori ovog rada predstavljaju algoritam u kojem su zamijenili AGP sa sekvencijalnom konveksnom optimizacijom, a TSP sa novim algoritmom traženja putanje.

4.2. Više objekata interesa

U slučaju da imamo više od jednog objekta ili točke interesa koje želimo obići isplaniranom putanjom okrećemo se drugom skupu algoritama. Ponovno, ovisno o pretpostavkama problema ako nam je prostor prikazan potpuno povezanim grafom možemo koristiti TSP i njegove varijante. S druge strane, ako trebamo uračunati i prepreke taj proces postaje kompliciraniji, kao što je objašnjeno u (Chen, Frey, Arm, & Hutter, 2023). U predavanju (Prague) prezentiran je pregled pristupa korisnih za probleme MTP i MGMP, u kojima planiramo putanju za robota s više objekata interesa u sceni.

U ovoj situaciji jedan od pristupa planiranju može biti izgradnja više RRT stabala (Devaurs, Simeon, & Cortes, 2014). U (Janoš, Vonásek, & Pěnička, 2021), autori koriste RRT algoritam kao inspiraciju za planiranje putanje s više statičkih objekata interesa. Algoritam koji su razvili, temelji se na kreiranju više stabala tehnikama iz RRT. Prvi korak algoritma definicija je redoslijeda posjećivanja objekata po TSP načelima. Za definiciju udaljenosti među objektima razvili su Space Filling Forest (SFF*), varijantu RRT algoritma, koja iz svakog objekta interesa generira stablo i pokušava ih povezati. Ovakav algoritam isprepliće provođenje TSP algoritma i njegovih evolucija i samo planiranje putanje po generiranom rasporedu te je efikasniji u usporedbi sa sličnim pristupima (Janoš, Vonásek, & Pěnička, 2021). Graf generiran ovim algoritmom prikazan je u sljedećoj slici.



Slika: Space Filling Forest (SFF*) algoritam primijenjen na 2D problem planiranja s više objekata interesa.

U (Forkan, Rizvi, & Chowdhury, 2022), autori koriste geometrijski pristup parametrizacije lukova kako bi riješili problem planiranja putanje za zračne dronove (UAV) pri obilasku više statičnih objekata interesa na istoj visini. Problem je predstavljen kao optimizacija u nelinearnom sustavu s ograničenjima te je validiran u nekoliko numeričkih eksperimenata. Iako sub optimalan, ovakav je pristup računalno efikasan i posebno primjenjiv kod dronova fiksiranih krila.

U (Chen & Chen, 2014), autori se ponovno bave planiranjem putanje za zračne dronove (UAV) u ratnim zonama na području pokrivenom radarima. Kako bi došli do željenog cilja i izbjegli detekciju radarom, generiraju graf napravljen na temelju Voronoi dijagrama radarskih pozicija. Često korišten kao metoda teselacije, u matematici je Voronoi dijagram podjela ravnine na područja blizu svakog od danog skupa objekata. Njihov pristup planira putanju Dijkstra algoritmom na stvorenom grafu. Takve su putanje odabrane maksimiziranjem udaljenosti od pojedinih radara i minimiziranjem duljine same putanje.

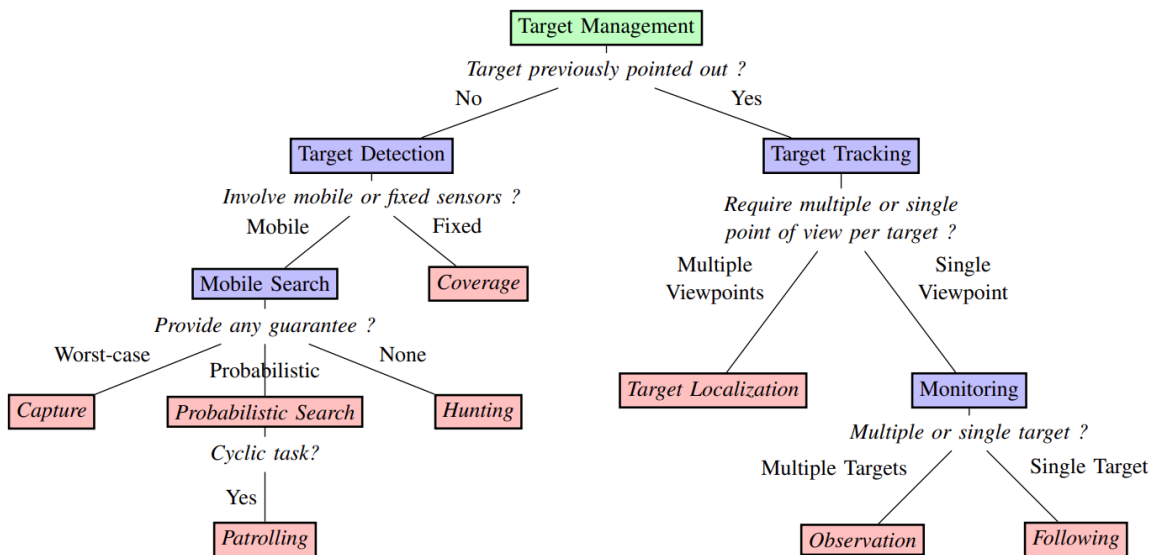
U (Hahn & Feld, 2017), autori su razvili algoritam planiranja temeljen na Cross Entropy Online Planning paradigmi u kontinuiranim okruženjima i aktuaciji za agente. U takvim domenama slično MPC pristupima, može se generirati i testirati sekvence akcija te ih izvršavati s obzirom na njihovu optimalnosti. Takav pristup inače podrazumijeva poznavanje modela cjelokupnog sustava i iscrpno računanje, ali u ovom članku autori koriste ideje polja intuitivne ili naivne fizike. Na taj način dobiju simplificirani model iz same interakcije

djelova okruženja po kojem mogu dalje optimizirati situaciju. Tako upravljane agente nazivaju Intuitive Active Autonomous Agents.

5. Planiranje u dinamičnim okolinama

Apsolutna brzina kretanja robota i ostalih elemenata okoline sama po sebi uglavnom ne čini okolinu dinamičkom. Kada pričamo o dinamičkim okruženjima, bitno je promatrati relativnu brzinu elemenata okoline s obzirom na samog robota. Sam proces planiranja se tad otklanja od pojma optimalnosti i tradicionalnih algoritama. Iako su oni još uvijek korisni, okrećemo se novom skupu pojmova i algoritama koji će biti predstavljeni u ovom poglavlju.

Pregled i taksonomija područja detekcije i praćenja objekata s više robota dani su u (Robin & Lacroix, 2014). U radu su definirane klase misija i problema, analizirani su brojni pristupi i modeli, te su izdvojeni nedostaci svake od spomenutih primjena. Primjeri predstavljenih misija su potraga, praćenje, patroliranje, nadgledanje, lov i izbjegavanje. Klasifikacija problema predstavljena je u slici.



Slika: Taksonomija područja detekcije i praćenja objekata.

5.1. Predviđanje kretanja

U situacijama gdje se dijelovi okruženja kreću, planiranje putanje za mobilnog robota temeljeno samo na trenutnom stanju tog okruženja gubi smisao. U teoriji, optimalna bi putanja trebala uzeti u obzir informacije o budućem kretanju ostalih objekata. Takvim informacijama uglavnom nemamo pristup tokom planiranja te je potrebno provoditi proces predviđanja njihove buduće putanje.

Problem predikcije kretanja objekta u okruženju izuzetno je zahtjevan sam za sebe. Rezultati algoritma u ovoj domeni ovise u velikoj mjeri o ustrojenosti ponašanja tih objekata. Ako su objekti pasivni elementi, njihovo se kretanje uglavnom može predvidjeti fizikalnim zakonima,

in (Chakrabarty, i dr., 2016). U suprotnom, ako njihova kretnja uključuje ponašanja kao što je donošenje odluka, sposobnost predviđanja njihovih kretnji opada.

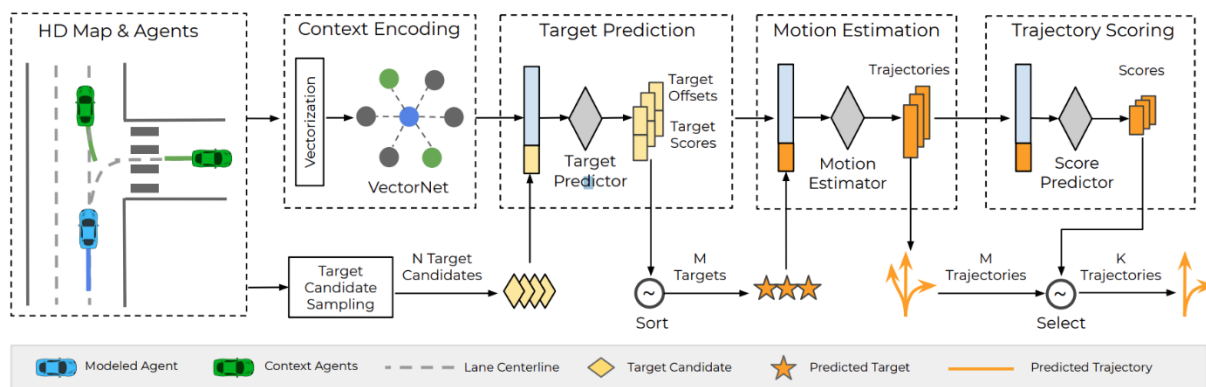
U (Helbing & Molnár, 1995), autori simuliraju interakciju pješaka u sceni preko koncepta socijalnih sila. U njihovom modelu te sile predstavljaju unutarnje motivacije ponašanja pojedinaca. Ovakvim su pristupom uspjeli s velikim stupnjem realizma simulirati ponašanje masa ljudi.

U (Makansi, Cicek, Marrakchi, & Brox, 2021), autori proučavaju utjecaj raznih faktora na kretnje agenata u višeagentskim okolinama, na primjer u košarci. Takvi problemi uključuju uzorke ponašanja na razini pojedinog igrača, dogovorene taktike na razini tima i interakciju među igračima u pravom vremenu. Koristeći uvažene metode predikcije kretnji u takvim okruženjima, zaključili su da te metode predikcije temelje uglavnom na kretnji samog agenta, ne oslanjajući se na interakciju, te bi trebalo razviti bolje metode.

U (Liu, He, Liu, & Chen, 2024), autori su predstavili algoritam Intention-aware denoising Diffusion Model (IDM), koji uz pomoć difuzijskog algoritma dubokog učenja predviđa buduće putanje pješaka u okruženju. Dva glavna problema takvog pristupa su povezanost namjere agenata sa nesigurnošću u okolini i sporost procesa difuzije. IDM rješava oba problema odvajanjem nesigurnosti namjere i akcije u procesu difuzije. Metoda je testirana na poznatim bazama podataka Stanford Drone Dataset i ETH/UCY (Pellegrini, Ess, Schindler, & Van Gool, 2009), (Robicquet, Sadeghian, Alahi, & Savarese, 2016), te je postigla vrhunske rezultate.

Još jedan od generativnih pristupa dubokog učenja, Generative Adversarial Network (GAN) mreže, korišten je u (Wang, Cui, Yalamanchi, Moorthy, & Djuric, 2020). Te su se mreže pokazale korisne za predikciju kretnji u prometu (Gupta, Johnson, Fei-Fei, Savarese, & Alahi, 2018), (Sadeghian, et al., 2019), (Kosaraju & Kanakamedala, 2019). Autori u ovom radu kombiniraju GAN mreže sa metodama rasterizacije pogleda na scenu iz ptičje perspektive u algoritmu koji zovu Scene-Compliant GAN. Metoda je ispitana na ATG4D bazi podataka (Meyer, Laddha, Kee, Vallespi-Gonzalez, & Wellington, 2019).

U (Gao, et al., 2020), umjesto predstavljanja situacije u prometu kao slike iz ptičje perspektive autori koriste Graph Neural Network (GNN) arhitekturu. Prednost nad konvolucijskim mrežama predstavlja vektorizacija okruženja visoke rezolucije, čime se preskače skup proces enkodiranja slike. Razvijena arhitektura VectorNET postiže jednake ili bolje rezultate predikcije putanje u usporedbi sa drugim metodama, koristeći smanjenje korištenih resursa za red veličine. Isti su autori na temelju VectorNET u (Zhang, Wang, & Meng, 2021) algoritam TNT, Target driveN Trjectory Prediction. TNT temeljen je na pretpostavci da se ponašanje agenata u sceni može predstaviti budućim ciljnim stanjima. Sekvence stanja koje predstavljaju buduću putanju su onda generirane s obzirom na ta predviđena ciljna stanja. Ovaj pristup rješava problem multi modalne prirode ponašanja agenata u sceni. Arhitektura TNT algoritma dana je u sljedećoj slici.



Slika: arhitektura TNT algoritma predstavljenog u (Zhao, et al., 2021).

U (Trentin, Artuñedo, Godoy, & Villagra, 2023), autori problem predikcije kretnji u prometu rješavaju kombinacijom kinetičkih modela vozila, Dynamic Bayesian Network (DBN) mreža i teorije markovljevih lanaca (MDP). DBN mreže koriste se za modeliranje odnosa među stanja i nemjerljivih namjera agenata u sceni. Markovljev lanac kreiran iz kinematičkog modela koristi se za predikciju budućih putanja vozila, temeljenu na trenutnom stanju, namjerama predviđenim DBN mrežom i nesigurnosti u same predikcije. Te se multi modalne predikcije onda šalju samom vozilu za planiranje putanje. Metoda se pokazala uspješnom na stvarnim primjerima iz javno dostupnih baza podataka.

5.2. Jedan objekt interesa

Jedan od originalnih je algoritama za problem dostizanja pokretnog cilja Moving Target Search (DTS) algoritam iz 1995. godine (Ishida & Korf, 1995). Generalizacija Learning Real Time A* (LRTA*), MTS koristi heurističke informacije o svim mogućim lokacijama cilja u matrici. Autori su dokazali da algoritam garantirano doseže objekt interesa kada se on kreće sporije od samog agenta koji ga hvata. Razvijene su dvije verzije algoritma, potpuna i efikasna.

U (Chimura, Toh, & Motooka, 1994), autori grade na radu predstavljenom u (Ishida & Korf, 1995) i predstavljaju novi algoritam Trailblazer Search za pretragu i dostizanje jednog objekta interesa. On kroz operaciju gradi bazu informacija u mapi o regiji u kojoj radi pretragu. Preko te mape algoritam pretražuje putanje za dohvaćanje nove pozicije u prostoru te postiže bolje performanse od prethodnog MTS algoritma.

U (Triharminto, Adji, & Setiawan, 2011), autori su razvili adaptivni dinamički algoritam planiranja putanje, za zadatak presretanja mobilnog cilja. Njihovo rješenje predstavlja evoluciju L+Dumo algoritma i testirano je na simulaciji u C++. Originalni L+Dumo algoritam (Triharminto, Prabuwo, Adji, & Setiawan, 2013) koristi se za presretanje mobilnog cilja u 3D prostoru pomoću zračnog drona efikasnim izračunom. Rezultati su testirani na problemu presretanja jednog objekta u 3D prostoru.

U (Drake, Koziol, & Chabot, 2018), autori su još jednom razvili algoritam planiranja putanje za presretanje mobilnog cilja s naglaskom na brzinu izračuna. Algoritam aproksimira okruženje diskretizacijom i koristi inkrementalni pristup koji ponovno računanje obavlja samo pri kretnji cilja. Također, početno rješenje za izračun nove putanje prethodna je putanja, čime se povećava efikasnost samog algoritma. Ovakav pristup pokazuje značajna poboljšanja s

povećanjem kompleksnosti problema, na primjer brzini kretanja cilja. S obzirom na druge algoritme, predstavlja ubrzanje od 12 puta, dok su pronađene putanje u prosjeku 1.5% dulje.

5.3. Više objekata interesa

U (Yin, Ding, & Li, 2008), autori se bave pronalaskom putanje koja pokriva više objekata u dinamičkom grafu koji predstavlja Spatial Network Database (SND). Razvijeni algoritam pronalazi najkraću putanju do mobilnih objekata vođenjem računa o kretanjima agenata i promjenama do kojih to u grafu dovodi. Kompleksnost algoritma je $O(k \log_2 i)$ dok je kompleksnost originalnog Dijkstrinog algoritma u takvom problemu $O((i + k)^2)$.

U (Cannon, Rose, & Ruml, 2012), autori su razvili algoritam Partitioned Learning Real Time A* (PLRTA*) za planiranje kretanja u stvarnom vremenu u dinamičkim okruženjima. S obzirom na druge heurističke algoritme, ovaj funkcionira efikasno i u višedimenzionalnim okruženjima, čestim u problemima robotike, na primjer kod robotskih manipulatora. Algoritam planira putanje s obzirom na predviđanja kretanja ostalih objekata u sceni te je testiran u simulaciji i stvarnom svijetu.

U (Nawaz & Ornik, 2023), autori se bave planiranjem za tim od više agenata i više objekata interesa u prostoru aproksimiranom grafom kao MDP proces. Agenti u timu ne komuniciraju tijekom operacije te je optimalan pristup u takvom okruženju eksponencijalno kompleksan s brojem objekata interesa. Autori stoga u svom algoritmu žrtvuju optimalnost za vremensku efikasnost. Takav je algoritam u stvarnom svijetu brži od optimalnih procedura i optimalniji od dostupnih heurističkih pristupa.

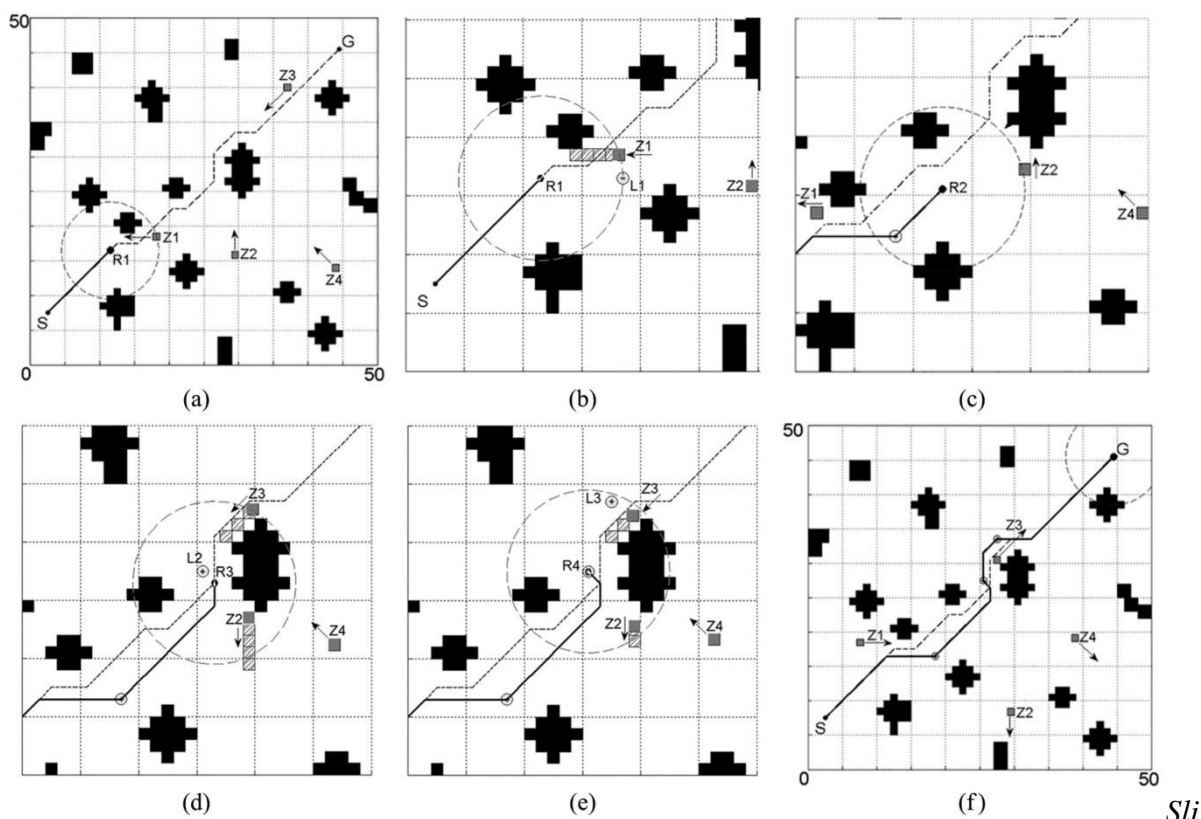
U diplomskom radu (Tin, 2004), autor piše o efektima nesigurnosti parametara u dinamičkim okruženjima za zračne robote (UAV). Za takve okolnosti, razvijena je hijerarhija upravljanja nazvana Receding Horizon Mixed Integer Linear Programming (RH-MILP). Suočena s brzo promjenjivim situacijama i nesigurnostima omogućuje robotskom sustavu da brzo reagira i ponovno planira optimalne putanje za UAV. Autor uzima u obzir više objekata interesa kao i prepreke u prostoru u kojem UAV operira. Algoritam je testiran u simulaciji i eksperimentima.

U (Zhou, Ding, Huang, Takei, & Tomlin, 2018), razmatra se igra s više agenata nazvana "reach-avoid". Ideja je igre da skupina napadača mora doseći cilj dok izbjegava skupinu branitelja koji ih pokušavaju uhvatiti. Za razliku od problema potjere, agenti ovdje osim izbjegavanja moraju i doseći konkretan cilj. Ovisno o broju agenata, pronalazak rješenja za ovaj problem izrazito je kompleksan. Autori kao rješenje predlažu algoritam otvorene petlje, gdje se igrači obvezuju na svoje ponašanje prije početka igre. Ovakvim pristupom smanjuje se složenost samog problema koji stoga postaje rješiv. Razvijena metoda testirana je u simulaciji i u stvarnom svijetu na zračnim dronovima u 2D prostoru.

U (Xie, Botea, & Kishimoto, 2017), razvijen je skalabilan suboptimalni algoritam za praćenje više objekata interesa s više agenata. Ključan detalj takvog problema dodjela je samih objekata agentima. Razvijeni algoritam tako uključuje dinamičko preračunavanje dodjela. U optimalnim verzijama taj postupak ima limitiranu skalabilnost, dok ovdje korišten pristup, iako suboptimalan, zadržava svojstvo potpunosti u razumnom vremenu. Metoda je testirana u brojnim simuliranim scenarijima i dokazani su ciljevi skalabilnosti, nadmašujući prethodne metode.

U (Abdel-Basset & Mohamed, 2023), autori koriste hibridni pristup Hybrid Differential Evolution (HDE) koji kombinira Differential Evolution (DE) algoritam i dvije nove sheme ažuriranja za pronalazak rješenja za planiranje s više pokretnih objekata u sceni. Temeljen na Bayesovoj teoriji vjerojatnosti, algoritam postavlja vjerojatnost susretanja objekta interesa kao vrijednosnu funkciju optimizacijskog problema. U diskretiziranoj aproksimaciji okruženja, metodama je evolucije onda rađena potraga za najboljim rješenjem, u ovom slučaju DE. Algoritam je temeljito testiran i pokazao se superioran u usporedbi s drugim pristupima za ovaj problem.

Još jedan pristup korišten u diskretiziranoj aproksimaciji okruženja predstavljen je u (Zhang & Li, 2017). Algoritam koji su autori razvili bavi se planiranjem putanje za agenta s jednim statičkim ciljem, ali u okruženju gdje postoji više dinamičkih prepreka. Zadatak je algoritma izbjeći prepreke i doseći cilj. Koraci su sljedeći: prvo Dijkstrinim algoritmom pronalazi se optimalna putanja od početka do cilja, nakon toga, agent je prati i predviđa kretanje prepreka. Kada se detektira potencijalan sudar, agent je algoritmom klizećeg prozora lokalno izbjegava te koristi A* za dohvaćanje novog plana. Primjer funkcioniranja opisanog algoritma dan je u sljedećoj slici.



ka: primjer epizode kretanja od početne do finalne točke s mobilnim preprekama u sceni.

Sličan problem rješava i (Afzalov, Lotfi, Inden, & Aydin, 2022). U okruženju sa više mobilnih agenata, algoritam MultiplePursuers TrailMax planira putanju s ciljem izbjegavanja drugih agenata na što duže vrijeme. Kao osnovu, autori su uzeli TrailMax algoritam (Moldenhauer & Sturtevant, 2009) te ga nadogradili za scenarij s više agenata. I u ovom slučaju autori uzimaju u obzir diskretiziran prikaz okruženja. U usporedbi s drugim algoritmima, njihov pristup poboljšava performanse za 13%.

6. Zaključak

Ovaj rad predstavlja pregled pristupa planiranju za autonomne mobilne robote u dinamičkim okruženjima. Na početku rada, dan je opis problema s kojim smo suočeni te su navedeni neki od *state-of-the-art* rješenja na tom području. Nakon toga su za definirani problem predstavljeni brojni korisni algoritmi, od temeljnih algoritama kao što su Dijkstra i A*, do najnovijih pristupa kao što je *diffusion planning* i kombiniranje podržanog učenja s tradicionalnim metodama optimizacije. Na kraju su predstavljeni brojni radovi koji se bave planiranjem u statičkim i dinamičkim okruženjima.

Ideje vezane uz planiranje u dinamičkim okruženjima razvijaju se već godinama. Ipak, tek su nedavno tehnologija i dostupne robotske platforme dosegle razinu razvoja gdje implementacija takvih algoritama ima smisla. Računalne su platforme postale dovoljno snažne i kompaktne, a sami robotski sustavi dovoljno agilni i spretni da ih izvršavaju. Ovaj je pregled prikazao trenutno stanje algoritama korištenih u tom području, osnovne verzije problema rješive su već poznatim pristupima, dok se u literaturi pojavljuju rješenja za sve kompleksnije zadatke mobilnih robota.

Dva su moguća smjera daljnjeg razvoja rada, specijalizirani algoritmi za konkretne uske primjene i opći algoritmi za pokrivanje šireg spektra problema. Također, sa rastom računalne moći dostupne na samom robotu otvaraju se nove mogućnosti pri korištenju spomenutih algoritama. Oba su smjera obećavajuća te će vrijeme pokazati koji će se pokazati korisnijim.

LITERATURA

- Abdel-Basset, & Mohamed. (2023). An Efficient Evolution-Based Technique for Moving Target Search with Unmanned Aircraft Vehicle: Analysis and Validation. *Mathematics*, 11, 2606.
- Afzalov, A., Lotfi, A., Inden, B., & Aydin, M. (2022, August 9). A Strategy-Based Algorithm for Moving Targets in an Environment with Multiple Agents. *SN Computer Science*, 3, 435. doi:10.1007/s42979-022-01302-x
- Bansal, A., Borgnia, E., Chu, H., Li, J., Kazemi, H., Huang, F., . . . T. (2024). Cold diffusion: Inverting arbitrary image transforms without noise. In *Advances in Neural Information Processing Systems* (p. 36).
- Cannon, J., Rose, K., & Ruml, W. (2012). Real-time heuristic search for motion planning with dynamic obstacles. *AI Communications*, 27, 345-362. doi:10.3233/aic-140604
- Čehovin, L., Kristan, M., & Leonardis, A. (2011). TraX: The visual Tracking eXchange protocol and library. *Neurocomputing*, 260, 5-8. doi:10.1016/j.neucom.2017.02.036
- Chakrabarty, D., Dourmachkin, P., Tomasik, M., Frebel, A., Vuletic, V., & Mechanics. (2016). Classical Mechanics. Fall 2016. Massachusetts Institute of Technology. *Classical Mechanics. Fall 2016. Massachusetts Institute of Technology*.
- Chen, C., Frey, J., Arm, P., & Hutter, M. (2023, November). SMUG Planner: A Safe Multi-Goal Planner for Mobile Robots in Challenging Environments. *IEEE Robotics and Automation Letters*, 8, 7170-7177. doi:10.1109/ra.2023.3311207
- Chen, X., & Chen, X. (2014, May). The UAV dynamic path planning algorithm research based on Voronoi diagram. In *The 26th Chinese Control and Decision Conference (2014 CCDC)*. IEEE. doi:10.1109/ccdc.2014.6852323
- Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., & Song, S. (2023). Diffusion policy: Visuomotor policy learning via action diffusion.
- Chimura, D., Toh, R., & Motooka, S. (1994). Ultrasonic Direction Measurement Method Using Sensitivity Compensated Transmitting Signal and Pulse Compression. *Physics Procedia*, 70, 476-479. doi:10.1016/j.phpro.2015.08.289
- Choset, H., Hager, G., & Dodds, Z. (2023). NONSMOOTH ANALYSIS, CONVEX ANALYSIS, AND THEIR APPLICATIONS TO MOTION PLANNING. *International Journal of Computational Geometry & Applications*, 09, 447-469. doi:10.1142/s0218195999000261
- Choset, H., Yeong Lee, J., Hager, G., & Dodds, Z. (2024). Robotic Motion Planning: Potential Functions. Retrieved from https://www.cs.cmu.edu/motionplanning/lecture/Chap4-Potential-Field_howie.pdf
- Choudhury, S., Gammell, J., Barfoot, T., Srinivasa, S., & Scherer, S. (2016, May). Regionally accelerated batch informed trees (RABIT*): A framework to integrate local information into optimal path planning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4207–4214). IEEE. doi:10.1109/icra.2016.7487615
- Chua, K. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in neural information processing systems* (p. 31).
- Devaurs, D., Simeon, T., & Cortes, J. (2014, September). A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2991–2996). IEEE. doi:10.1109/iros.2014.6942975
- Dijkstra, E. (2022, July 12). A Note on Two Problems in Connexion with Graphs. In *Edsger Wybe Dijkstra* (pp. 287-290). ACM. doi:10.1145/3544585.3544600

- Drake, D., Koziol, S., & Chabot, E. (2018). Mobile Robot Path Planning With a Moving Goal. *IEEE Access*, 6, 12800-12814. doi:10.1109/access.2018.2797070
- Eberhard, O., Hollenstein, J., Pinneri, C., & Martius, G. (2022, September). Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *The Eleventh International Conference on Learning Representations*.
- Ellefsen, K., Lepikson, H., & Albiez, J. (2017, December). Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures. *Applied Soft Computing*, 61, 264-282. doi:10.1016/j.asoc.2017.07.051
- Eysenbach, B., Gupta, A., Ibarz, J., & Levine, S. (2018). Diversity is all you need: Learning skills without a reward function.
- Forkan, M., Rizvi, M., & Chowdhury, M. (2022, October 14). Optimal path planning of Unmanned Aerial Vehicles (UAVs) for targets touring: Geometric and arc parameterization approaches. *PLOS ONE*, 17, e0276105. doi:10.1371/journal.pone.0276105
- Gammell, J., Barfoot, T., & Srinivasa, S. (2020, January 27). Batch Informed Trees (BIT*): Informed asymptotically optimal anytime search. *The International Journal of Robotics Research*, 39, 543-567. doi:10.1177/0278364919890396
- Gammell, J., Srinivasa, S., & Barfoot, T. (2014, September). Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2997–3004). IEEE. doi:10.1109/iros.2014.6942976
- Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., & Schmid, C. (2020, June). VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11525–11533). IEEE. doi:10.1109/cvpr42600.2020.01154
- Gentilini, F., & Shimada, K. (2012). Multi-Goal Path Optimization for Robotic Systems with Redundancy based on the Traveling Salesman Problem with Neighborhoods.
- Gentilini, I., Margot, F., & Shimada, K. (2013, April). The travelling salesman problem with neighbourhoods: MINLP solution. *Optimization Methods and Software*, 28, 364-378. doi:10.1080/10556788.2011.648932
- Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., & Alahi, A. (2018, June). Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2255–2264). IEEE. doi:10.1109/cvpr.2018.00240
- Hahn, C., & Feld, S. (2017). Collision Avoidance using Intuitive Physics. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE. doi:10.1109/inista.2018.8466298
- Hart, P., Nilsson, N., & Raphael, B. (1968, July). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4, 100-107. doi:10.1109/tssc.1968.300136
- Hauser, K. (2015, May). Lazy collision checking in asymptotically-optimal motion planning. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2951–2957). IEEE. doi:10.1109/icra.2015.7139603
- Helbing, D., & Molnár, P. (1995, May 1). Social force model for pedestrian dynamics. *Physical Review E*, 51, 4282-4286. doi:10.1103/physreve.51.4282
- Ishida, T., & Korf, R. (1995, June). Moving-target search: a real-time search for changing goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 609-619. doi:10.1109/34.387507
- Janner, M., Du, Y., Tenenbaum, J., & Levine, S. (2022). Planning with diffusion for flexible behavior synthesis.

- Janoš, J., Vonásek, V., & Pěnička, R. (2021). Calculations of critical speeds of multi-span shafts by the method of successive addition of spans. *Applications of Mathematics*, 07, 201-226. doi:10.21136/am.1962.102802
- Janson, L., & Pavone, M. (2015). Fast Marching Trees: A Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions. In *Springer Tracts in Advanced Robotics* (Vol. 34, pp. 667-684). Springer International Publishing. doi:10.1007/978-3-319-28872-7_38
- Jenelten, F., He, J., Farshidian, F., & Hutter, M. (2024, January 17). DTC: Deep Tracking Control. *Science Robotics*, 9. doi:10.1126/scirobotics.adh5401
- Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4, 237–285. doi:10.1613/jair.301.S2CID1708582
- Karaman, S., & Frazzoli, E. (2011). Incremental Sampling-based Algorithms for Optimal Motion Planning. In *Robotics: Science and Systems VI* (Vol. 30, pp. 846–894). Robotics: Science and Systems Foundation. doi:10.15607/rss.2010.vi.034
- Karapetyan, D., & Gutin, G. (2011, February). Lin–Kernighan heuristic adaptations for the generalized traveling salesman problem. *European Journal of Operational Research*, 208, 221-232. doi:10.1016/j.ejor.2010.08.011
- Karur, K., Sharma, N., Dharmatti, C., & Siegel, J. (2021, August 4). A Survey of Path Planning Algorithms for Mobile Robots. *Vehicles*, 3, 448-468. doi:10.3390/vehicles3030027
- Kavraki, L., Svestka, P., Latombe, J.-C., & Overmars, M. (1996, August). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12, 566-580. doi:10.1109/70.508439
- Kobilarov, M. (2012). Cross-Entropy Randomized Motion Planning. In *Robotics: Science and Systems VII* (Vol. 31, pp. 855–871). Robotics: Science and Systems Foundation. doi:10.15607/rss.2011.vii.022
- Kondo, K., Tordesillas, J., Figueroa, R., Rached, J., Merkel, J., Lusk, P., & How, J. (2023, May 29). Robust MADER: Decentralized and Asynchronous Multiagent Trajectory Planner Robust to Communication Delay. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1687–1693). IEEE. doi:10.1109/icra48891.2023.10161244
- Kosaraju, N., & Kanakamedala, V. (2019). FRIEND RECOMMENDATION USING GRAPH MINING ON SOCIAL MEDIA. *International Journal of Engineering Technology and Management Sciences*, 4, 57-65. doi:10.46647/ijetms.2020.v04i05.011
- Lavalle, S., & Kuffner, J. (2001). Randomized kinodynamic planning. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 20, pp. 378–400. IEEE. doi:10.1109/robot.1999.770022
- Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., & Hutter, M. (2020, October 21). Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5, 5986. doi:10.1126/scirobotics.abc5986
- Lerner, A., Chrysanthou, Y., & Lischinski, D. (2007, September). Crowds by Example. *Computer Graphics Forum*, 26, 655-664. doi:10.1111/j.1467-8659.2007.01089.x
- Liang, J., Payandeh, A., Song, D., Xiao, X., & Manocha, D. (2024). DTG: Diffusion-based Trajectory Generation for Mapless Global Navigation.
- Liu, C., He, S., Liu, H., & Chen, J. (2024). Intention-aware denoising diffusion model for trajectory prediction.
- Makansi, O., Cicek, O., Marrakchi, Y., & Brox, T. (2021, October). On Exposing the Challenging Long Tail in Future Prediction of Traffic Actors. doi:10.1109/iccv48922.2021.01290
- Margolin, L. (2005, February). On the Convergence of the Cross-Entropy Method. *Annals of Operations Research*, 134, 201-214. doi:10.1007/s10479-005-5731-0

- Meier, R., & Mujika, A. (2022). Open-ended reinforcement learning with neural reward functions. *Advances in Neural Information Processing Systems*, 35, 2465–2479.
- Meyer, G., Laddha, A., Kee, E., Vallespi-Gonzalez, C., & Wellington, C. (2019, June). LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12–677). IEEE. doi:10.1109/cvpr.2019.01296
- Moldenhauer, Carsten, and Nathan Reed Sturtevant. "Evaluating strategies for running from the cops." *Twenty-First International Joint Conference on Artificial Intelligence*. 2009.
- Nagabandi, A., Kahn, G., Fearing, R., & Levine, S. (2018, May). Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 7559–7566). IEEE. doi:10.1109/icra.2018.8463189
- Nawaz, F., & Ornik, M. (2023, December). Multiagent, Multitarget Path Planning in Markov Decision Processes. *IEEE Transactions on Automatic Control*, 68, 7560-7574. doi:10.1109/tac.2023.3286807
- Noon, C., & Bean, J. (1991, August). A Lagrangian Based Approach for the Asymmetric Generalized Traveling Salesman Problem. *Operations Research*, 39, 623-632. doi:10.1287/opre.39.4.623
- Noon, C., & Bean, J. (1993, February). An Efficient Transformation Of The Generalized Traveling Salesman Problem. *INFOR: Information Systems and Operational Research*, 31, 39-44. doi:10.1080/03155986.1993.11732212
- Sucan, I., Moll, M., & Kavraki, L. (2012). The open motion planning library. *IEEE Robotics & Automation Magazine*, 19, 72–82.
- Pan, X. (2010). Approximate shortest path algorithms for sequences of pairwise disjoint simple polygons. *Proceedings of the 22nd Annual Canadian Conference on Computational Geometry, CCCG 2010*, (pp. 175–178).
- Pellegrini, S., Ess, A., Schindler, K., & Van Gool, L. (2009, September). You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE. doi:10.1109/iccv.2009.5459260
- Pinneri, C. (2021). Sample-efficient cross-entropy method for real-time planning. *PMLR*.
- Pintea, N., Pop, P., & Chira, C. (2017, August 7). The generalized traveling salesman problem solved with ant algorithms. *Complex Adaptive Systems Modeling*, 5, 8–2017. doi:10.1186/s40294-017-0048-9
- Plaku, E., Kavraki, L., & Vardi, M. (2010, June). Motion Planning With Dynamics by a Synergistic Combination of Layers of Planning. *IEEE Transactions on Robotics*, 26, 469-482. doi:10.1109/tro.2010.2047820
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *1*, 3.
- Richards, A., & How, J. (2006). Robust variable horizon model predictive control for vehicle maneuvering. *International Journal of Robust and Nonlinear Control*, 16, 333-351. doi:10.1002/rnc.1059
- Robicquet, A., Sadeghian, A., Alahi, A., & Savarese, S. (2016). Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. In *Computer Vision – ECCV 2016* (pp. 549-565). Springer International Publishing. doi:10.1007/978-3-319-46484-8_33
- Robin, C., & Lacroix, S. (2014). Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots*, 40, 729-760. doi:10.1007/s10514-015-9491-7
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022, June). High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10684–10695). IEEE. doi:10.1109/cvpr52688.2022.01042
- Romero, A., Song, Y., & Scaramuzza, D. (2023). Actor-critic model predictive control.

- Rubinstein, R., & Kroese, D. (1999). Combinatorial Optimization via Cross-Entropy. In *Information Science and Statistics* (pp. 129-186). Springer New York. doi:10.1007/978-1-4757-4321-0_4
- Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofghi, H., & Savarese, S. (2019, June). SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1349–1358). IEEE. doi:10.1109/cvpr.2019.00144
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. *International conference on machine learning, 17*, 375-403. doi:10.1353/jph.2005.0024
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms.
- Shi, L., Mehrooz, G., & Jacobsen, R. (2021, June 15). Inspection Path Planning for Aerial Vehicles via Sampling-based Sequential Optimization. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. doi:10.1109/icuas51884.2021.9476784
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., . . . Hassabis, D. (2017). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362, 1140-1144. doi:10.1126/science.aar6404
- Smith, K., Battaglia, P., & Vul, E. (2013). A Null Method for Measuring Electromotive Force with a Ballistic Galvanometer and Condensers. *American Journal of Physics*, 6, 263-264. doi:10.1119/1.1991352
- Snyder, L., & Daskin, M. (2006, October). A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research*, 174, 38-53. doi:10.1016/j.ejor.2004.09.057
- Song, Y., Romero, A., Müller, M., Koltun, V., & Scaramuzza, D. (2023, September 13). Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Science Robotics*, 8, 1462. doi:10.1126/scirobotics.adg1462
- Stanley, K., & Miikkulainen, R. (2002). Efficient evolution of neural network topologies. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. 2. IEEE. doi:10.1109/cec.2002.1004508
- Stentz, A. (1995, August). The focussed d^* algorithm for real-time replanning. *IJCAI*, 95, 1652–1659.
- Strub, M., & Gammell, J. (2020, May). Adaptively Informed Trees (AIT*): Fast Asymptotically Optimal Path Planning through Adaptive Heuristics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3191–3198). IEEE. doi:10.1109/icra40945.2020.9197338
- Strub, M., & Gammell, J. (2020, May). Advanced BIT* (ABIT*): Sampling-Based Planning with Advanced Graph-Search Techniques. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 130–136). IEEE. doi:10.1109/icra40945.2020.9196580
- Tang, Z., & Ma, H. (2021, July 1). An overview of path planning algorithms. *IOP Conference Series: Earth and Environmental Science*, 804, 022024. doi:10.1088/1755-1315/804/2/022024
- Tin, C. (2004). Robust multi-UAV planning in dynamic and uncertain environments. *Diss. Massachusetts Institute of Technology*.
- Tordesillas, J., & How, J. (2021). MADER: Trajectory Planner in Multiagent and Dynamic Environments. *IEEE Transactions on Robotics*, 38, 463-476. doi:10.1109/tro.2021.3080235
- Trentin, V., Artuñedo, A., Godoy, J., & Villagra, J. (2023, May). Multi-Modal Interaction-Aware Motion Prediction at Unsignalized Intersections. *IEEE Transactions on Intelligent Vehicles*, 8, 3349-3365. doi:10.1109/tiv.2023.3254657

- Triharminto, H., Adji, T., & Setiawan, N. (2011, November). Dynamic uav path planning for moving target intercept in 3D. In *2011 2nd International Conference on Instrumentation Control and Automation* (pp. 157–161). Bandung, Indonesia: IEEE. doi:10.1109/ica.2011.6130148
- Triharminto, H., Prabuwo, A., Adji, T., & Setiawan, N. (2013, July 1). Adaptive Dynamic Path Planning Algorithm for Interception of a Moving Target. *International Journal of Mobile Computing and Multimedia Communications*, 5, 19-33. doi:10.4018/jmcmc.2013070102
- Vonásek, V., & Pěnička, R. (2019, September). Space-filling forest for multi-goal path planning. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1587–1590). IEEE.
- Wang, E., Cui, H., Yalamanchi, S., Moorthy, M., & Djuric, N. (2020, August 20). Improving Movement Predictions of Traffic Actors in Bird's-Eye View Models using GANs and Differentiable Trajectory Rasterization. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2340–2348). ACM. doi:10.1145/3394486.3403283
- Wang, R., Lehman, J., Clune, J., & Stanley, K. (2019, July 13). Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. doi:10.1145/3321707.3321799
- Wang, T., & Ba, J. (2019). Exploring model-based planning with policy networks.
- Wang, W., Fernández-gutiérrez, D., Doornbusch, R., Jordan, J., Shan, T., Leoni, P., . . . Rus, D. (2023, August). Roboat III: An autonomous surface vessel for urban transportation. *Journal of Field Robotics*, 40, 1996-2009. doi:10.1002/rob.22237
- Wang, X., Talluri, S., Rose, C., & Koedinger, K. (2020). Enhanced poet: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. *International Conference on Machine Learning. PMLR. ACM*. doi:10.1145/3330430.3333614
- Weinacht, P. (2004). Prediction of Projectile Performance, Stability, and Free-Flight Motion Using Computational Fluid Dynamics. *Journal of Spacecraft and Rockets*, 41, 257–263. doi:10.21236/ada417123
- Williams, G., Aldrich, A., & Theodorou, E. (2015). Model predictive path integral control using covariance variable importance sampling.
- Xie, F., Botea, A., & Kishimoto, A. (2017, August). A Scalable Approach to Chasing Multiple Moving Targets with Multiple Agents. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization*. doi:10.24963/ijcai.2017/624
- Xing, J., Bauersfeld, L., Song, Y., Xing, C., & Scaramuzza, D. (2023). Contrastive Learning for Enhancing Robust Scene Transfer in Vision-based Agile Flight.
- Yang, J., Shi, X., Marchese, M., & Liang, Y. (2008, November). An ant colony optimization method for generalized TSP problem. *Progress in Natural Science*, 18, 1417-1422. doi:10.1016/j.pnsc.2008.03.028
- Yin, X., Ding, Z., & Li, J. (2008, July). A shortest path algorithm for moving objects in spatial network databases. *Progress in Natural Science*, 18, 893-899. doi:10.1016/j.pnsc.2008.01.031
- Zhang, H.-M., & Li, M.-L. (2017, December). Rapid path planning algorithm for mobile robot in dynamic environment. *Advances in Mechanical Engineering*, 9, 168781401774740. doi:10.1177/1687814017747400
- Zhang, T., Wang, J., & Meng, N. (2021). Generative Adversarial Network Based Heuristics for Sampling-Based Path Planning. *IEEE/CAA Journal of Automatica Sinica*, 9, 64-74. doi:10.1109/jas.2021.1004275
- Zhao, C., Zhu, Y., Du, Y., Liao, F., & Chan, C.-Y. (2022, October). A Novel Direct Trajectory Planning Approach Based on Generative Adversarial Networks and

- Rapidly-Exploring Random Tree. *IEEE Transactions on Intelligent Transportation Systems*, 23, 17910-17921. doi:10.1109/tits.2022.3164391
- Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., . . . D. (2021, October). Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning* (pp. 895–904).
- Zhou, Z., Ding, J., Huang, H., Takei, R., & Tomlin, C. (2018, March). Efficient path planning algorithms in reach-avoid problems. *Automatica*, 89, 28-36. doi:10.1016/j.automatica.2017.11.035

POPIS OZNAKA I KRATICA

Popis oznaka i kratica navodi se abecednim redom.

MPC	Model Predictive Control
CEM	Cross Entropy Method
iCEM	Iterative Cross Entropy Method
PETS	Probabilistic Ensembles with Trajectory Sampling
POPLIN	Model Based Policy Planning
PlaNet	Deep Planning Network
PRM	Probabilistic Road Map
PRM*	Probabilistic Road Map Star
TSP	Traveling Salesman Problem
MTP	Multi-Goal Path Planning
MGMP	Multi-Goal Motion Planning
IDP	Iterative Dynamic Programming
OMPL	Open Motion Planning Library
RRT	Rapidly Exploring Random Tree
RRT*	Rapidly Exploring Random Tree Star
RL	Reinforcement Learning
MDP	Markov Decision Process
MBRL	Model Based Reinforcement Learning
MPPI	Model Predictive Path Integral
AMR	Autonomous Mobile Robot
FMT*	Fast Marching Tree star
BIT*	Batch Informed Trees star
RABIT*	Regionally Accelerated Batch Informed Trees star

ABIT*	Advanced Batch Informed Trees star
TSP	Travelling Salesman Problem
GTSP	Generalized Travelling Salesman Problem
TSPN	Travelling Salesman Problem Neighbourhoods
UAV	Unmanned Aerial Vehicle
UUV	Unmanned Underwater Vehicle
UGV	Unmanned Ground Vehicle
USV	Unmanned Surface Vehicle
PPO	Proximal Policy Optimization
TRPO	Trust Region Policy Optimization
SND	Spatial Network Database
MTS	Moving Target Search
LRTA*	Learning Real Time A*
PLRTA*	Partitioned Learning Real Time A*
RH MILP	Receding Horizon Mixed Integer Linear Programming
MOEA	Multiobjective Evolutionary Optimization
CPP	Coverage Path Planning
DE	Differential Evolution
HDE	Hybrid Differential Evolution
IDM	Intention-aware denoising Diffusion Model
GNN	Graph Neural Network
TNT	Target Driven Trajectory Prediction

SAŽETAK

Za uspješno funkcioniranje u prostoru, ljudi i životinje koriste se brojnim pretpostavkama razvijenim evolucijom i učenjem kroz život. To može biti intuicija o interakciji sa objektima, putanji kojom se stvari kreću ili namjerama drugih elemenata u sceni. U procesu automatizacije bilo kojeg mobilnog robota, njegov kreator polazi od puno užeg vida svijeta. Sliku o stanju svijeta gradi samo iz dostupnih senzora, a utječe na njega samo preko dostupnih aktuatora. Ograničen dostupnom računalnom snagom, robot treba pomoću senzora percipirati svijet oko sebe, planirati svoje ponašanje s obzirom na predodređene generalne smjernice i izvršavati svoj plan utjecajem na okolinu kroz svoje aktuatore. Problem planiranja za robota obuhvaća generiranje plana budućeg ponašanja robota za neki vremenski horizont s obzirom na dostupne informacije o svijetu oko sebe i krajnjim ciljem misije koju robot želi postići.

U stvarnom se svijetu ipak susrećemo sa brojnim preprekama koje otežavaju taj proces. Naš model možda ne obuhvaća svu kompleksnost interakcije robota s okolinom ili radimo s okruženjem koje uključuje druge agente/objekte. U tim slučajevima nesigurnosti i dinamičkih okolnosti, dovodi se u pitanje samo postojanje optimalnog plana, koja ovisi o budućem raspletu događaja. Ipak, iako planiranje u dinamičkim okolinama predstavlja iznimno težak skup problema, nove metode upućuju na dovoljno dobra rješenja. Ovaj rad predstavlja pregled trenutno dostupnih pristupa za problem planiranja za autonomne mobilne robote u statičnim i dinamičnim okruženjima.